

Utilizing Brain Activity to Non-Invasively Predict Blood Glucose Levels

by

Bryce A. Cranwell

November, 2016

Director of Thesis: Dr. Ricky Castles

Major Department: Engineering

Diabetes Mellitus is a chronic condition that must be routinely self-monitored to prevent life threatening complications, diabetic coma, and to mitigate long-term complications, such as neuropathy and kidney failure. Despite these risks, many people with diabetes do not properly self-monitor. An often cited reason for not properly self-monitoring is the discomfort associated with the lancing procedure. To alleviate the need for finger-sticks, this research has focused on a non-invasive blood glucose monitoring technique by combining electroencephalography (EEG) with artificial neural networks (ANN). Eleven volunteers participated in three, hour-long, recording sessions. Not all volunteers completed all sessions. During this time the volunteer's blood glucose level (BGL) was monitored in fifteen-minute increments, with the exception of the fifteenth minute. At fifteen minutes into the data collection, the volunteers drank a 52g carbohydrate load. EEG data was recorded continuously over the data collection period. From

the EEG data, frequency features were extracted and used to train an ANN that would predict the users BGL based upon EEG input. This data was used to determine which channels that correlated most strongly with glucose levels, determined the optimal number of neurons used for the training algorithm, and finally trained/tested an ANN. The channels that corresponded most consistently and strongly are channels AF4, F4, F8, FC5, O1, O2, and P8. The optimal number of neurons for the network was determined to be 99 neurons. Ultimately, the method of training an ANN to non-invasively predict BGLs has moderate success, but needs more research to be conclusive.

Utilizing Brain Activity to Non-Invasively Predict Blood Glucose Levels

A Thesis

Presented To the Faculty of the Department of Engineering

East Carolina University

In Partial Fulfillment of the Requirements for the Degree

Masters of Science Biomedical Engineering

by

Bryce A. Cranwell

November, 2016

© Bryce A. Cranwell, 2016

Utilizing Brain Activity to Non-Invasively Predict Blood Glucose Levels

by

Bryce A. Cranwell

APPROVED BY:

DIRECTOR OF
THESIS:

Ricky T. Castles, PhD

COMMITTEE MEMBER:

Jianchu Yao, PhD

COMMITTEE MEMBER:

Jitka A. I. Virag, PhD

CHAIR OF THE DEPARTMENT
OF ENGINEERING:

O. Hayden Griffin, PhD

DEAN OF THE
GRADUATE SCHOOL:

Paul J. Gemperline, PhD

DEDICATION

This Thesis work is dedicated to

KATHERINE RACKLEY

For inspiring my love of learning and science, you were my personal Miss Frizzle

ACKNOWLEDGEMENTS

Thank you to all of the volunteers that participated in data collection.

I would like to thank Barbara Muller-Borer, PhD for allowing me to use her lab space during the completion of this research.

A special thanks to Robert J. Tanenberg, PhD for donating five glucose monitors to assist with the research.

To my thesis committee Jitka Virag, PhD, and Jianchu Yao, PhD thank you for agreeing to be a part of my committee and spending time reading and reviewing the research.

Last, but certainly not least, I would like to extend a very special thanks to my friend and mentor Ricky Castles, PhD who, without his help, none of this would have come to fruition.

Contents:

List of Tables	x
List of Figures	xi
Chapter 1: Introduction	1
Types of Diabetes Mellitus	2
Blood Glucose Ranges for People With and Without Diabetes	3
Complications of Diabetes Mellitus	4
Glucose Monitor	5
EEG Technology	5
EEG Headset	7
Summary	8
Chapter 2: Literature Review	9
Complications of Unregulated Diabetes Mellitus	9
Hyperglycemia	9
Hypoglycemia	9
Diabetic Neuropathy	10
Vision Problems	10
Diabetic Ulcers	11
Monitoring of Diabetes Mellitus	11
Blood Glucose Monitor	11

Glucometer Accuracy	13
Google's Glucose Glass	14
Diabetic Breath Monitor	14
Previous EEG and Glucose Studies	15
Limitations of Previous Research	16
Brain Activity Frequency Ranges	18
The Lobes of the Brain and the Emotiv Channels	20
Artificial Neural Network	23
Example Network	25
Artificial Neural Networks in Biomedical Applications	32
Limitations with Artificial Neural Networks in Biomedical Applications	32
Chapter 3: Methodology	33
Hypothesis 1	33
Hypothesis 2	33
Experimental Setup	33
Code and Software Being Utilized	35
Study Participants	38
Variables Gathered	38
Dependent Variable (Target)	38
Independent Variable (Input)	38

Other Information Gathered	39
Evolution of Protocol	39
Initial Protocol	39
Protocol Amendment	41
Chapter 4: Results	45
Evaluating the Channels and Frequency Spectrum	45
Determining Features that have a Significant Correlation with BGL	47
Data Normalization per Participant	52
Method for Determining Number of Neurons	56
Overview of Code BGL Code for Number of Hidden Neuron Determination	56
Training Neural Network on Two Hours of Participant's Data	57
Predicting Third Hour of Participant's Data	57
Chapter 5: Discussion	62
Challenges with Brain Activity Signal	62
Challenges with Emotiv Hardware	63
Challenges with Training ANN	64
Comments on Neural Network Results	65
Chapter 6: Conclusion	67
Future Work	68
References	70

APPENDIX A: IRB Approval Letters	73
APPENDIX B: Participant Surveys Taken Before Each Data Recording Session	80
APPENDIX C: Utilizing Brain Activity to Detect Blood Glucose Levels	104
BGL_Code_For_R_Coefficient_Neural_Networks_Training.m	150
BGL_Code_For_R_Number_Of_Hidden_Neuron_Determination.m	218
Graphs_Participant_X	224
APPENDIX D: Network Figures Depicting Training/Testing Graphs	227

List of Tables

Table 1: Four Blood Glucose States a Person with Diabetes experiences	3
Table 2: Example Neural Network Problem	25
Table 3: Normalized Example Neural Network Problem	25
Table 4: Blood Glucose Levels of Volunteer 1 during their first data recording session	40
Table 5: Volunteer 1 Visit 1 Measured BGL, with BGL at 15 mins = BGL at 0 mins	41
Table 6: Volunteer 3 Visit 2 Blood Glucose Chart with 4 Finger Sticks	43
Table 7: Values for where 2/3 of the Features were "Good" P6 and P8...	50
Table 8: Table Showing R Value for 50 ANN during Example Calculation	53
Table 9: Averaged Data for Each Participant Example Calculation	53
Table 10: Averaged Data Normalized For Participant 1 Example Calculation	54

List of Figures

Figure 1: Hypothetical Blood Glucose Level Chart Showing bands...	4
Figure 2: Freestyle Freedom Lite Blood Glucose Meter [7]	5
Figure 3: Example of International 10-20 Placement [8]	6
Figure 4: Emotiv EPOC Headset [12]	8
Figure 5: Instructions for the safe use of a Blood Glucose Meter [22].	12
Figure 6: International 10-20 EEG Grid, channels utilized...	16
Figure 7: Example Recording of Brain Activity,[32]	18
Figure 8: Different Frequency Bands for an EEG recording	19
Figure 9: Lobes of the Brain [36]	21
Figure 10: 14 Different Emotiv EPOC Channels and their Location	23
Figure 11: ANN with Two Inputs, Three Hidden Layers, and an Output Layer	24
Figure 12: ANN for Example Network Problem	26
Figure 13: ANN System in Matrix Form	27
Figure 14: Sigmoid Function	28
Figure 15: Hyperbolic Tangent Function	28
Figure 16: Linear Function	29
Figure 17: Example Showing Gradient Descent of the MSE	32
Figure 18: Volunteer Wearing Emotiv EPOC	34
Figure 19: Example of Raw and Filtered Data with DC Offset Removed	37
Figure 20: First Volunteer's Blood Glucose Levels during First Visit with Trend-line	40
Figure 21: Volunteer 1 Visit 1 with BGL at 15 minutes equal to BGL at 0 minutes	41
Figure 22: Volunteer 3 Visit 2 Blood Glucose Graph with Trend-line	43

Figure 23: BGL with all assumptions in place	44
Figure 24: Example of three, 1-hour, data collections concatenated into single recording	46
Figure 25: Example P-Values Participant 1 Channel AF4	48
Figure 26: Channels Used to Train ANN	51
Figure 27: Frequency and Channel BGL Correlation: Each Channel/Activity band...	55
Figure 37: Network with good following predicting data it had been trained on....	58
Figure 38: Network with good following predicting data it has never seen before	59
Figure 39 Network with poor following predicting data it had been trained on...	60
Figure 40: Network with poor following predicting data it has never seen before	61
Figure 41: Noisy Raw Data from Emotiv EPOC due to poor conduction	64

Chapter 1: Introduction

Diabetes Mellitus (DM) is a chronic disease that affects the lives of roughly 30 million Americans [1]. It is caused by the inability of the pancreas to produce adequate levels of insulin. Insulin, a hormone secreted by the pancreas, allows the body to utilize glucose, a simple sugar, for the energy necessary to perform bodily functions such as digestion. Glucose is the main source of energy for cellular function, and is the only source of energy supply for red blood cells and the brain. Without properly metabolizing glucose the body shuts down. Diabetes creates many health complications that require medical attention such as neuropathy, glaucoma, and non-healing ulcers. These complications can be avoided, or mitigated, with proper self-monitoring. However, many people with diabetes do not properly self-monitor for a variety of reasons including embarrassment, expense, and pain. Current methods of self-monitoring require blood samples to be collected, often from a finger-stick. This is an uncomfortable process and it is often necessary that it is performed multiple times per day. This lancing procedure increases the diabetic person's risk of contracting blood borne diseases, if they are not following proper safety precautions. Some examples of precautions that should be followed are not sharing lancing equipment, not reusing lancets, or not sharing meters; however due to cost some people with diabetes do not follow such guidelines. The focus of this research is to mitigate pain, and reduce exposure to blood borne diseases by eliminating the need to collect blood samples in order to test glucose levels.

As a proposed method to rectify the shortcomings of conventional blood glucose monitoring it may be possible to monitor electrical activity on the scalp through the use of an electroencephalograph (EEG) to non-invasively monitor a person's blood glucose level (BGL). This correlation between brain activity and glucose is possible because the voltage fluctuations

measured by the EEG are caused by the ionic current flow of neuronal cells. Ionic current flow is directly related to the neuronal cells' metabolism of glucose, and a failure or change in glucose supply will cause early changes in EEG signal [2-5]. It is these changes that have been examined and monitored in order to develop an artificial neural network capable of predicting a person's BGL in a standard form, such as mg/dL or mmol/L based on brain activity. The unit mg/dL is typically used within the United States, whereas mmol/L is used everywhere else in the world, Europe, Asia, etc. The conversion between these two units of measurement is 1 mmol/L is approximately equal to 18.02 mg/dL. For the purpose of this research, mg/dL will be utilized. This neural network is a regression network as opposed to a classification network. A regression network is one that will return a numerical value such as 100 mg/dL. whereas a classification network is one that returns a statement about the input, such as this data indicates that a person is in a normal glycemic state, or this person is in a hyperglycemic state, etc.

Types of Diabetes Mellitus

DM can be broken into two subgroups Type 1 DM (T1DM) and Type 2 DM (T2DM).

Type 1 Diabetes Mellitus: T1DM is commonly referred to as hereditary diabetes and accounts for 5% of diabetes cases in North America, or 1.5 million cases. T1DM is typically diagnosed while the individual is a child or young adult. T1DM is an autoimmune disorder where the beta pancreas cells, the cells that produce insulin, are attacked and destroyed by a person's immune system. This causes the body to become incapable of producing insulin.

Type 2 Diabetes Mellitus: T2DM is known as lifestyle diabetes, and accounts for 95% of diabetes cases in North America or 27.5 million cases [1]. Individuals suffering from T2DM tend to be overweight or obese with a body mass index greater than 25, and middle to late aged, older

than 40 years old. This type of diabetes is a progressive disease, meaning that the body gradually becomes less capable of producing insulin. The major risk factors for T2DM are obesity, age, poor diet, and sedentary lifestyle.

Blood Glucose Ranges for People With and Without Diabetes

It is important to preface that every individual case is different and that these values are generalized numbers.

Non-Diabetic: A typical range for a non-diabetic person is a BGL of approximately 70-100mg/dL. After a meal the value can rise upwards of 140mg/dL, however, this value typically returns to normal after 1-2 hours.

Diabetic: A person with diabetes' BGL is much more difficult to generalize than someone living without diabetes. Their BGL can experience four separate states, hypoglycemia, normal, elevated, and hyperglycemia, as depicted in Table 1.

Table 1: Four Blood Glucose States a Person with Diabetes Experiences

Hypoglycemia:	BGL less than 70mg/dL
Normal:	BGL between 70 – 140 mg/dL
Elevated:	BGL between 140-180 mg/dL
Hyperglycemia:	BGL greater than 180mg/dL

After a meal, a person with diabetes may have their BGL rise upwards of 160 mg/dL. In addition it may take as long as 6 hours to return to normal levels, or the BGL may not return to normal without an injection of insulin.

Complications of Diabetes Mellitus

Unregulated DM can cause severe health issues requiring medical attention. Some severe cases may result in death due to the body's mishandling of its blood glucose level. These complications can be broken into two mutually exclusive glucose states, hyperglycemia and hypoglycemia. Figure 1 shows an example of what an individual's BGL chart may look like. It should be noted that it is not uncommon for a diabetic individual to experience states of elevated and hyperglycemic BGLs within the same day. However, ventures past the normal bounds, 70 mg/dL and 140 mg/dL, should be limited in number, amplitude, and duration. For instance, in elevated and hyperglycemia these spikes cause an increase in glycosylated hemoglobin, hemoglobin with a bound glucose molecule, concentration within the blood. This elevation of glycosylated hemoglobin has been linked to long term diabetes complications, such as neuropathy and cardiovascular disease. [6].

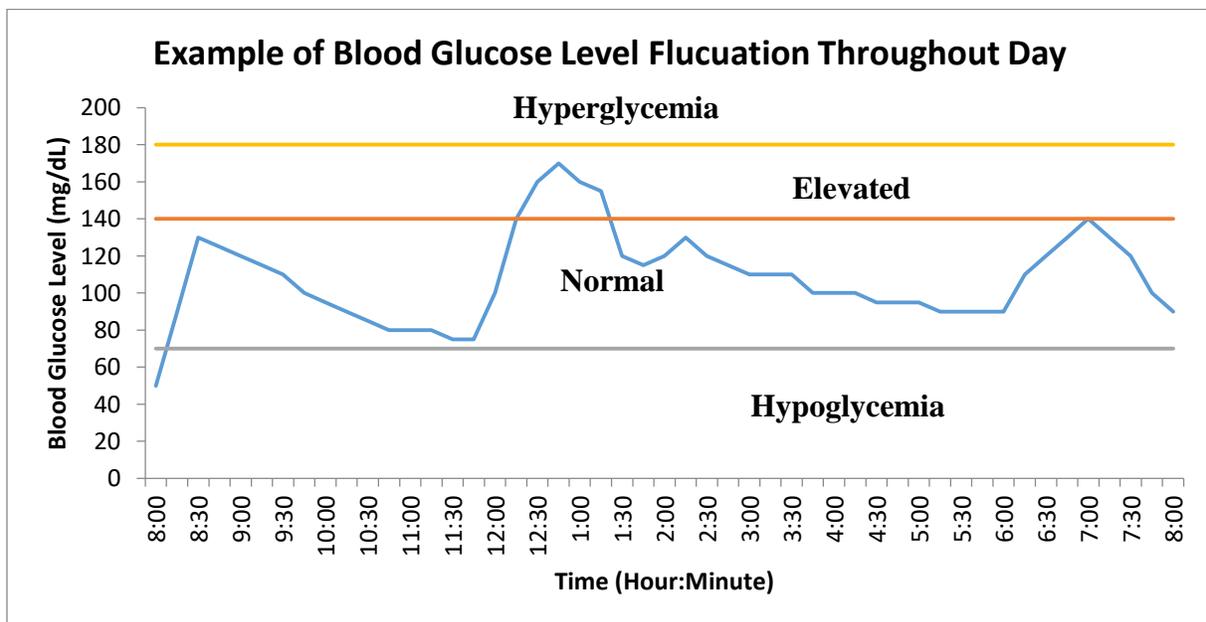


Figure 1: Hypothetical Blood Glucose Level Chart Showing bands for Hypoglycemia, Hyperglycemia, Elevated and normal range.

Glucose Monitor

This research collected data from a standard glucometer in addition to collecting EEG data to train an artificial neural network. A standard finger-stick glucose monitor is utilized due to its accessibility and it being the primary device that people living with diabetes use to self-monitor. The model of glucose meter that each participant was given is the Freestyle Freedom Lite, depicted in Figure 2. The Freestyle Freedom Lite is a basic home-use glucometer capable of taking recordings from multiple testing sites, including finger stick and palm stick, it requires a 0.3 μL blood sample for testing, and can test as often as once per minute.



Figure 2: Freestyle Freedom Lite Blood Glucose Meter [7]

EEG Technology

An EEG measures the changes in voltage fluctuation directly from the scalp. This is an aggregate signal that can be recorded by 1 – 100+ sensors. The measured signal is typically between 10 – 100 μV , with a frequency between 1 – 40 Hz. The genesis of this electrical activity is ionic

current fluctuations caused by neuron excitation. Clinically, EEGs have been used in the diagnosis and treatment of disorders including epilepsy, sleep disorders, migraines, and monitoring intensive care patients. In order to ensure reproducible results from EEG, scientists have agreed to adhere to the International 10-20 system, see Figure 3, as a reference for EEG electrode placement. The 10-20 system is the distance between the placements of the electrodes, i.e. each of the electrodes takes up 10 to 20% of the front, rear, right, or left side of the skull. These locations correspond to different underlying regions of the brain.

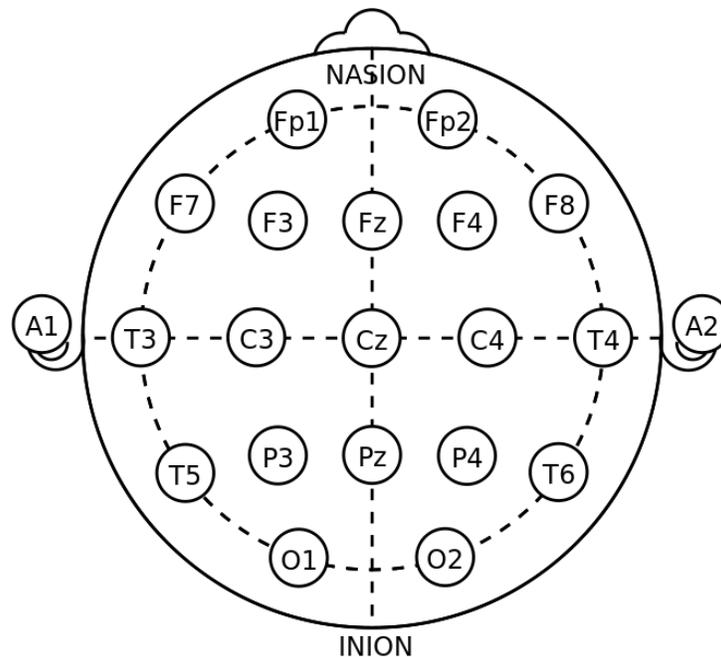


Figure 3: Example of International 10-20 Placement [8]

These regions, or lobes, are the frontal, temporal, central (not an actual brain lobe), parietal and occipital and are abbreviated using F, T, C, P, and O. These different lobes are linked to different brain activity. For instance, the frontal lobe is associated with planning, and retention, but the temporal lobe is associated with hearing, deep memory, and language processing. [9-11].

EEG Headset

The headset that is used in this research is the Emotiv EPOC headset, depicted in Figure 4. The EPOC headset is a non-invasive recording device that is Bluetooth compatible, rechargeable, and contains 14 channels sampling at a rate of 128 Hz [12]. While most EEG headsets require wired electrodes, the Emotiv EPOC is a wireless device. This allows study participants to move more freely during testing, making them more comfortable by allowing for more natural movements of the head and neck. In addition, this particular EEG does not require an electrolytic gel solution for the skin electrode interface; rather it uses a saline solution on felt pads for conduction. Some disadvantages of the Emotiv EPOC are that the headset plastic may become stretched out between participants; this makes it difficult to secure a stable connection for all participants. Another disadvantage is that it may become uncomfortable to wear after an extended period of time, prompting the wearer to shift it on their head and move an increased amount, thus causing disturbances in the recording signal.



Figure 4: Emotiv EPOC Headset [12]

Summary

Diabetes Mellitus is a condition where the body does not effectively utilize glucose. Without proper management DM causes potentially severe medical emergencies and is currently placing a large financial burden on the United States' healthcare system. The goal of this research is to non-invasively explore the link between blood glucose levels and brain electrical activity via an electroencephalograph. This is done in order to determine if EEG has the potential to be a non-invasive blood glucose measurement device.

Chapter 2: Literature Review

Complications of Unregulated Diabetes Mellitus

Hyperglycemia

Hyperglycemia, or excess glucose, is typically diagnosed when an individual's BGL is greater than 200mg/dL. Hyperglycemia occurs when there is not enough insulin for the body to effectively utilize glucose, this causes the body to starve for energy and resort to burning body fat for fuel. Burning body fat produces a chemical byproduct known as a ketone, a carbon double bonded to an oxygen with two R groups ($RC(=O)R$). Large amounts of ketones in the blood are damaging to body organs, therefore the renal system removes ketones through urination. However, the renal system cannot remove ketones from the blood fast enough in a hyperglycemic state and untreated hyperglycemia will result in an individual going into a diabetic coma, or ketoacidosis, and if left unchecked may result in death [13].

Hypoglycemia

On the other end of the glucose spectrum is hypoglycemia, or too little glucose, which is typically diagnosed when an individual's BGL is less than 70 mg/dL [14]. Hypoglycemia can be caused by a variety of factors such as fasting, vigorous exercise, and using too much insulin [15]. Due to the lack of glucose, body functions begin to slow and shut down. The individual may begin to experience symptoms such as lethargy, anxiety, irritability, confusion, in addition to headache and nausea. If left untreated an individual experiencing hypoglycemia may eventually experience seizure, coma, and death [16].

Diabetic Neuropathy

Diabetic neuropathy is a generic term for nerve damage that is caused by diabetes. While it is known to be a complication of diabetes that develops over time, it generally occurs due to elevated blood glucose levels, or hyperglycemia. This nerve damage affects approximately half of all people who have diabetes. The nerve damage is not limited to one specific area as all nerves can be affected, however feet and legs are the most commonly affected. This causes complications such as pain in legs and feet eventually giving way to numbness. Due to the lack of sensation in legs and feet, this causes sufferers of diabetic neuropathy to not notice when they have stepped on something sharp or have a wound on their feet. This often leads people with diabetes to develop diabetic sores that do not properly heal and may result in amputation. Other areas of the body that are also affected by diabetic neuropathy include hands, digestive system, gonads, blood vessels, and heart [17].

Vision Problems

Other complications of hyperglycemia are vision problems such as glaucoma, swelling of the lens, and cataracts. Diabetic glaucoma is caused by fluid improperly draining from the eye. This improper fluid draining causes an increase in fluid pressure within the eye and damages nerves and blood vessels within the eye causing vision loss, and in some severe cases, blindness. A less severe complication of hyperglycemia is swelling of the lens of the eye. While the onset of this is often sudden, it does not often cause permanent damage as long as BGLs are returned to proper levels. Another complication of the eye lens is cataracts. A cataract is the fogging of the eye lens and can often be remedied with surgery. Anyone can get cataracts; however, people with diabetes often develop cataracts at a younger age than people without diabetes [18, 19].

Diabetic Ulcers

A diabetic ulcer is a lesion, most commonly in the foot. These lesions are slow to heal and are often prone to infection. If treatment of these lesions is delayed it can cause necrosis of the surrounding tissue. It is these complications that make diabetic foot ulcers the leading cause for non-traumatic injury amputation in the United States. Approximately 5% of people with diabetes will experience a diabetic ulcer, and 1% of these cases require amputation every year. Since 29 million American suffer from diabetes, this means approximately 1.5 million cases of diabetic ulcers and 14.5 thousand amputations occur annually [20, 21].

Monitoring of Diabetes Mellitus

With these potentially life-threatening complications, it is imperative for diabetic individuals to monitor and be aware of their BGL. Due to the frequency that these measurements need to be taken it is not feasible for a medically trained professional to constantly take these measurements. Therefore, diabetic individuals are taught to self-monitor their BGL.

Blood Glucose Monitor

The currently accepted method for BGL self-testing is through the use of a Blood Glucose Meter (BGM). These are commercially available items; however they are often given to patients by health care providers once they are diagnosed with diabetes, along with instructions for their safe use. Figure 5 shows generic instructions for the safe use of a BGM. In order to test BGL with a BGM the individual typically lances their finger, although there are other acceptable locations, with a lancet drawing a small quantity of blood and placing it on a test strip. This blood is then analyzed by the BGM and the person's BGL is quantified. This procedure is uncomfortable, and has led many diabetic individuals to be noncompliant with monitoring their blood glucose level.

The frequency that the person with diabetes monitors their BGL varies and is typically established in consultation with a healthcare provider. Some people with T1DM monitor their BGL over 8 times per day. Due to the difficulty in obtaining accurate self-monitoring information making an estimate on how many diabetic individuals are noncompliant is inherently problematic. However, in 2012 the United States Center for Disease Control (CDC) estimated that the cost of diabetes in the United States was \$245 billion [1].

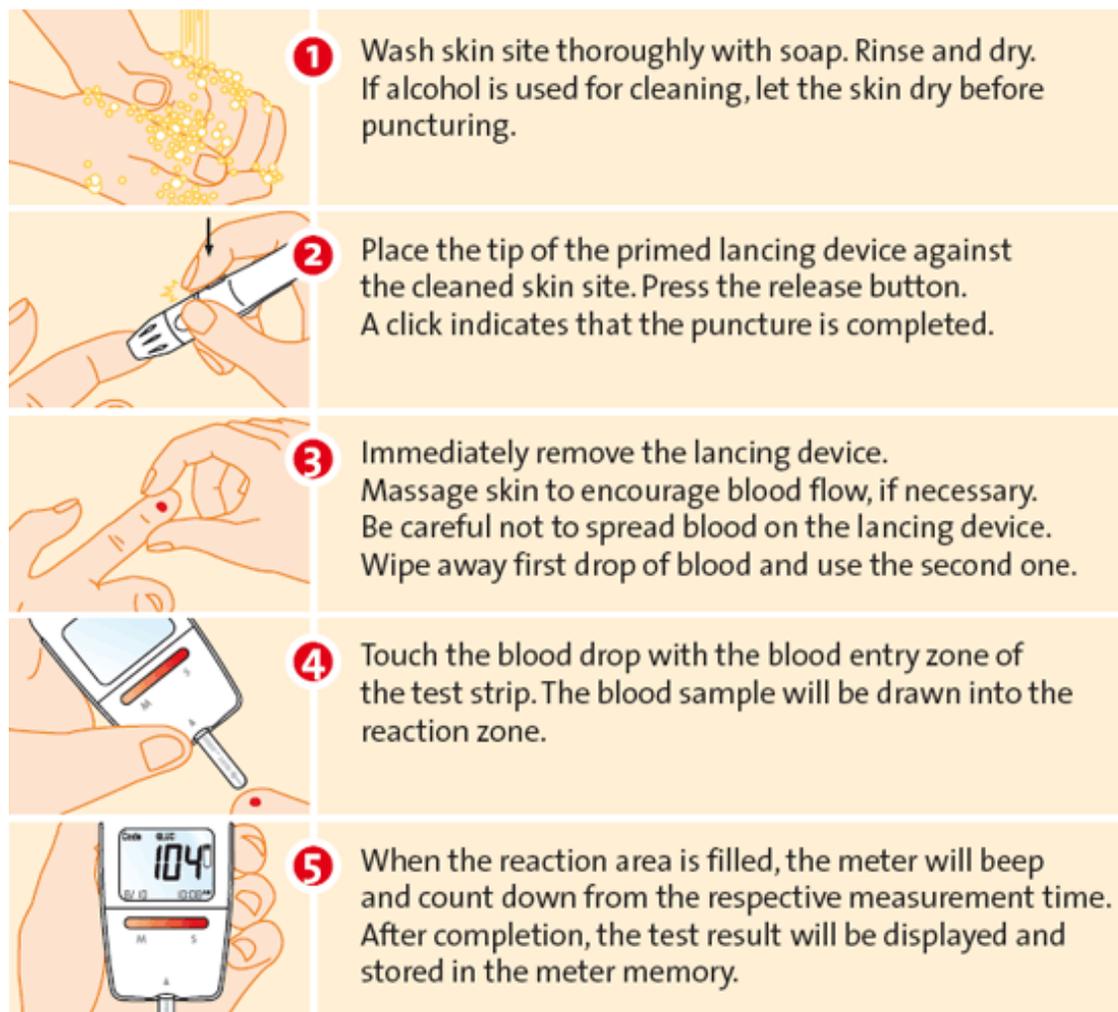


Figure 5: Instructions for the safe use of a Blood Glucose Meter [22].

Glucometer Accuracy

The accuracy of home use glucometers is a source of much debate between industry officials, medical caregivers, the Food and Drug Administration (FDA), and patients. This is because the standard at home glucometer can have an error range of 15-20%, 95% of the time, when measuring above 80mg/dL. This means that a BGL reading of 100 mg/dL can correspond to a true BGL anywhere between 80 mg/dL and 120 mg/dL [23]. There are many factors that can impact meter accuracy that a user has direct control over, such as hand washing, leaving the meter in a too hot/cold environment, test strips expiring, improperly loading strips, etc. Meter accuracy remains very important subject for people who use it on a daily basis to keep their BGLs in check [24]. In 2010 the FDA held a 2-day hearing to discuss the accuracy of home use glucose meters. After the meeting Courtney Lias of the FDA issued this statement

"The technology isn't currently sufficient for manufacturers to meet +/-10% accuracy at the huge volume of lay use, but for the smaller volume of tests done in clinical settings, they can... In an ideal world, everyone would have a meter that's within 5-10% of the accuracy reference, but as it stands right now, if we called for that, manufacturers wouldn't be able to provide meters at all. [25]

A major issue with meter accuracy is that despite glucometers not being FDA approved for monitoring diabetes, they are routinely used. This means that people are making decisions based on a meter that may, or may not be accurate, and in best case scenarios, is only within 20% of their true glucose reading, even more for outlier readings. This is problematic particularly for people who are insulin dependent. For example, a user takes their BGL and the monitor inaccurately reports that the BGL is 147mg/dL, while their true BGL is around 117. The user then proceeds to slightly overdose with insulin, causing the glucose levels to become too low. A

few hours later, when the user checks again, the reading shows 60 mg/dL, the user then eats something to raise their blood sugar, repeating the cycle and causing the BGL to ping-pong between low and elevated blood glucose levels despite the individual attempting to tightly control their BGL. This is further complicated by the fact that any device that is calibrated through the use of a glucometer, including the artificial neural networks (ANNs) used in this research, has a greater potential for error than the device that calibrated it. For example, suppose someone takes a ruler and measures out an inch on a sheet of paper. That person then uses the drawn inch to measure out another inch; the ruler has an error percentage associated with it. Then, the drawn ruler has an error stacked on top of the ruler that drew it, and the second drawn ruler has a stacked error associated with it.

Google's Glucose Glass

An experimental method for monitoring BGL is through the use of Google Glass technology. The "Glucose Glass" is intended to be a continuous monitoring device without the use of lancets. The contact lenses that are being developed by Google utilize small embedded wireless sensors in order to take the readings from the individual's tears [26, 27]. There are, however, issues with the contact lens design. This is because it is generally not advised for diabetic individuals to wear contacts due to being susceptible to developing ocular complications [28].

Diabetic Breath Monitor

Another experimental method for monitoring BGL is through the use of a diabetic breathalyzer. Similar to an alcoholic breathalyzer the person with diabetes breathes into the device and receives a recording of their blood glucose level. The device works similarly to a traditional breathalyzer by using infrared beams that are absorbed by methyl groups, including acetone. Acetone is released in more abundance by people with diabetes due to it being one of the

ketones. However, this device also has its limitations; people with diabetes have had issues receiving false positives with a traditional breathalyzer due to the nature of the breathalyzer measuring methyl groups. Therefore, it is plausible that using a glucose breathalyzer may not result in repeatable and accurate measurement [29, 30].

Previous EEG and Glucose Studies

While utilizing a BGM remains the standard method for determining BGL, a variety of research in non-invasive monitoring of nocturnal hypoglycemia has been underway since at least 1997. These early studies often recorded a variety of physiological signals such as an individual's sweating, snoring, heart rate, and electroencephalogram (EEG) to obtain an index of hypoglycemia [31]. These methods relied on fuzzy logic to determine whether a sleeping individual was experiencing hypoglycemia. These signals have since been weaned down to monitoring only a few EEG channels. The EEG channels that are monitored for hypoglycemia are O1, O2, C3, and C4 according to the International 10-20 system, these channels are referenced to A1 and A2, as depicted in Figure 6.

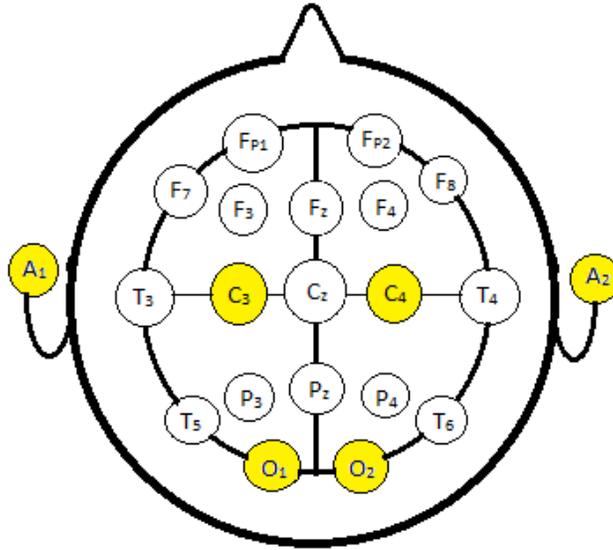


Figure 6: International 10-20 EEG Grid, channels utilized in Nguyen's study are highlighted in yellow.

Limitations of Previous Research:

The experiments [2-5] Nguyen *et al.* performed showed that EEG data combined with signal processing and computer modeling is capable of determining if someone with T1DM was experiencing nocturnal hypoglycemia. Nguyen was successful in training an ANN classifier to show that a person was experiencing hypoglycemia. They accomplished this by training a feedforward ANN using undisclosed features from the EEG. This ANN then classified the data into normal glucose range or hypoglycemia. While this research is outstanding in its own merits, the primary issue is that it does nothing to avert the problem until someone is already in a hypoglycemic state.

The research done by this project seeks to expand upon this by monitoring the BGL and brain activity of healthy volunteers, who are awake at the time of recording, then developing an artificial neural network that will predict the volunteer's BGL based upon their brain activity.

This is important for several reasons: monitoring healthy volunteers whose BGL will be in the normal range can help determine the sensitivity of this regression technique. For example, can a

network be trained to determine the difference between two normal glucose values, 100 mg/dL and 110 mg/dL? Many people living with diabetes tend to have elevated blood sugar levels rather than low blood sugar, due to natural body resistance to insulin, and being intentionally under-dosed with insulin. While hypoglycemia is a medical emergency and can lead to coma and death, elevated blood glucose levels are associated with the chronic, long term, complications of diabetes such as neuropathy. If the trained network has a high degree of sensitivity and specificity it could be used to develop a feedback control mechanism for an artificial pancreas that would be capable of more tightly regulating BGL with insulin, without the fear of overdose. Finally, the research performed by Nguyen *et al.* was performed on sleeping volunteers. While, nocturnal hypoglycemia is a very serious concern, particularly with young children, it does not account for a person's time while awake. Brain activity, reduces frequency and increases amplitude while asleep, this may have made the data collected by Nguyen *et al.* simpler to analyze.

Brain Activity Frequency Ranges

Brain activity normally ranges between 0.5 Hz and 40 Hz. An example of an EEG recording is shown in Figure 7.



Figure 7: Example Recording of Brain Activity.[32]

This recording can be separated into five distinct frequency bands. These bands are delta, theta, alpha, beta, and gamma. Each band can be attributed to certain activities [26]. For example, delta waves are typically associated with sleep, whereas gamma is associated with focus and intense cognitive thought; Figure 8 shows an example of what each band may look like. Based upon Nguyen's studies, the expectation is that changes in frequency and amplitude of alpha, beta, and theta ranges will play the most significant role in predicting BGL based off of brain activity [2-5, 33].

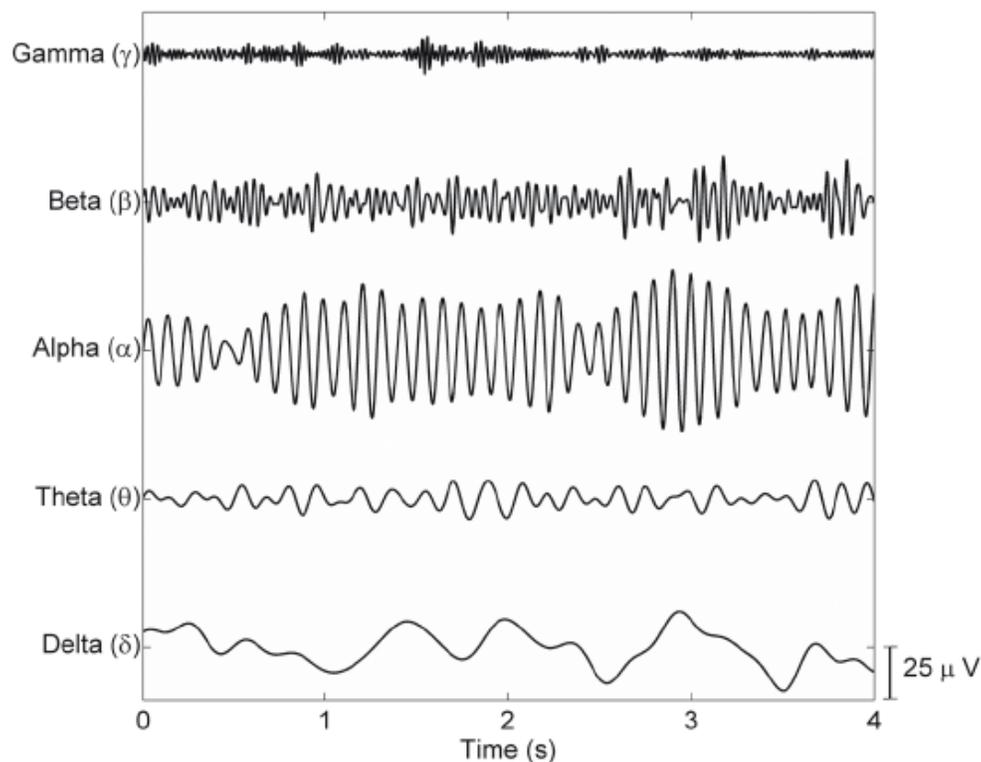


Figure 8: Different Frequency Bands for an EEG recording

Gamma (γ): The highest frequency range is gamma, 31+ Hz. Gamma displays the highest frequency and lowest amplitude of the frequency ranges and is associated with memory recognition.

Beta (β): The second highest frequency range is beta 16-31 Hz. Beta brainwaves are observed during active thinking, increased alertness, stress, or anxiety.

Alpha (α): Alpha frequencies occur between 8 and 15 Hz. These brainwaves are observed most prominently during relaxation. Alpha brainwaves are also associated with coma.

Theta (θ): Theta brainwaves occur between 3 Hz to 8 Hz. Theta waves are observed most prominently during sleep. The sleep associated with theta brainwaves is considered to be a less deep sleep than delta. However, theta brainwaves are associated with dreaming and drowsiness. In addition the theta waves are observed during metabolic encephalopathy.

Delta (δ): Delta brainwaves are the lowest frequency brainwave; however, they have the highest amplitude. The frequency of delta brainwaves is between $\frac{1}{2}$ Hz to 3 Hz. They are observed most prominently during deep sleep, but it is always visible. Pathologically delta brainwaves are associated with metabolic dysfunction and hydrocephalus.

The Lobes of the Brain and the Emotiv Channels

The Emotiv EPOC is a 14-channel EEG recording device; each of these 14 channels corresponds to different locations on the brain, and those locations correspond to the 4 different lobes of the forebrain.

Lobes of the brain: There are four major lobes of the brain and while there is a high degree of plasticity for these regions each region is typically associated with a particular function. The four lobes are the frontal, parietal, occipital, and temporal. Figure 9 shows the lobes and their associated locations.

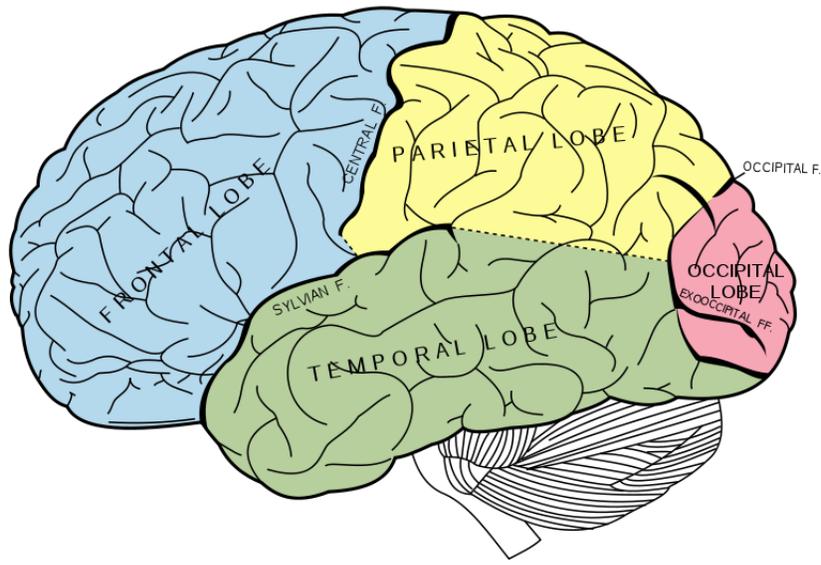


Figure 9: Lobes of the Brain [34]

Frontal Lobe: The foremost lobe, it is associated with a broad range of functions such as judgement, motor control, and emotions. It is the lobe of the brain that is associated with personality and sense of self. There are debatable differences between the left and right hemisphere of the brain. The left hemisphere is said to be more responsible for controlling language and movement, and the right hemisphere is said to control non-verbal abilities. However, this is not always the case and these functions are not exclusive to one hemisphere or the other. [35]

Parietal Lobe: The parietal lobe is centrally located within the brain and is responsible for integrating all the different senses into perception. In addition to this, it also provides the body with a sense of spatial mapping or awareness. A simple example of this would be if you had a drink sitting on a table in front of you, then someone asked you to close your eyes and pick up the drink, you would be capable of knowing the approximate location of the drink based upon the mental map of your surroundings. [36]

Occipital Lobe

The occipital lobe is located towards the rear base of the skull. This lobe of the brain is responsible for processing visual stimulation.

Temporal Lobe

The temporal lobe is located near the left and right ear. It is the auditory integration center and is responsible for processing hearing. In addition to hearing the temporal lobe is also associated with understanding language, and deep memory.

The 14 Channels Recorded by the Emotiv EPOC:

While all brain activity measured by an EEG is an aggregate signal measured from the scalp, the different channels an EEG utilizes enables the user to more closely observe specific areas of the brain. For example, the signal recorded in the frontal lobe and the occipital lobe will appear extremely similar, but a visual stimulus may show more strongly, increased amplitude, in the channel located closer to the occipital lobe. To do this, the Emotiv EPOC utilizes 14 different channels scattered over the scalp. Figure 10 shows the locations of the channels in relationship to the scalp.

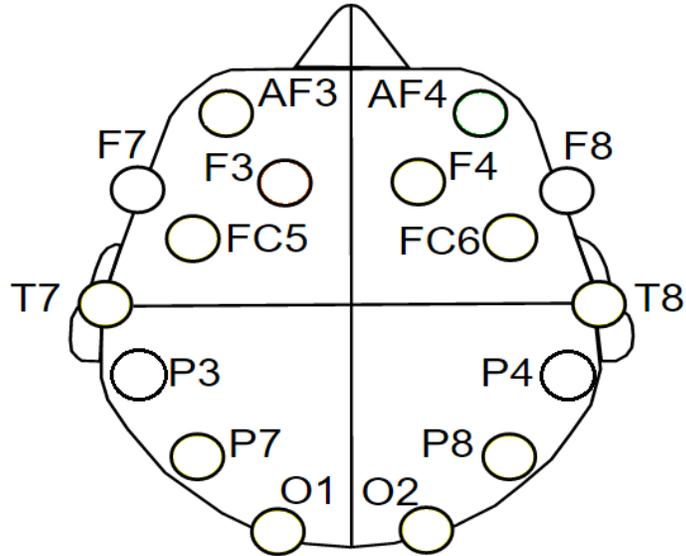


Figure 10: 14 Different Emotiv EPOC Channels and their Location

Artificial Neural Network

An ANN is a form of machine learning that attempts to mimic a biological neural network. This is achieved by attempting to mimic the response of biological neurons, by weighting information coming from previous neurons. Most artificial neural networks can be broken into three separate layers called the input layer, hidden layer, and output layer. The hidden layers contain a number of neurons, and each neuron may be independent of the previous hidden layer. Figure 11 shows an example of a feedforward neural network with 3 hidden layers, the first containing 5 neurons, second 7, and third 3.

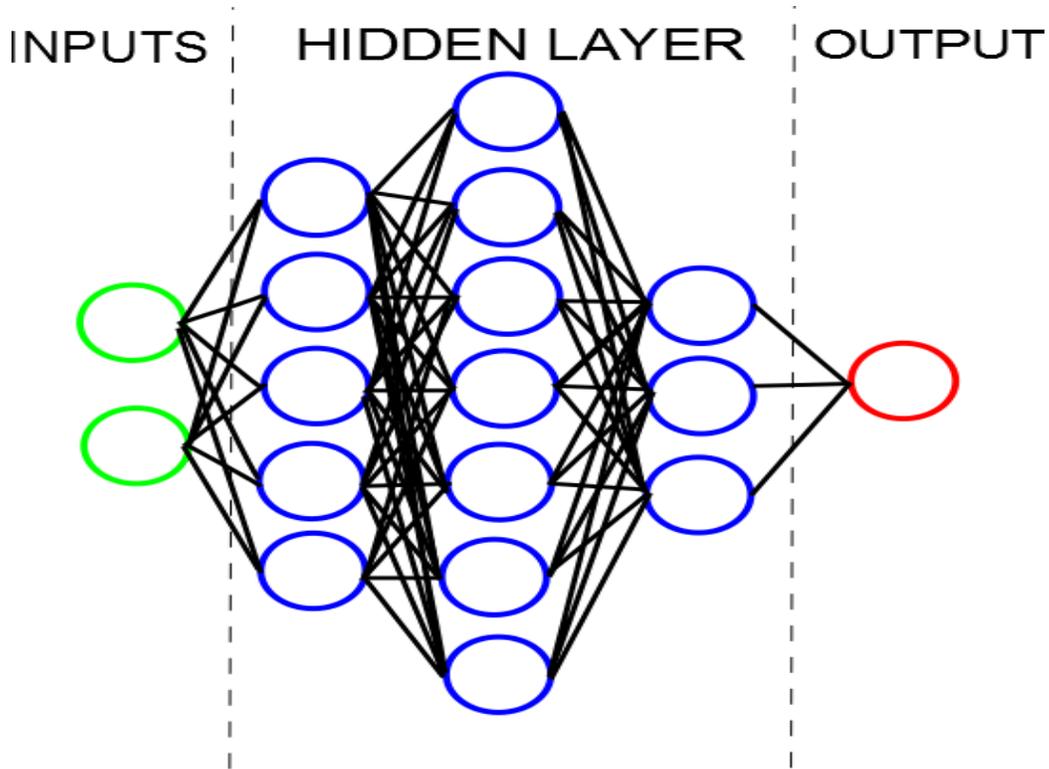


Figure 11: ANN with Two Inputs, Three Hidden Layers, and an Output Layer

The end result is a model that has learned to generalize data and predict, model, or interpret an input data set for an unknown output. In the case of this research, the inputs are the features extracted from the EEG, and the desired outputs are the BGL. Neural networks accomplish generalization through training. This is a process that involves taking the inputs, weighting them, passing them to the hidden nodes, where a simple mathematical operation is performed; this “solution” is then passed into the next layer. Once the process has been completed, the output error is calculated by comparing the output from the neural network, to the output that is given in the training data set. The network then adjusts the weights accordingly and tries again. This process can converge very quickly, or can take several thousand attempts.

Example Network

A simple example of a neural network problem could be to predict how many flowers bloom in a garden based upon the temperature, and amount of rain. Table 2 shows some example data where temperature and inches of rain are the inputs, which will be used to predict the target of number of flowers in bloom.

Table 2: Example Neural Network Problem

Temperature (F)	Inches of Rain	Flowers Bloom (max of 10)
70	3	4
83	6	8
90	1	2
91	10	7

So that everything is on an equal playing field, the variables are all divided by the maximum of each variable in order to normalize data. Table 3 shows the normalized Table.

Table 3: Normalized Example Neural Network Problem

Normalized Temperature	Normalized amount of Rain	Normalized # of Flowers Bloom (max of 10)
0.77	0.3	0.4
0.91	0.6	0.8
0.99	0.1	0.2
1	1	0.7

A potential neural net that will solve this problem is a network with an input layer, 3 hidden neurons in the hidden layer, and an output layer, as depicted in Figure 12.

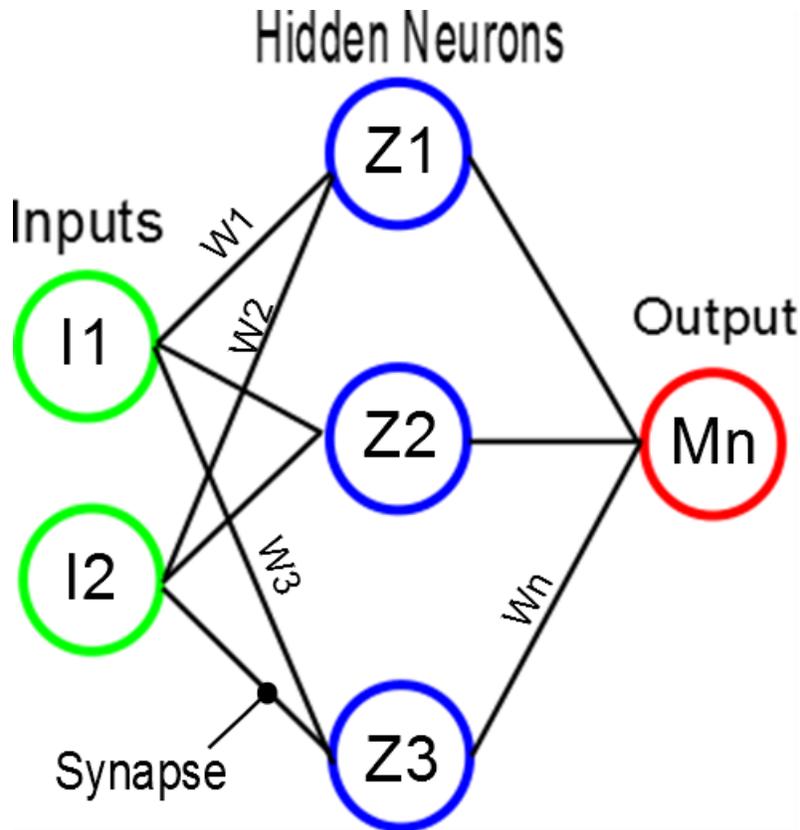


Figure 12: ANN for Example Network Problem

The inputs, such as temperature, and amount of rain, are passed through the synapses, represented by lines, to each of the neurons. The synapses are randomly seeded with weights that multiply the input variable, I_n , by each weight, W_n . As shown in Eq. 1. In the example calculation below $I_1 = 3$ is a randomly chosen weight. Each weighted input, $I_n W_n$, is then summed up by the neuron to become Z_n and passed through an activation function, f .

$$\text{Weighted Input} = I_1 * W_1 \tag{Eq. 1}$$

$$\text{Weighted Input} = 0.77 * 3 = 2.31$$

The neuron will then sum all of the weighted inputs from each of the synapses, Eq. 2. In the case of this example temperature and amount of rain will be summed at each neuron. W_2 was randomly chosen to be 5

$$Z_1 = I_1W_1 + I_2W_2 \quad (\text{Eq. 2})$$

$$Z_1 = 2.31 + 1.5 = 3.81$$

The entire system to this point can be expressed in matrix form as follows in Figure 13.

Inputs	
0.77	0.30
0.91	0.60
0.99	0.10
1.00	1.00

Weights		
W1	W2	W3
W5	W6	W7

Weighted Inputs Going to Output		
$0.77*W1 + 0.30*W5$	$0.77*W2 + 0.30*W6$	$0.77*W3 + 0.30*W7$
$0.91*W1 + 0.60*W5$	$0.91*W2 + 0.60*W6$	$0.91*W3 + 0.60*W7$
$0.99*W1 + 0.10*W5$	$0.99*W2 + 0.10*W6$	$0.99*W3 + 0.10*W7$
$1.00*W1 + 1.00*W5$	$1.00*W2 + 1.00*W6$	$1.00*W3 + 1.00*W7$

Figure 13: ANN System in Matrix Form

This Z_1 value is then passed through an activation function, f , and becomes the value A_n , Eq. 3. The role of the activation function is to limit the output of the network to a value typically between 0 and 1 or -1 and 1.

$$A_n = f(Z_n) \quad (\text{Eq. 3})$$

There are several different types of activation functions, some of the more common are sigmoid, hyperbolic tangent, and linear activation functions, shown in Figure 14 - Figure 16. For this example hyperbolic tangent will be used as f .

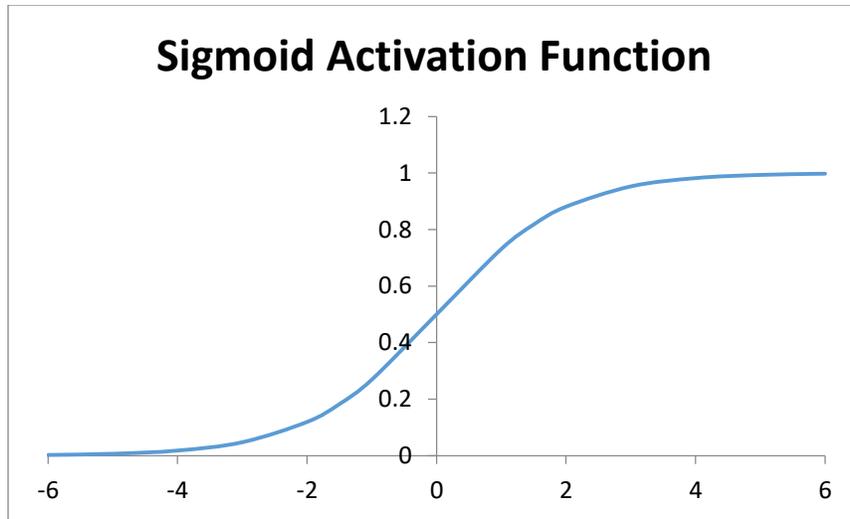


Figure 14: Sigmoid Function

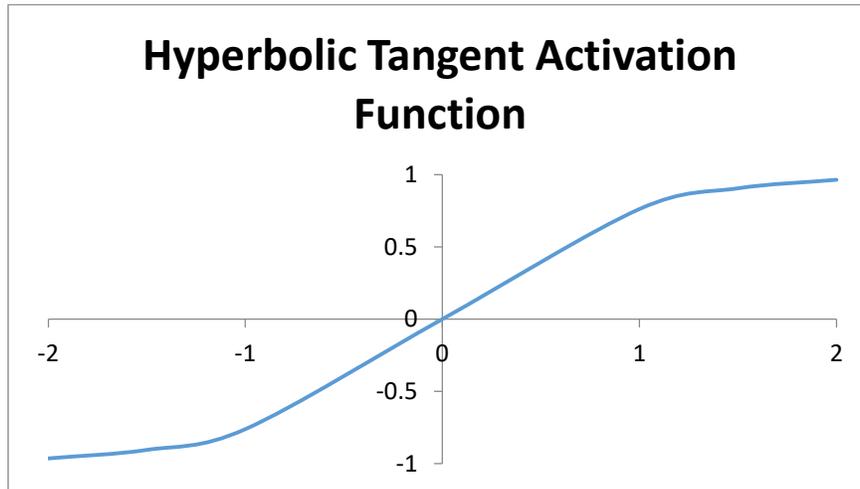


Figure 15: Hyperbolic Tangent Function

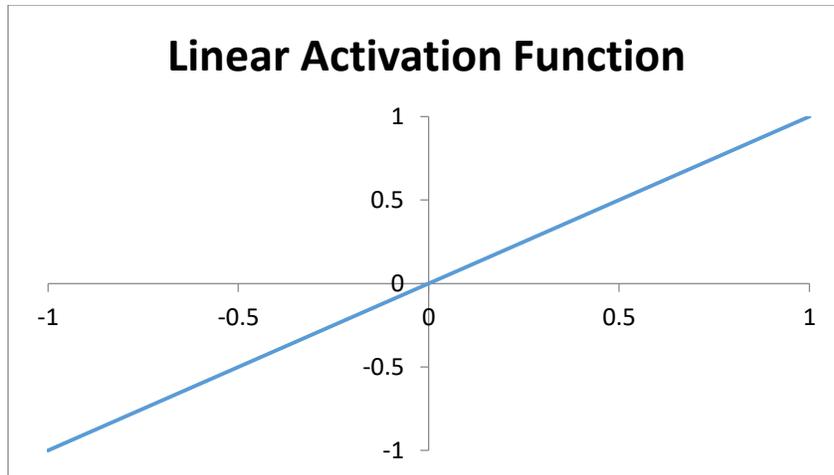


Figure 16: Linear Function

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

In the context of this problem:

$$A_n = \tanh(Z_n) = \frac{e^{z_n} - e^{-z_n}}{e^{z_n} + e^{-z_n}}$$

Calculating the value that will be passed to the output layer:

$$A_n = \frac{e^{3.81} - e^{-3.81}}{e^{3.81} + e^{-3.81}} = 0.99$$

After this has been repeated for all weighted inputs, it is passed to the next hidden layer and multiplied by the next synapse weights, or if no hidden layers are left, to the output layer, and multiplied by the remaining synapses. In the output layer, all final weighted neuronal inputs are passed through another activation function. This activation function may be the same as the previous activation functions, or may be different. The activation function for this example will be the sigmoid function. For this example, the summed values going to the output will be M_n , Eq.

4.

$$M_1 = A_1W_1 + A_2W_2 + A_3W_3 \quad (\text{Eq. 4})$$

These values are passed through the output activation function, F, for result t_n , Eq. 5.

$$t_n = F(M_n) \quad (\text{Eq. 5})$$

$$\text{Sigmoid} = \frac{1}{1 + e^{-t}}$$

$$t_n = \frac{1}{1 + e^{-M_n}}$$

For the sake of this example let's assume that the activation function returned a value of 0.83.

This value needs to be re-multiplied by the value used to normalize the outputs. In this problem there were a maximum of 10 flower blooms possible. Therefore:

$$0.83 * \text{Max Flowers (10)} = 8.3 \text{ Flowers}$$

However, let's also assume that the target number of flowers is actually 5, and that the network over estimated the number of flowers that should be returned, Eq. 6. The error in this example is -3.3.

$$\text{Error: } e_n = \text{Target} - \text{Estimate} \quad (\text{Eq. 6})$$

$$e_1 = 5 \text{ Flowers} - 8.3 \text{ Flowers} = -3.3 \text{ Flowers}$$

To compensate for this error the algorithm attempts to minimize the error, or "train the network" which is the adjustment of weight of the synapses, the network looks to minimize the MSE, Eq.

7.

$$MSE = \sum_{i=1}^n \frac{e_i^2}{n} \quad (\text{Eq. 7})$$

By combining the previous equations, this formula can be condensed to a single equation, Eq. 8.

$$MSE = \sum_{i=1}^n \frac{(Target - F(f(I_n W_{n_1}) W_{n_2}))^2}{n} \quad (\text{Eq. 8})$$

Using this equation the primary goal is to drive the MSE as low as possible by changing the weights W_n and M_n . While there are many ways to achieve this, one of the most efficient ways is to look at the rate of change between MSE, and W_{n1} and W_{n2} , shown in Eq. 9.

$$\frac{\partial MSE}{\partial W_{n_1}} \quad (\text{Eq. 9})$$

Or,

$$\frac{\partial MSE}{\partial W_{n_2}}$$

This method is called gradient descent. The ANN interface includes a graphical depiction of this that can be viewed while training a network. Figure 17 shows an example.

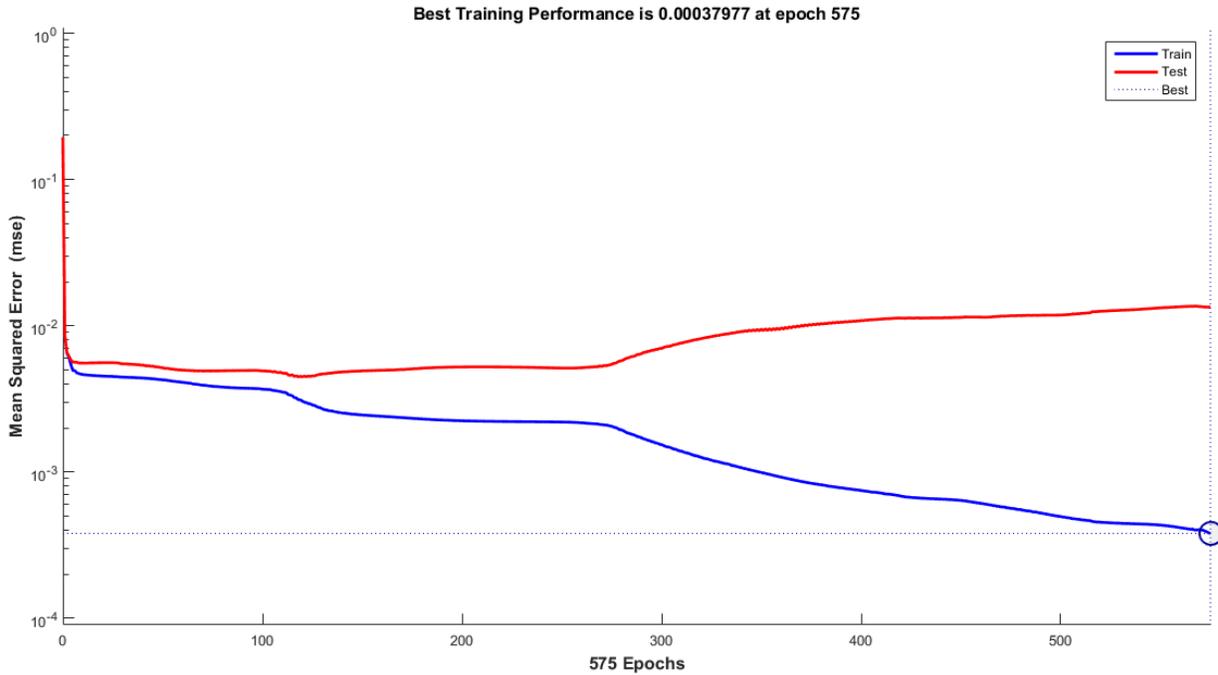


Figure 17: Example Showing Gradient Descent of the MSE

Artificial Neural Networks in Biomedical Applications

ANNs have been used in a wide variety of biomedical applications including, oncology (cancer) [37, 38], medical image analysis [39, 40], human gait analysis [41], electrocardiogram [42, 43], electromyogram, and simulation of elastic tissue, among others.

Limitations with Artificial Neural Networks in Biomedical Applications

Perhaps the greatest shortcoming of an ANN is the difficulty in understanding the process the network uses to make decisions. It effectively becomes a black box that may not always achieve a “correct” answer, and when the correct answer is unknown, this becomes problematic.

Chapter 3: Methodology

This chapter provides the details on how the data was collected, and finally how the data is analyzed. In order to correctly build an ANN that is capable of non-invasively predicting BGLs based upon brain activity many questions need to be answered. What parts of the brain are most sensitive to glucose changes, how complicated does the ANN needs to be, and how well does the ANN track the BGL? Therefore, these research questions were divided into two hypotheses to address two similar, yet different areas of ANN relating to BGL.

Hypothesis 1

BGL will have a stronger correlation with different channels on the Emotiv EPOC and frequency bands.

Hypothesis 2

A healthy volunteer's BGL can be predicted from brain activity by training an artificial neural network.

Experimental Setup

Each volunteer was given his/her own personal glucose meter, lancing device, lancets, and test strips. Volunteers were asked to fast before attending a data recording session. Before each data recording session participants were surveyed about their diet and exercise that day, estimated height and weight, any medications they are on, and their sleeping patterns the night before This was to provide background information to aid in interpreting experimental results, and to help ensure that volunteers were not taking any medications that may have led to complications during the experimental procedure, such as blood thinners. Participants were free to omit any information they were uncomfortable disclosing in the survey. Volunteers in this study measured their blood sugar levels through a standard BGM a total of 4 times, once at the 0-minute mark, once at the 30-minute mark, once at the 45-minute mark, and once at the 60- minute mark. Data

collection was performed at Brody 3S16, a biosafety level 2 lab. Data collection took place roughly between the hours of 10:00am and 12:00pm. During the hour-long data recording process the volunteers wore the Emotiv EPOC headset and had EEG frequency and amplitude data recorded from their scalp. Figure 18 shows a volunteer wearing the headset. In order to elicit a change in the volunteer's BGL they drank 52g of carbohydrates in the form of apple juice. Apple juice was chosen because it was more palatable than the commercially available glucose solution used for tolerance testing, and easier to rapidly drink than soft drinks. This serves to provide ~15 minutes of normal range glycemic data, ~15-30 minutes of high blood glucose levels, and a return to normal range. For the duration of the experiment the volunteers watched a documentary to provide audio and visual stimulation. The auditory and visual stimulation was to help standardize the testing process, in addition to evaluating changes in different brain regions such as temporal and occipital. Despite the data recording length being approximately 1-hour long, the data collection session lasted approximately 1 hour and 30 minutes due to breaks. Each volunteer attended a recording session 3 times.



Figure 18: Volunteer Wearing Emotiv EPOC

The volunteers were asked to fast, or to eat a balanced breakfast before attending the recording session. Initially, the volunteers were allowed to consume whatever they liked before attending data collection, however, some volunteers consumed very sugary, and high caloric intake breakfasts before attending. This made the juice that they drank ineffective at raising their BGL. For example, one participant ate a sausage, egg, and cheese McGriddle®, 48 Carbs [44], and drank a large Coke, ~104 Carbs assuming 52g Carbs per 16 Oz [45], before attending. This had the effect of making their BGL, 110 mg/dL at 0 minutes, 110 mg/dL at 30 minutes, and 72 mg/dL at 60 minutes.

Code and Software Being Utilized

The experiment utilized data acquisition code for the Emotiv EPOC that was previously written by undergraduate research workers, Chris Helms, John Dixon, and Andre Smedstad who were working in the laboratory of Dr. Ricky Castles, Department of Engineering, East Carolina University. This code collects and saves the raw time and amplitude data from the Emotiv EPOC and stores it as a .CSV file.

The .CSV file is converted into an .XLSX file through Excel and imported into MATLAB. This code filters the original 14 channels of time amplitude data with a median filter, a filter that tries to minimize the impact of noise, and a bandpass filter set at 0.5 Hz and 40 Hz. Figure 19 shows an example of the raw and filtered data for the O1 channel. The code also removes DC offset because this offset varies between participants. In addition it performs a windowed Fourier transform on 16 seconds of the time/amplitude data, the code then moves 1 second forward, this results in 1 second of new data and 15 seconds of overlapped data. From this, the delta, theta, alpha, beta, and gamma, activity bands are decomposed into separate arrays. These activity band arrays are averaged over the 16 second/1 second window and turned into a percentage of the

whole activity happening within that channel for a particular activity band. These percentages are collected into a large array, and this array is the input into the ANN.

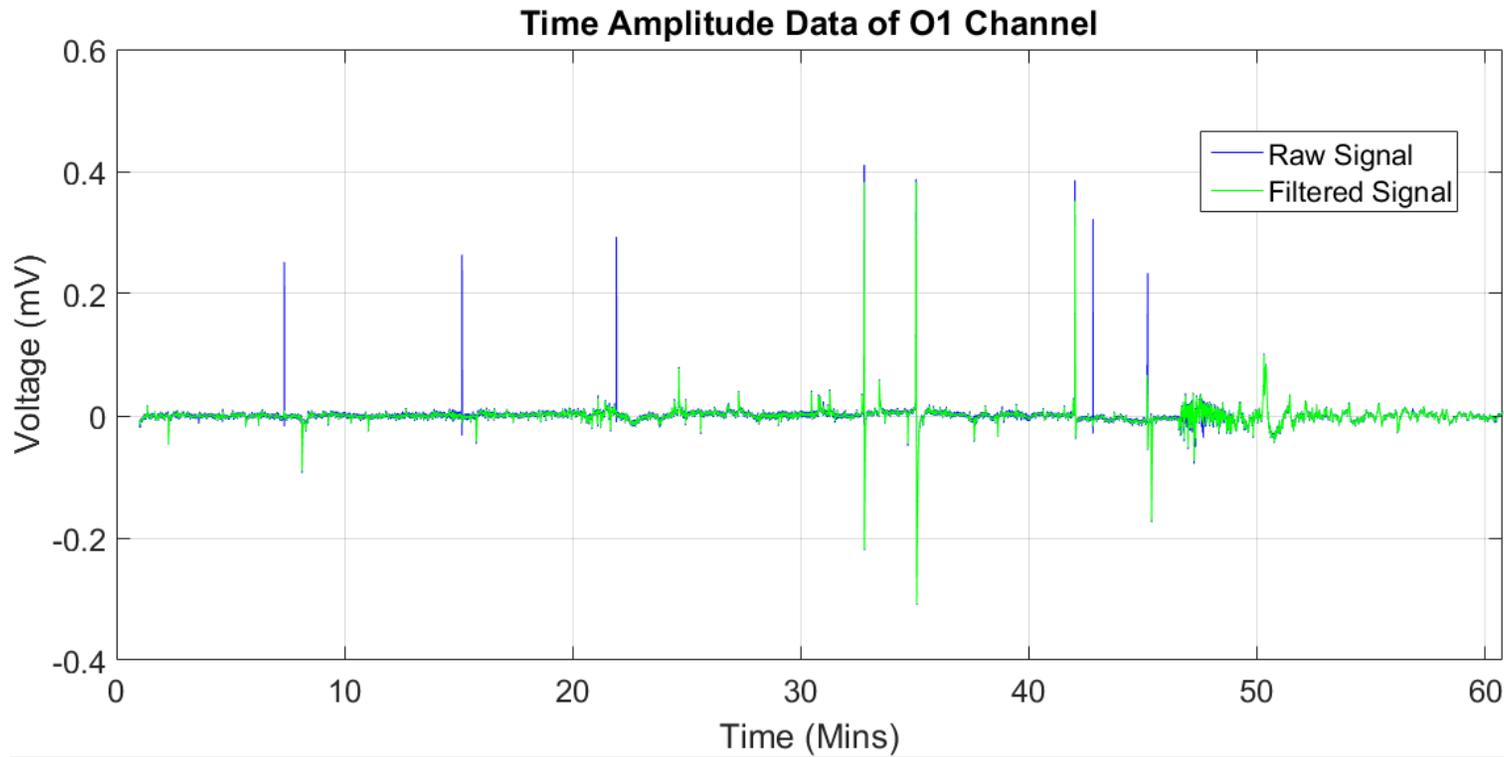


Figure 19: Example of Raw and Filtered Data with DC Offset Removed

This research also utilized MATLAB's artificial neural network toolbox in order to create the network that predicts BGL. This was done by training the network on the first two hours of the input data, brain activity, to the first two hours of the target data, BGLs. The network is then tested on the third hour of data.

Study Participants

The study consisted of 11 volunteers, both male and female, between the ages of 23-35 years of age, who had no known history of diabetes. Originally there was only to be 10 volunteers, however one volunteer did not return past a first visit. Most volunteers participated in 3 recording sessions; however a few volunteers were unable to complete all 3 trials for a variety of reasons, such as leaving town and being non-communicative. In total there are 31 trials worth of data with 8 participants completing 3 trials each, 1 volunteer completed 4 trials due to a recording error, 1 volunteer completing 2 trials, and 1 volunteer completing 1 trial.

Variables Gathered

Dependent Variable (Target)

The target that the ANN is trained to predict is the volunteer's BGL. This is the data that is gathered by the Freestyle Freedom Lite, when the volunteer collects their BGL through a standard home-use glucose monitor. The data was collected at time points, 0 minutes (mins), 30 mins, 45 mins, and 60 mins. It is assumed that for each volunteer their BGL did not deviate between 0 to 15 mins under fasting conditions. A polynomial curve fit was used for the 4 data points represent the volunteer's BGL over the hour-long period.

Independent Variable (Input)

The input data that the ANN is trained with is the power spectrum of the volunteer's EEG data. This is done because the information of interest was most readily available in the power spectrum. It is also the method that Nguyen et al. utilized [2-5].

Other Information Gathered

Additional information that was gathered during the collection is the participant's sex, estimated height and weight, sleeping pattern, and if they are currently on any anti-depressant medication or blood thinner.

Evolution of Protocol

Initial Protocol

The final study had 11 healthy volunteers that were both men and women, 9 volunteers completed all trials. The initial protocol was written for 10 healthy normal male volunteers between the ages of 23 – 35, and was approved by University IRB on September 9th 2015, Appendix A. The protocol stated that the volunteers would participate in 3, 1-hour long, data recording sessions. In addition each volunteer would measure their BGL through finger-stick monitors 3 times per recording session, once at the 0-minute mark, once at the 30-minute mark, and once at the 60-minute mark. The participants would drink juice at 15-minutes to move them away from their basal glucose level.

After collecting preliminary data, it was determined that this protocol did not provide enough information regarding the volunteer's BGL to accurately curve fit the data. Table 4 shows the first volunteer's blood glucose levels during their first visit. Figure 20 shows the BGL plot with respect to time.

Table 4: Blood Glucose Levels of Volunteer 1 during their first data recording session

Volunteer 1 Visit 1		
Finger-Stick	BGL	Time
1	94	10:48
2	106	11:15
3	92	11:36

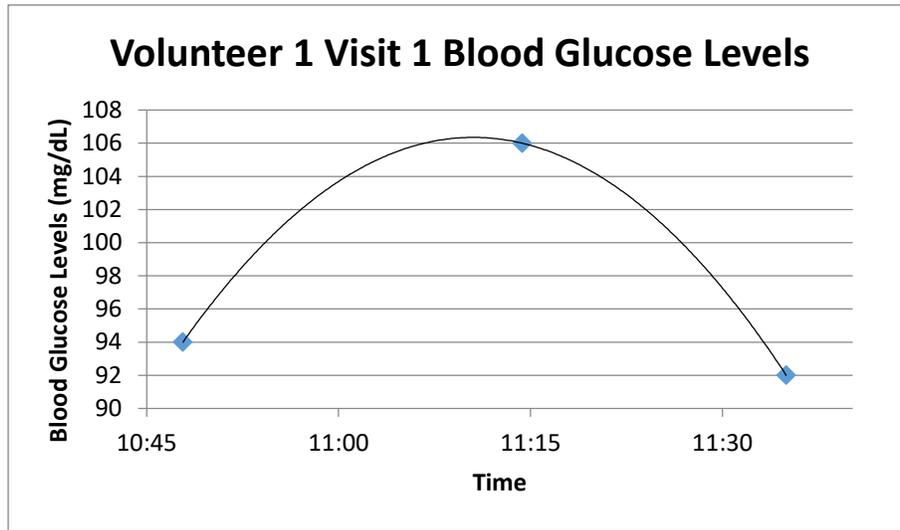


Figure 20: First Volunteer's Blood Glucose Levels during First Visit with Trend-line

Examining the data points that were collected, it was decided that without any input and because the volunteer was stationary, there was no way that the volunteers BGL had deviated from the basal level to a value of ~ 102 mg/dL. Therefore, it was decided that it could be assumed that the glucose level at a time of 0 minutes and 15 minutes was equal. Table 5 shows the BGL with this assumption in place and Figure 21 shows the points and estimated curve with this assumption in place.

Table 5: Volunteer 1 Visit 1 Measured BGL, with BGL at 15 mins = BGL at 0 mins

Participant 1 Visit 1		
Trial	BGL	Time
1	94	10:48
-	94	11:00
2	106	11:15
3	92	11:36



Figure 21: Volunteer 1 Visit 1 with BGL at 15 minutes equal to BGL at 0 minutes

Protocol Amendment

Based upon the previously discussed observations a fourth BGL measurement at the 45 mins mark was added to the protocol. The additional measurement would further improve the curve fit, thus making the eventual trained network more accurate when trying to depict glucose levels. An amendment to the IRB protocol was submitted to allow for a fourth finger-stick. In addition, a volunteer criterion was expanded to allow for female volunteers to participate in the study. This amendment was approved December 15th 2015, Appendix A. Table 6 shows volunteer 3's BGL

and times during their second visit, whereas Figure 22 shows the corresponding graph with trend-line.

Table 6: Volunteer 3 Visit 2 Blood Glucose Chart with 4 Finger Sticks

Volunteer 3 Visit 2		
Trial	BGL	Time
1	82	10:45
-	82	11:00
2	104	11:16
3	106	11:37
4	90	11:55



Figure 22: Volunteer 3 Visit 2 Blood Glucose Graph with Trend-line

Based on the recordings shown in Figure 22 several additional assumptions were made to the final model that was used in training the neural network. The BGL at a time of 5 minutes and 10 minutes were made equal to the BGL measured at time 0. The BGL at time 50 and 55 were made to be interpolated points using the measured values at 45 minutes and 60 minutes. These assumptions were made to prevent the curve fitting procedure from producing gross errors such as predicting that the volunteers BGL dropped to -600 mg/dL, during the time between 0 and 15 minutes, or spiking to 1000mg/dL during the time between 45 minutes and 60 minutes. Figure 23 shows the results of these assumptions.

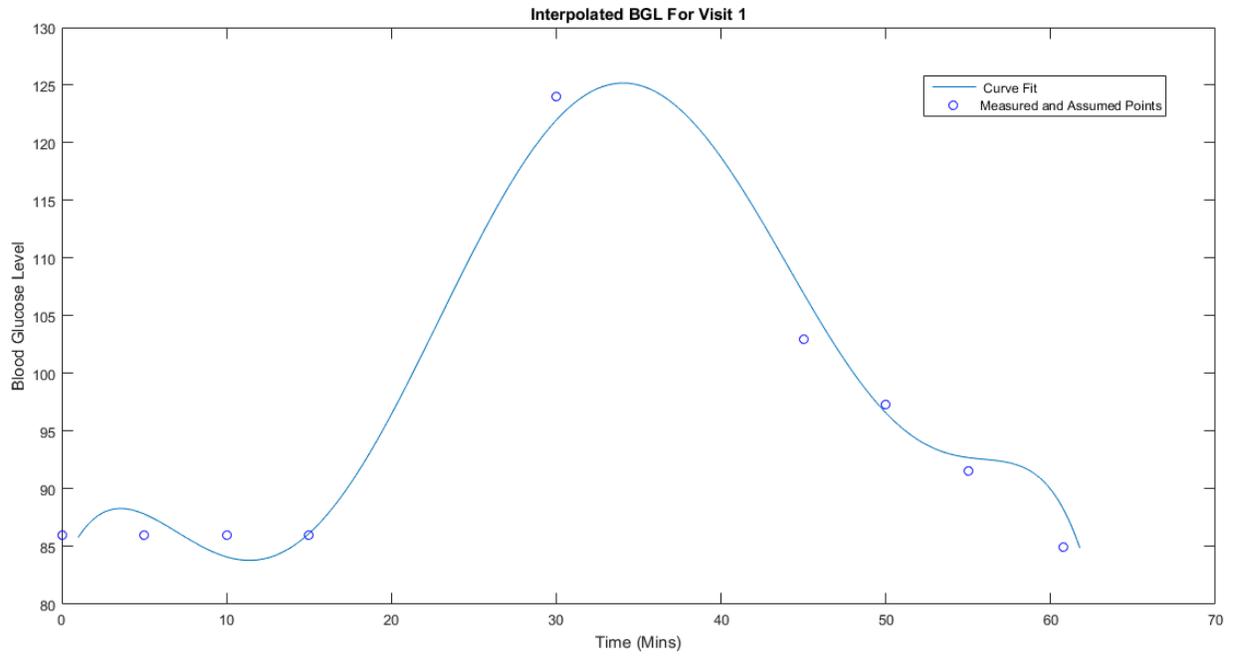


Figure 23: BGL with all assumptions in place

Chapter 4: Results

Evaluating the Channels and Frequency Spectrum

This method pertains exclusively to Hypothesis 1: BGL will have a stronger correlation with different EEG channels and frequency bands. To determine if there was a pattern for which channels/frequencies have a significant correlation to BGL, the programs “BGL Preparation” and “Individual BGL Channel Comparison”, were developed, see Appendix C for full code.

Overview of BGL Preparation

BGL Preparation is the data preparation code; see Appendix C for full code. The recorded EEG and BGL data sets are imported, and a polynomial curve is fit to the BGL data. Then, the three trials of EEG and BGL data collected from each participant are concatenated into a 3-hour long session. An example is shown in Figure 24. The data point for minute 45 is omitted in Trial 1 because this data set was taken before the amendment to the IRB protocol. Both the EEG, and the BGL data are divided by the maximum of each on a per participant basis, this is to ensure that values are unitless. For example, if Participant 1’s maximum BGL is 124, then every BGL value is divided by 124 in order to normalize the data into the range 0 -1. After being concatenated, the data is chunked into 16-second windows. These 16-second windows are then averaged and stored as an array. For example, the BGL data for 0 seconds to 16 seconds is averaged. Then, the window moves 1 second along the BGL data, to times 1 second to 17 seconds, this data is then averaged and stored. This repeats until the end of the BGL data. The EEG data is then filtered. The first filter that is applied is a median filter; this is done to assist the removal of occasional artifacts that are present within the signal.

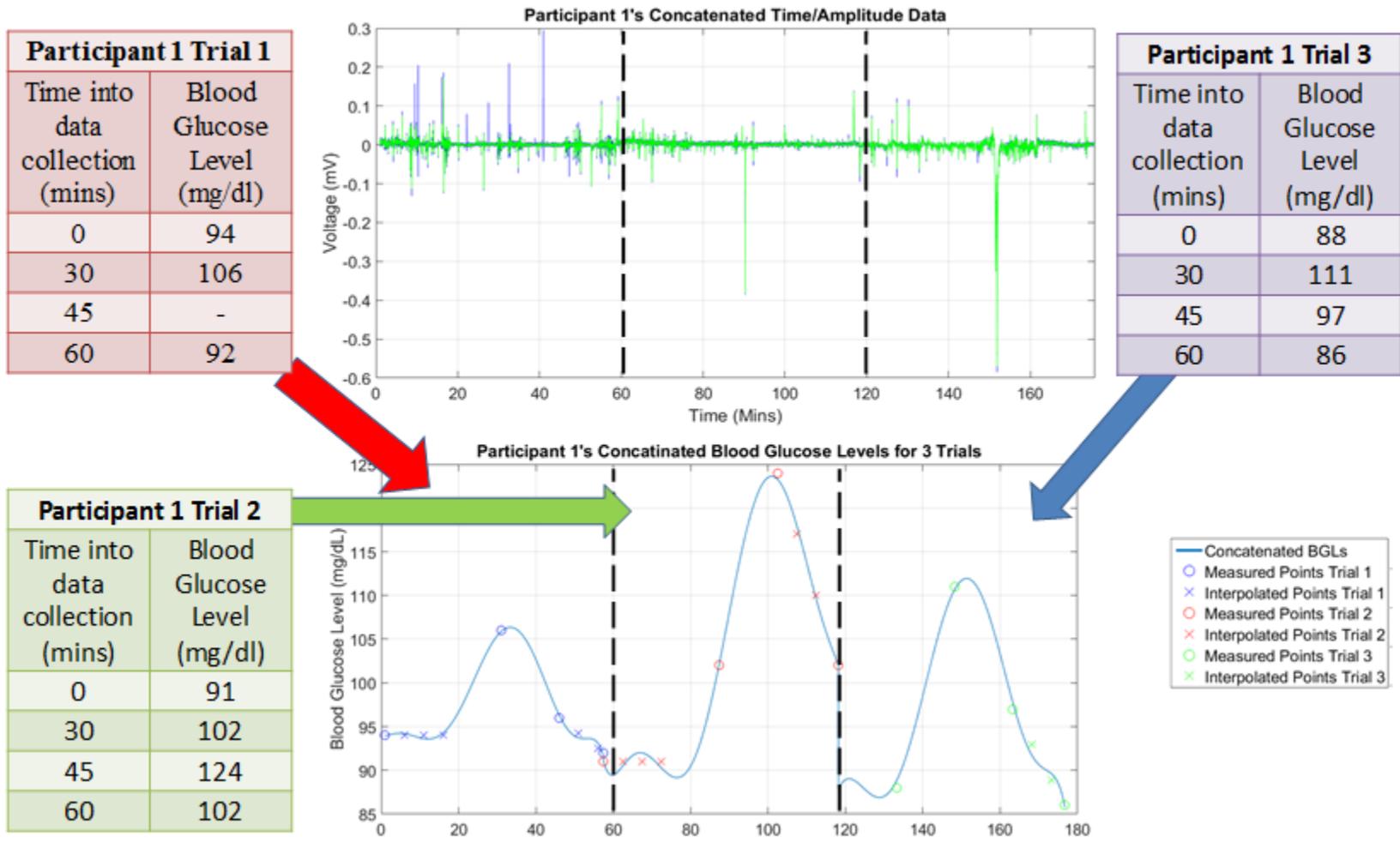


Figure 24: Example of three, 1-hour, data collections concatenated into single recording

The next filter that is applied is a band pass filter; the filter is set for between 0.5 Hz and 40 Hz. This is because brain activity lies along this spectrum and anything outside this frequency spectrum can be discarded as noise. The DC offset of the signal is also removed; this is done to look at the amplitude fluctuations directly. Similar to the BGL data, the EEG data is chunked into 16-second windows. The Fourier transform (FT) is then performed on this data, in addition the average amplitude of the EEG signal is taken in the same way that the average BGL is taken. This data is then decomposed into the frequency bands for Delta, Theta, Alpha, Beta, and Gamma. This frequency spectrum is then normalized by being changed into a percent of all frequency activity for that specific channel,

$$\% \text{ Delta AF3} = \frac{\text{Delta}_{AF3}}{\text{Delta}_{AF3} + \text{Theta}_{AF3} + \text{Alpha}_{AF3} + \text{Beta}_{AF3} + \text{Gamma}_{AF3}}$$

The final step in this code is to place all the data into an input matrix, the EEG data, and an output matrix, the BGL data. These two matrices are used by further code, including MATLAB's ANN GUI, the code for determining which features are significant, and determining the optimal number of neurons. This data is then saved to an excel spreadsheet.

Determining Features that have a Significant Correlation with BGL

A total of 84 different features were examined for their significance with BGL. These features included the delta, theta, alpha, beta, and gamma frequency bands for each of the 14 Emotiv Channels. In addition, the amplitude change for each channel was also examined. To determine significance of a feature a two sample t-test assuming unequal variance with a hypothesized mean difference of 0 was used to compare a participant's first 10 minutes of data. This window of time is assumed to be baseline data with little change since the participant has not eaten nor drank anything and is stationary, to their data between minutes 30 and 40, time frame where the

juice that has been drank has had time to be metabolized. Despite being a within subject comparison, unequal variances are assumed because of the volatility of EEG data. A p-value of 0.1 is considered to be significant due to the small number of samples. The t-test is repeated for each of the three trials the participant completed. An example of the features extracted from channel AF3 is shown in Figure 25.

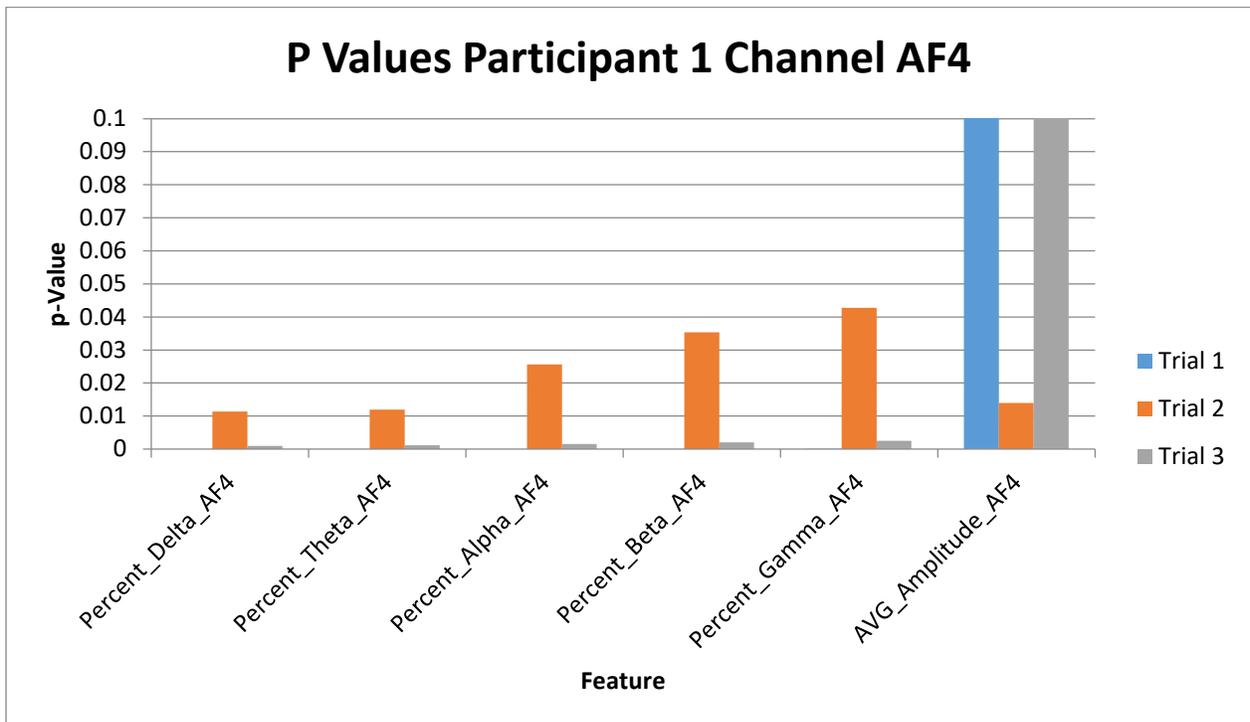


Figure 25: Example P-Values Participant 1 Channel AF4

If two out of the three trials indicated that the feature had a p-value less than 0.1 it was considered to be a good feature. From the above graph all of the features with the exception of the amplitude are considered to be good features, some of the p-values are so small that they are not noticeable. The amplitude p-value is considerably larger than 0.1 thus, it exceeds the graph area; if it was included the p-values for the other features would not be observable on the same graph. The p-values being so small for some features can be attributed to a few potential factors; the first of which is that during that specific trial there was good correlation between

BGL and that specific channel; a second reason is that the signal was noisy during this time, potentially due to poor conductivity, a third possibility is that the participant was more stimulated during that point of time in the video being watched. Table 7 shows what channels were considered to be good, highlighted in green, these are the channels where 2/3 of the trial's p-value is below 0.1. The channel is highlighted in red otherwise.

Table 7: Values for where 2/3 of the Features were "Good" P6 and P8 did not complete the three trials therefore the data is blacked out, and P7's data is white as it was too noisy to include.

	Channels													
	AF3	AF4	F3	F4	F7	F8	FC5	FC6	O1	O2	P7	P8	T7	T8
P1	Red	Green	Green	Red	Red	Green	Green	Red	Green	Green	Green	Red	Green	Green
P2	Green	Green	Green	Green	Green	Red	Green	Red	Green	Green	Green	Green	Green	Red
P3	Red	Red	Red	Green	Green	Green	Red	Red	Green	Red	Red	Green	Red	Red
P4	Green	Green	Green	Red	Green	Green	Red	Red	Green	Green	Red	Green	Red	Red
P5	Green	Red	Red	Green	Red	Green	Green	Red	Red	Green	Red	Green	Green	Green
P6	Black													
P7	White													
P8	Black													
P9	Green	Green	Green	Green	Green	Green	Green	Green	Green	Green	Red	Green	Red	Green
P10	Red	Green	Red	Green	Green	Green	Green	Green	Green	Red	Red	Green	Green	Green
P11	Green	Green	Red	Green	Red	Green	Green	Green	Red	Green	Red	Green	Green	Red
% Good	62.5	75.0	50.0	75.0	62.5	87.5	75.0	37.5	75.0	75.0	25.0	87.5	62.5	50.0

The final features that are used for training the neural network are channels where 75% or greater of the channels were considered “good”. These channels are AF4, F4, F8, FC5, O1, O2, and P8.

Figure 26 shows the location of the channels used for training. The amplitude never fit the inclusion criteria and is excluded.

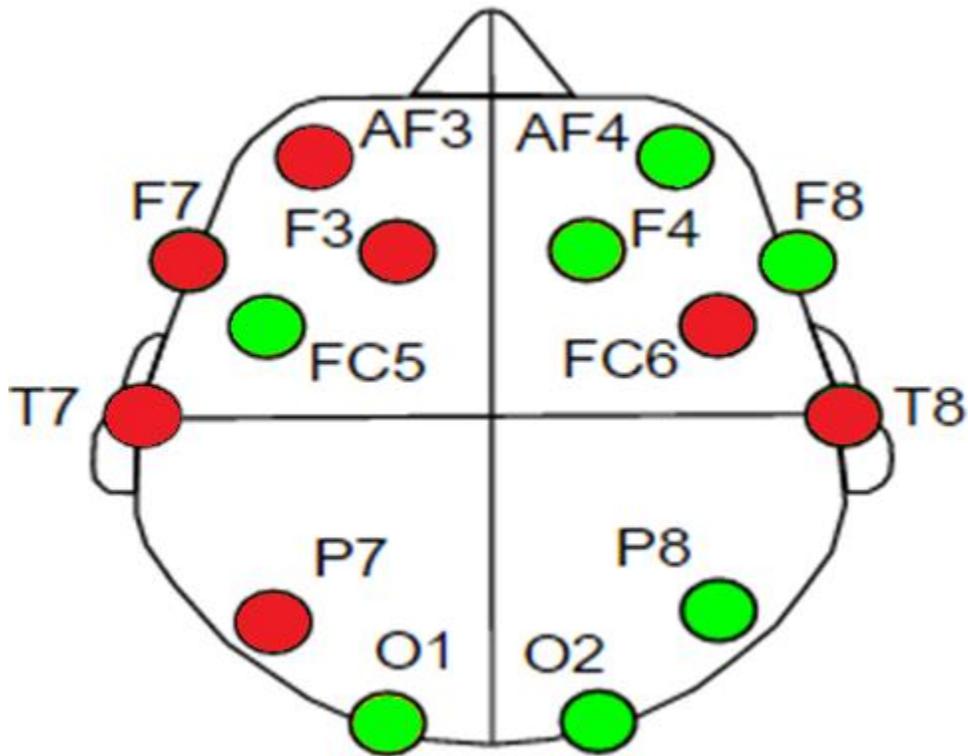


Figure 26: Channels Used to Train ANN

In addition to determining which channel locations were the most influential by using a T-test the data was also normalized on a per-participant basis. This method is detailed in sections “Overview of “BGL Code for R Coefficient Neural Networks Training” up to the section “Method for determining number of hidden neurons”. However it is not used in determining which channels are used to train the ANN.

Overview of “BGL Code for R Coefficient Neural Networks Training”

This code trains an ANN for each of the 14 channels measured by the Emotiv EPOC for all of the frequency bands 50 times, i.e. “Delta F3 ANN 1”. These networks are all saved to an Excel file for further data processing and observation. In general the code performs the following: Previously processed data from “BGL Code for R Coefficient Preparation” is loaded. The user is then prompted to save data to as an Excel file. The code then trains 50 ANN for all 14 channels, and all frequency bands. For example, there would be 50 networks trained exclusively on Delta AF3. The code then saves the R coefficient for networks to an Excel Worksheet.

Data Normalization per Participant

After the correlation coefficients R was determined for each ANN, the average R value across the 50 training trials was determined for all 5 frequency bands on each of the each of the 14 channels for each participant, i.e. Participant 1 has the 50 R values returned by the ANN averaged for “Delta AF3”, then “Theta AF3”, etc. These averaged values, are normalized by determining the mean R value for each participant by summing each R value and then dividing by 70(5 frequency bands on 14 channels). By dividing by all possible frequency bands input combinations normalized values for every frequency band on every channel for all participants to account for individual differences. The normalized values were categorized as follows: a value between 0.80 and 1.20 is considered to be average, a value less than 0.80 is considered below average meaning that frequency range on that channel is not well correlated with BGL, and a value greater than 1.20 is considered above average meaning that frequency range on that channel shows an above average correlation with BGL. An example calculation is shown below.

Table 8: Table Showing R Value for 50 ANN during Example Calculation

Trial	Delta_AF3
1.00	0.09
2.00	0.38
3.00	0.16
4.00	0.08
50.00	0.12
AVERAGE R	0.27

$$\text{Average } R = \frac{\text{Trial}_1 + \text{Trial}_2 + \text{Trial}_3 \dots + \text{Trial}_{50}}{50}$$

This average is divided by 50 because there were 50 ANN trials run.

$$\frac{0.09 + 0.38 + 0.16 \dots + 0.12}{50} = 0.27$$

Table 9: Data Showcasing Example Averaged R Values for Different Participants and Different Channel Locations.

Participant #	Delta AF3	Delta AF4	Delta F3	Delta F4
1	0.27	0.17	0.27	0.34
2	0.17	0.16	0.2	0.15
n	-	-	-	-

The data is then normalized for each Participant. For the calculation below Delta AF3, all the way through Gamma AF4 is averaged. This results in 70 R values being summed.

$$\text{Normalized Value for Participant} = \frac{\text{Delta AF3} + \text{Delta AF4} + \text{Delta F3} \dots \text{Gamma AF4}}{70}$$

$$\text{Participant 1 Normalized Value} = \frac{0.27 + 0.17 + 0.27 \dots 0.12}{70} = 0.21$$

Table 10 shows a few examples of few averaged values for Participant 1. Certain channels such as channel F4 have a stronger than average correlation, with a normalized value of 1.60 in comparison to the rest of Participant 1's channels.

Table 10: Averaged Data Normalized For Participant 1 Example Calculation

Delta AF3	Delta AF4	Delta F3	Delta F4	Delta F7
1.27	0.81	1.30	1.60	1.39

This information was calculated by dividing the averaged R value for the participant, by the normalized value for that participant. An example calculation is shown below.

$$\text{Averaged Normalized Data} = \frac{\text{Averaged Delta AF3}}{\text{Normalized Value for Participant}}$$

$$\text{Averaged Normalized Data for Participant 1 Delta AF3} = \frac{0.27}{0.21} = 1.27$$

The final results are displayed in Figure 27.

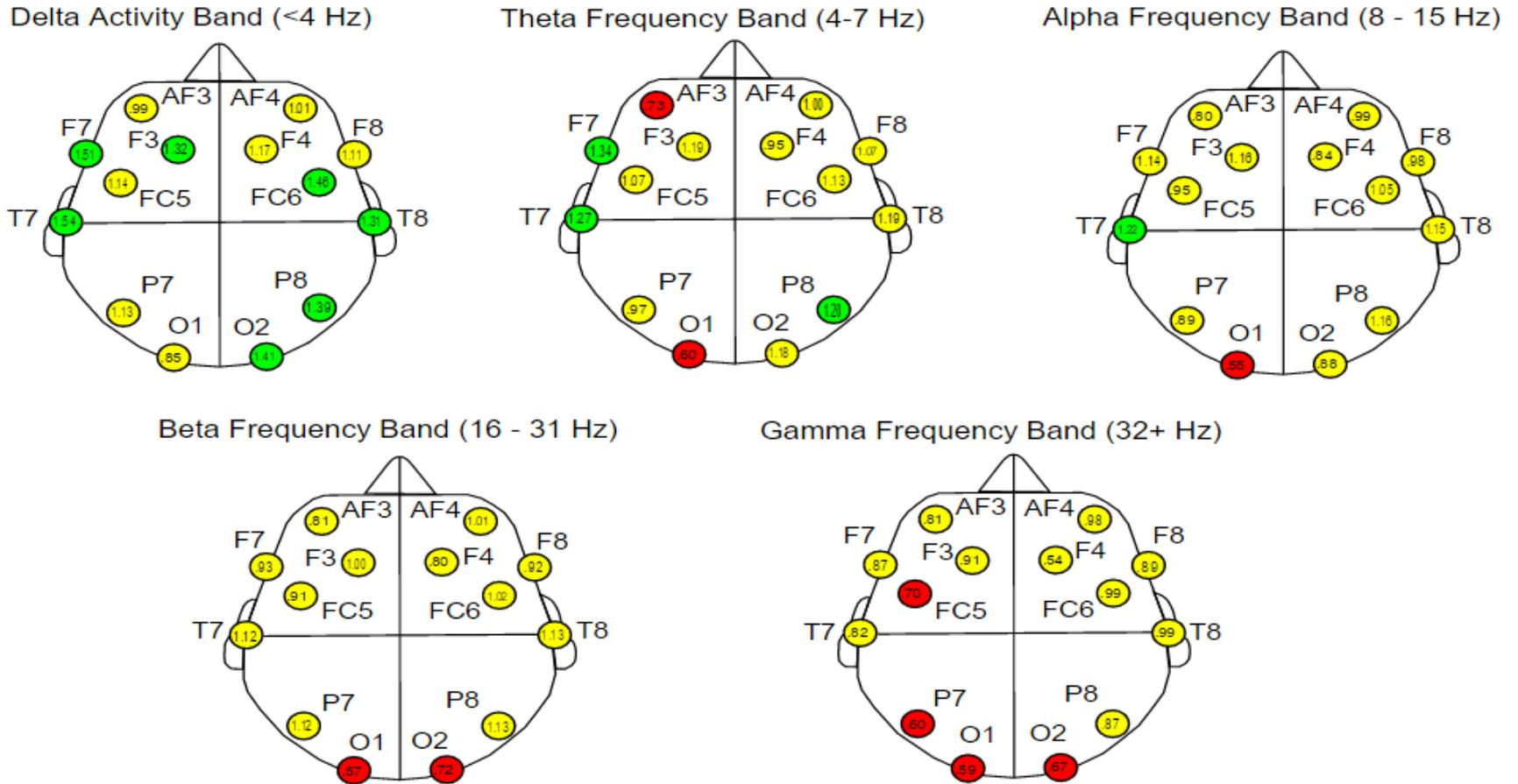


Figure 27: Frequency and Channel BGL Correlation: Each Channel/Activity band had an ANN trained 50 times for each participant's combined data. These values are then averaged and normalized. The mean of the normalized data is presented for each of the different frequency bands. Green is a channel showing a normalized value 1.20 or greater, this is a strongly correlated channel. Yellow is a channel with a normalized value between .80 and 1.19, this is an average correlation. Red is a channel with a normalized value .79 or less, a below average correlation.

Method for Determining Number of Neurons

After determining the significant features that occur most frequently the number of hidden neurons was determined by iteratively training a network of 1 to 100 neurons. Each size network was trained 3 separate times and the resulting R values and mean square error (MSE) was averaged. The number of hidden neurons selected represents the point where the network most consistently returned a high R value and low MSE. The code used for determining the number of neurons is “BGL Code for Number of Hidden Neuron Determination”.

Overview of Code BGL Code for Number of Hidden Neuron Determination

This code imports data from the “BGL Preparation” code and places the significant features into a matrix as the training set and the BGL values into a target matrix. The code then proceeds to train a network with hidden neurons ranging from 1 to 100 repeating each number of neurons 3 times and recording the R value, the MSE of the training, and the MSE of the testing to an Excel spreadsheet. A total of 3 participant’s data was run out of the 9 total participants. 3 participants were chosen due to the length of time it took to run the program, approximately 7 days. In order to make the final selection for the number of neurons the total R value of the network is examined, alongside the MSE of the training data set, and the MSE of the testing data set. The data from each of the participants is averaged in order to look for the most consistently strong number of hidden neurons. In an ideal neural network the number of neurons chosen would reflect the point where the R value is at its highest, and the MSE of the training, validation, and testing, is at its minimum. The networks trained in this experiment are Bayesian Regularization networks, which seek to minimize the network weights, and can return better networks for small and noisy data sets. However, these networks do not use a validation set to stop training in order to prevent overfitting, an example of a neural network method that does this is the Levenberg-

Marquardt algorithm. For this experiment the optimal setup did not occur, therefore the number of neurons that resulted in the smallest MSE for the testing dataset was selected; this number is 99 neurons.

Training Neural Network on Two Hours of Participant's Data

This method pertains exclusively to Hypothesis 2; a healthy volunteer's BGL can be predicted from brain activity by training an artificial neural network.

The final step in developing an ANN to predict BGL based upon brain activity is to train a network using a participant's first two visits as training data. This is accomplished through the use of the program "BGL Code for Two Hour Preparation" the full code is available in Appendix C. This code is extremely similar to "BGL Code Preparation" except that it prepares two hours of data instead of one.

Predicting Third Hour of Participant's Data

After training a network for each participant that completed all three training sessions, the network is tested with the features from the third training hour. These features have never been seen before by the network. Figure 28 and Figure 29 shows an example of a network predicting the data it was trained upon, and predicting the third hour of data, or data it has never seen before. Figure 30 and Figure 31 show an example of a network that did not yield good results. The results of each participant's network can be seen in Appendix D

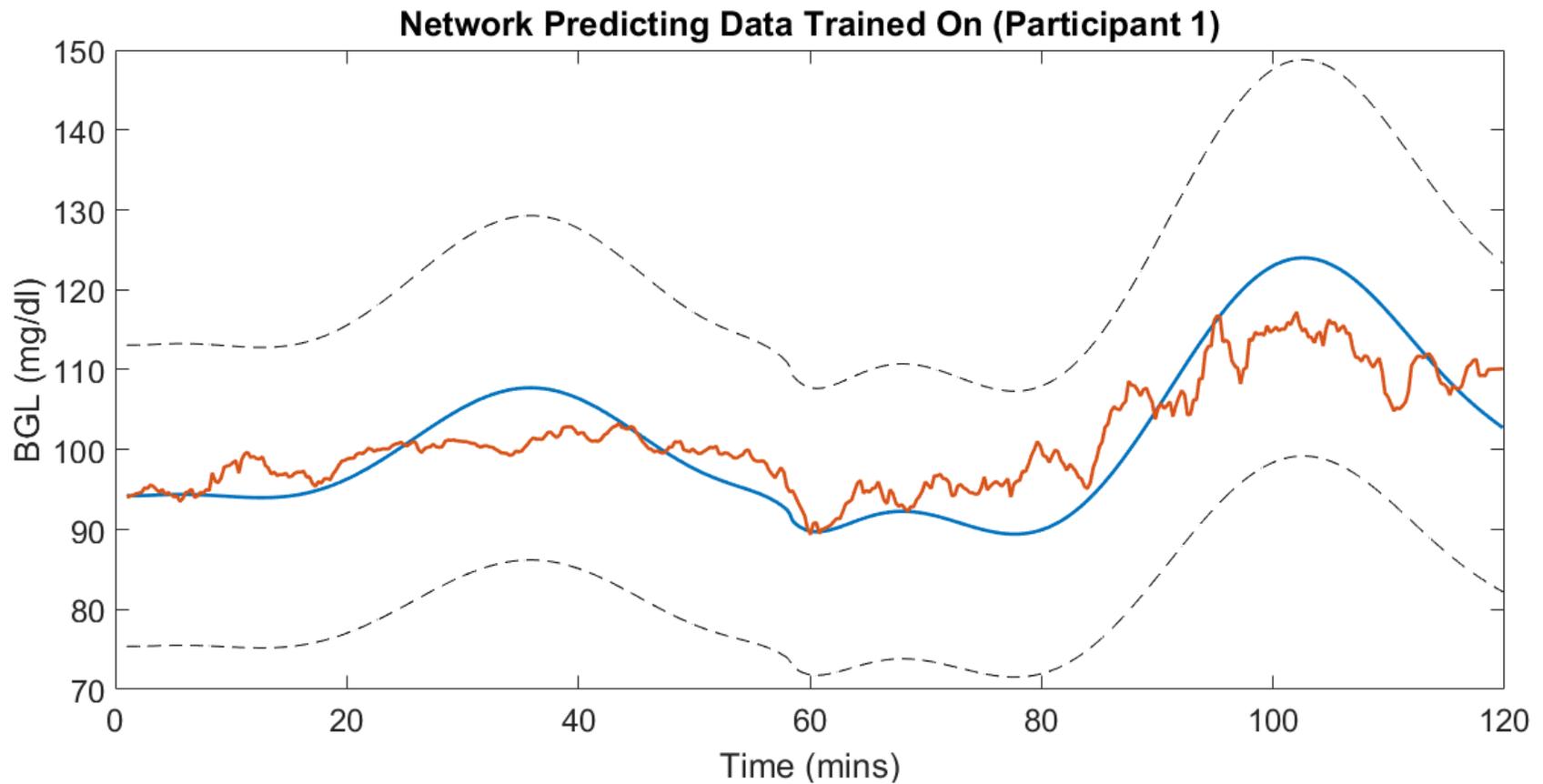
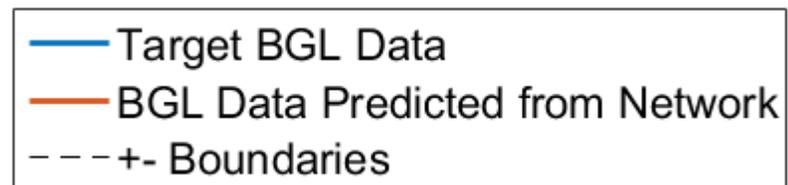


Figure 28: Network with good following predicting data it had been trained on. The Blue line is the theoretical BGL of the participant that has been curve fit; the red line is the predicted BGL from the ANN. The dashed lines are $\pm 20\%$ error bands from the Target BGL value.



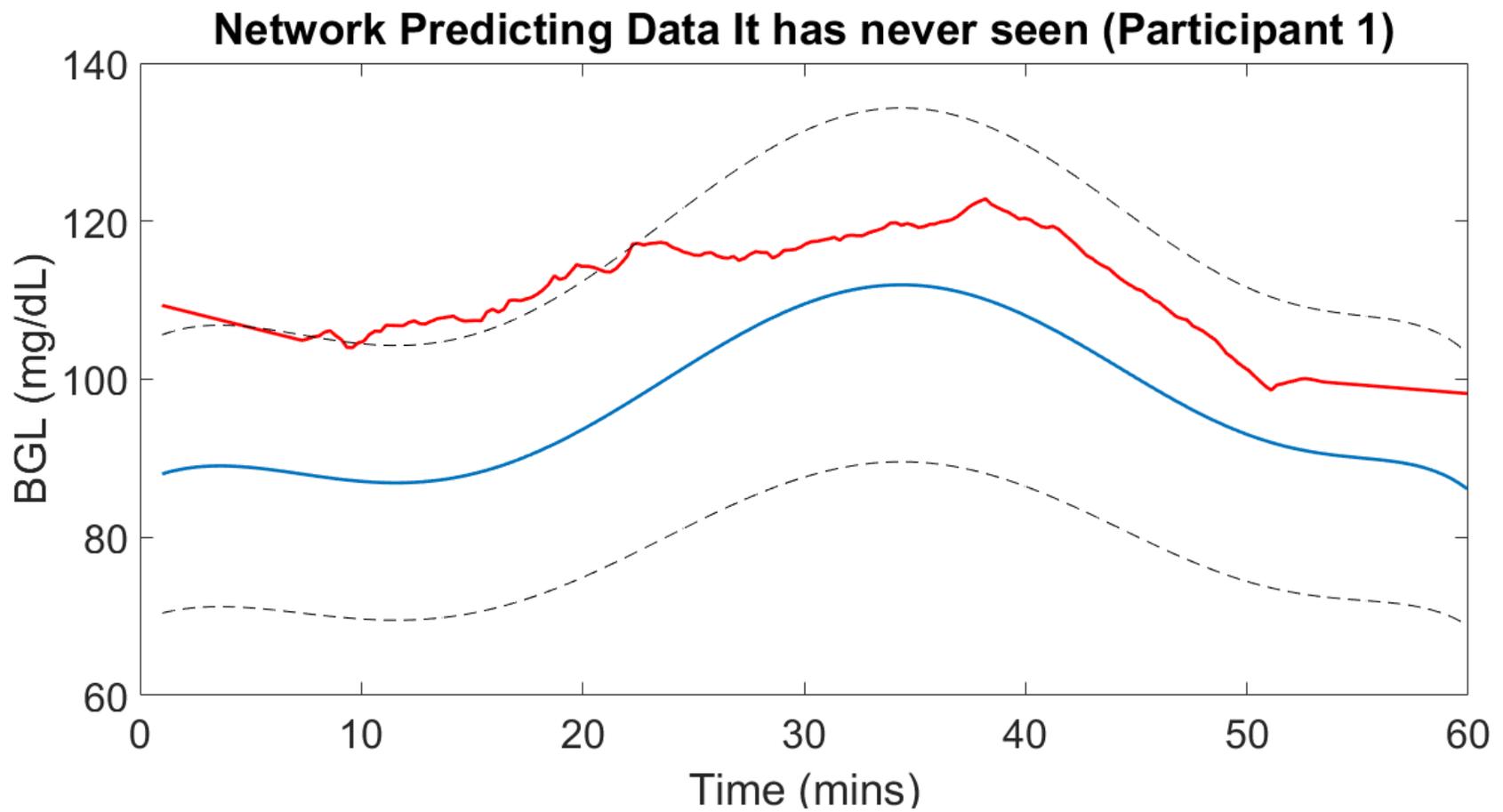


Figure 29: Network with good following predicting data it has never seen before

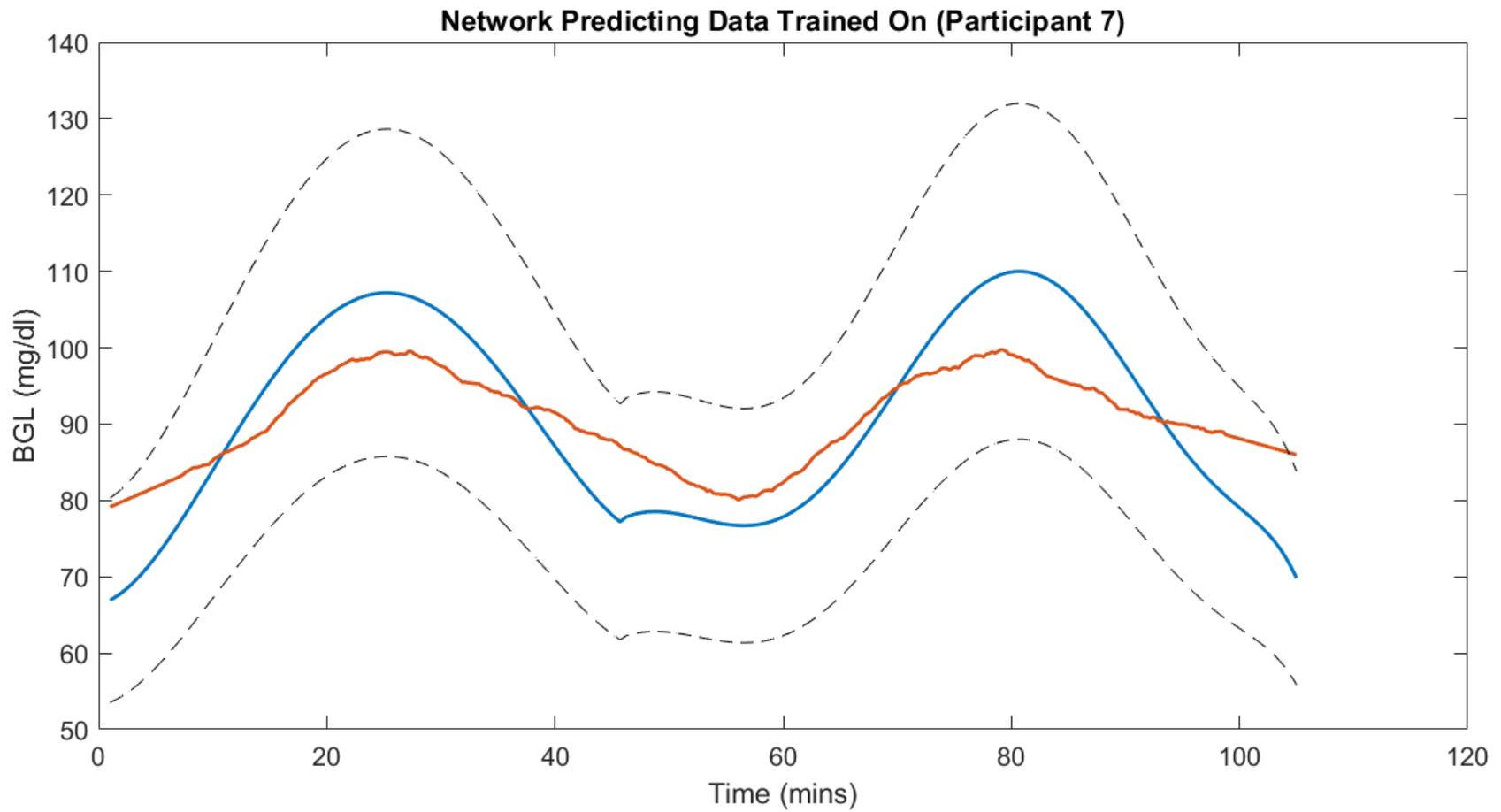


Figure 30 Network with poor following predicting data it had been trained on. The Blue line is the theoretical BGL of the participant that has been curve fit; the red line is the predicted BGL from the ANN. The dashed lines are +/- 20% error bands from the Target BGL value:

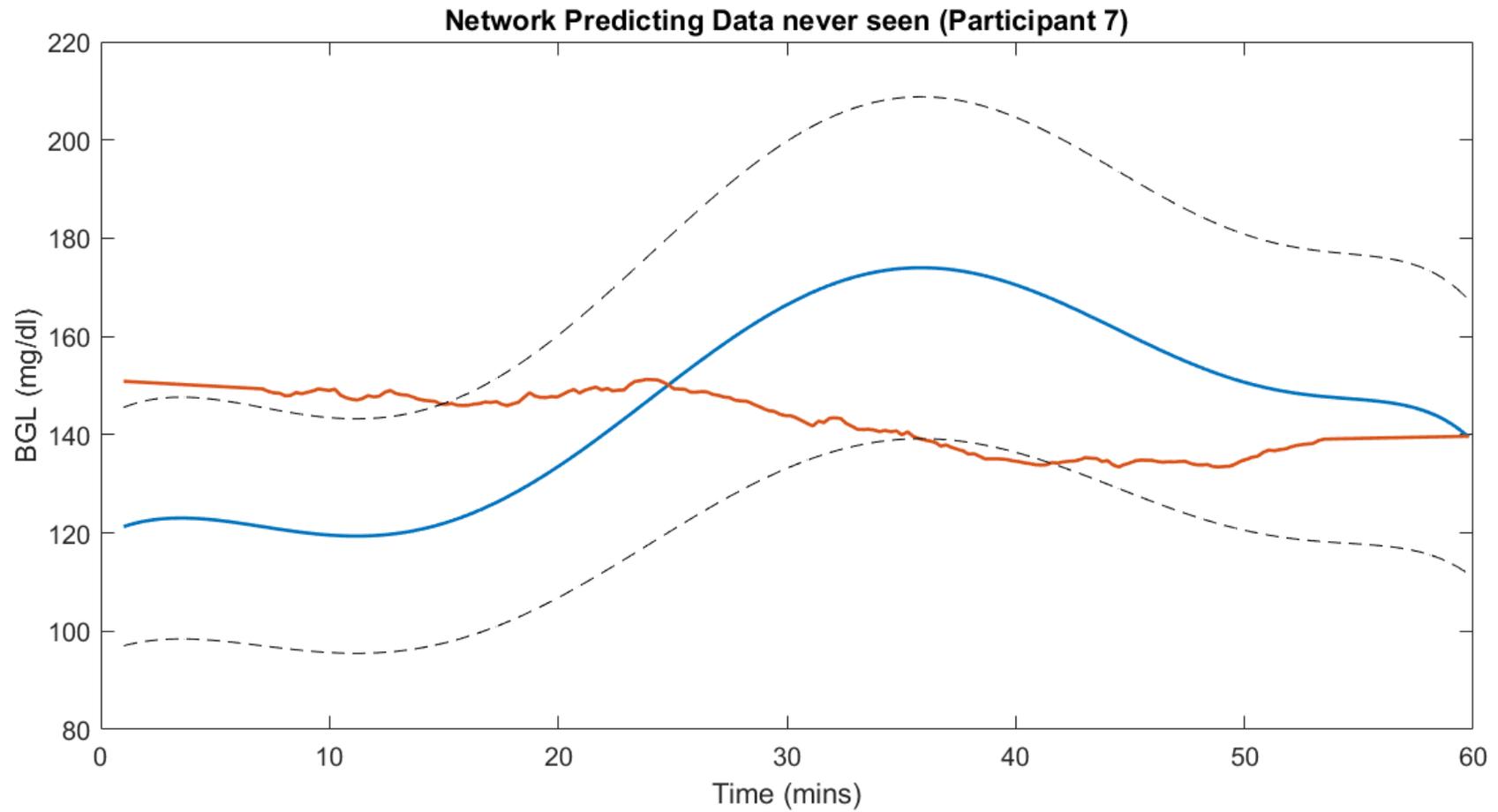


Figure 31: Network with poor following predicting data it has never seen before

Chapter 5: Discussion

Using brain activity to predict someone's BGL is a challenging and complex problem. This is due in part to the complexity of the signals coming off of the human brain. This difficulty is compounded by challenges with the hardware itself. Despite these challenges and limitations utilizing brain activity to non-invasively monitor BGLs appears to have merit, indicated by some positive results, such as Participant 1's data. However, due to the low sample size no statistically significant conclusion can be made to either accept or reject the hypotheses and merits a considerable amount of further study and refinement.

Challenges with Brain Activity Signal

Brain activity measured via EEG is a complex signal. It is influenced by all of someone's conscious thoughts, memories, controls autonomic systems in the body, and is what makes an individual. In an experimental setting it is impossible to control exactly what someone is thinking or feeling during the course of the experiment. For example, while someone could be watching a movie, they could be tired, disinterested, contemplating what they need to accomplish that day, considering what they would like for lunch/dinner. Any number of things could be going through their head, and what one person feels is certainly not exactly how another person feels. This experiment was set up so that the participant would have some form of stimulus throughout the data collection process. Initially the participants watched a beach scene, something very neutral and relaxing, for the duration of data collection. Their preliminary responses were that it was extremely boring, and often found they were daydreaming, or fighting off the urge to doze off. Out of courtesy for the participants the visual stimulus was changed to a video documentary, available on Netflix. However, this also had a variety of responses, for example, Participant 4 was "grossed out" by some scenes of the documentary, and there were a

few scenes that Participant 7 found to be very humorous. Furthermore, not everyone elected to watch the same video, one participant had already seen all of the documentaries and requested to instead watch something else, another participant requested to read in order to catch up on their schoolwork, and yet another requested to play their Gameboy during the data collection process. Other challenges regarding brain activity existed as well, such as how much sleep a person gets per night, if they are on medication, stress level, etc. A small survey was given to each participant at the beginning of each data recording session, the participant was free to ignore any question they were uncomfortable answering, and the results of the surveys were only for assistance in understanding any abnormalities that may have arisen in the data. The surveys are available in Appendix B. Despite all of the issues and complexities of monitoring brain activity, in practice this method would have to be robust enough to compensate for all of these difficulties and more.

Challenges with Emotiv Hardware

In addition to difficulties with deciphering a participant's brain activity, the quality of the data measured per participant was sometimes problematic due to the hardware being utilized. The biggest issue with this device is that the leads holding each of the channels are prone to being stretched out, and it can be challenging to achieve good conductivity between various participants. For example, the female participants in this study tended to have smaller skull sizes than the males who participated in the study. Therefore, it was often challenging to get a good signal from all of the channels simultaneously. Figure 32 shows an example where the raw signal being monitored from the Emotiv EPOC became extremely noisy. If this noise occurred early or late in the recording session it was often removed from the data, typically in 15 minute increments.

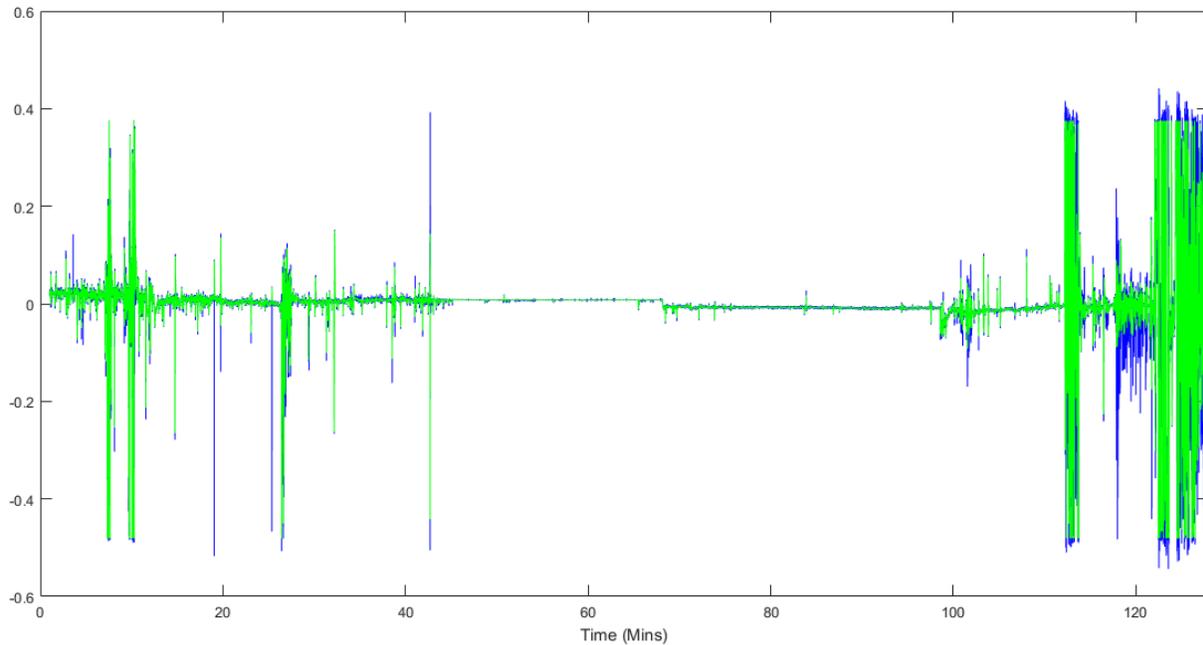


Figure 32: Noisy Raw Data from Emotiv EPOC due to poor conduction

Another issue with the Emotiv EPOC is that the channels have difficulty piercing through thick hair; one of the participants had braided hair and therefore the EPOC had extreme difficulty in obtaining good conduction on that participant.

Challenges with Training ANN

In addition to the challenges with interpreting the EEG data and hardware difficulties there were many challenges with training the neural network. Many of these challenges dealt with scope creep and simply not having the amount of time and processing power required. For example, when determining the optimal number of neurons to place in the neural networks hidden layer it took a week to run the program. This program only examined a single hidden layer with the number of hidden neurons ranging from 1 to 100. A better solution to the network could easily exist with an increased number of hidden neurons, or by potentially adding a second, third, or fourth hidden layer. However, while it is possible, it is also challenging without a significant

increase in processing power. It took approximately one week to train 300 networks ranging from 1 to 100 hidden neurons. It could have taken months to attempt to discover the optimal number of neurons in a 2 layer neural network that was 1 to 100 neurons in the first layer and 1 to 100 neurons in the second layer, or 90,000 separate neural networks. It was found that the increased number of neurons resulted in the smallest MSE for the Testing data of the network. A further complication is that the weights for the hidden neurons are randomly seeded. Therefore, successive training can yield drastically different results. In this experiment the number of neurons that most consistently returned the lowest MSE for the Testing data was chosen, however there were only three repeated trials and the results may have been different with more trials run.

Comments on Neural Network Results

Despite the multitude of difficulties and complications with the data, hardware, and experiment, it seems that utilizing brain activity to predict an individual's BGL has promise. Examining the figures in Appendix D the predicted BGL reported by the neural network tends to track the BGL that was measured and approximated by a blood glucose monitor. This can be seen especially well in Participant 1, and also in Participants 3, 5, and 7. While it certainly does not perfectly track the target BGL at all times, looking at the end of Participant 3's data, or looking at Participant 5 being well below the curve. These problems could potentially have been caused by any of the multitude of problems previously mentioned, in addition when examining the raw data the signal is polluted by a considerable amount of noise. This being said, there were several participants that had very poor results, such as Participant 2 and 4. Participant 2 had a significant amount of noise in the data, see Figure 32, however, unlike Participant 1, 3, 5, and 7 there was almost no tracking of the BGL, the target often goes down when it should go up and vice-versa.

This could potentially be due to the network overfitting the data. While 99 neurons was selected as the most generalizable, optimal number of neurons, it certainly may not be the optimal number on a case by case basis. Perhaps the most challenging aspect of these network results is due to the limited amount of data, only 3 hours, and that all blood glucose measurements are considered within the normal range. To improve a neural network's ability to generalize new data it is almost always best to increase the amount of data that is has been trained on. It is also difficult for a neural network to accurately predict a situation that is outside of the scope of training data. For example, a network cannot predict a BGL number that it has never seen before, in other words, since all of this data was normal glucose level data, the network would never accurately return a result for a glucose level that would be classified as hyperglycemia.

Chapter 6: Conclusion

The results of the experiments show that using brain activity to non-invasively predict BGLs is a promising idea that may offer an alternative to standard home monitoring techniques, and supports both hypothesis 1 and 2.

Hypothesis 1:

Observing the results in Table 7 it was found that there did seem to be some channels that corresponded more strongly to BGL changes than other channels. These channels are AF4, F4, F8, FC5, O1, O2, and P8. The channels that corresponded most strongly are similar to the one utilized in the studies conducted by Nguyen et al., Figure 6. The Emotiv EPOC hardware used in this research does not have a C3, C4, channel, but Channels F4 and FC5 are relatively close.

Interestingly the channels appear to be right side dominant, with 5 out of 7 channels being on the right side whereas the left side counterpart returned insignificant. The reasons for this right side domination is unknown and merits further study and data collection. Despite this supporting evidence no hard statistical correlation can be drawn due to the low number of both participants enrolled in the study, and number of repeated measures.

Hypothesis 2:

After training the neural networks and observing the resulting data it can be concluded that the trained neural network is capable, in some instances, of generalizing new cases and tracking previously unseen BGLs based upon features extracted from brain activity data. Of all the participants who completed the trial 3 out of 9 seemed to have networks that would track the changing BGL, Participant 1, Participant 3, and Participant 9. An inversion of what was expected happened in 2 of the 9 cases, Participant 2, and Participant 9. The reasons for this inversion are

yet unknown. The remaining participants exhibited a mostly static BGL reported by the trained ANN. More information is needed in order to make a statistically strong conclusion one way or another.

Future Work

Some potential improvements moving forward in attempting to non-invasively predict BGL through the use of brain activity are.

- 1) To increase the sample size of the experiment. By increasing the sample size of the experiment it would serve to strengthen the case that utilizing brain activity to predict BGL is a plausible solution, or poor solution.
- 2) Increase the number of samples taken from each participant. Increasing the number trials each participant undergoes and increase the number of BGL samples taken from each participant will allow for more training data, and also allow for more data to test the artificial neural network. Perhaps there is an optimal length of time that returns good generalizability without the network overfitting.
- 3) Recruit participants who have diabetes. By recruiting people who have diabetes it will be possible to collect BGL values that are outside of the normal range, such as hyperglycemia and hypoglycemia.
- 4) Collect and test network data during a variety of activity. This non-invasive method of monitoring blood glucose levels is nearly worthless if the individual is unable to move without causing the device connection problems, or reporting inaccurate data.

- 5) Evaluate the relationship between removing channels and increasing length of time required to train network. While the results of this experiment used the channels that were shown in Figure 26, it could be possible to further remove channels required to train a network.
- 6) Examine other potential features hidden in brain activity data that may better correlate to BGL. This experiment examined the features and amplitude data for each of the channels. It is possible that there is another feature that is better suited to predicting BGL.
- 7) Purchase adjustable EEG recording device and dry electrodes. This will help ensure that good conduction is present regardless of a participant's skull size or hair thickness.
- 8) Examine alternative neural network sizes, and number of hidden layers. This will help ensure that the optimal network solution is most consistently found.

References

- [1] Centers for Disease Control and Prevention, "National diabetes statistics report: Estimates of diabetes and its burden in the united states, 2014," U.S. Department of Health and Human Services, Atlanta, GA, 2014.
- [2] L. B. Nguyen, S. S. H. Ling, T. W. Jones and H. T. Nguyen. Identification of hypoglycemic states for patients with T1DM using various parameters derived from EEG signals. Presented at Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE. 2011, . DOI: 10.1109/IEMBS.2011.6090756.
- [3] L. B. Nguyen, A. V. Nguyen, Sai Ho Ling and H. T. Nguyen. A particle swarm optimization-based neural network for detecting nocturnal hypoglycemia using electroencephalography signals. Presented at Neural Networks (IJCNN), the 2012 International Joint Conference On. 2012, . DOI: 10.1109/IJCNN.2012.6252745.
- [4] L. B. Nguyen, A. V. Nguyen, Sai Ho Ling and H. T. Nguyen. Analyzing EEG signals under insulin-induced hypoglycemia in type 1 diabetes patients. Presented at Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE. 2013, . DOI: 10.1109/EMBC.2013.6609917.
- [5] L. B. Nguyen, A. V. Nguyen, Sai Ho Ling and H. T. Nguyen. Combining genetic algorithm and levenberg-marquardt algorithm in training neural network for hypoglycemia detection using EEG signals. Presented at Engineering in Medicine and Biology Society (EMBC), 2013 35th Annual International Conference of the IEEE. 2013, . DOI: 10.1109/EMBC.2013.6610766.
- [6] Gary Scheiner. (). *Strike the Spike II - Diabetes Self-Management* URL: <http://www.diabetesselfmanagement.com/managing-diabetes/blood-glucose-management/strike-the-spike-ii/> Accessed: 5/31/2015 .
- [7] Abbott, "Freestyle Freedom Lite," .
- [8] トマトン124, "Electrodes of International 10-20 system EEG," 2010. [https://en.wikipedia.org/wiki/10-20_system_\(EEG\)#/media/File:21_electrodes_of_International_10-20_system_for_EEG.svg](https://en.wikipedia.org/wiki/10-20_system_(EEG)#/media/File:21_electrodes_of_International_10-20_system_for_EEG.svg)
- [9] Herbert Jasper. Report of the committee on methods of clinical examination in electroencephalography. *Electroencephalogr. Clin. Neurophysiol.* 10(2), pp. 370 <last_page> 375. 1958. . DOI: 10.1016/0013-4694(58)90053-1.
- [10] E. Niedermeyer and F. H. Lopes da Silva. *Electroencephalography Basic Principles, Clinical Applications, and Related Fields* 2005 Available: E-Books (Ovid) Access this publication online [http://ezproxy.ballarat.edu.au/login?url=http://ovidsp.ovid.com/ovidweb.cgi?T=JS&PAGE=booktext&NEWS=N&DF=bookdb&AN=01222988/5th_Edition/8&XPATH=/PG\(0\).](http://ezproxy.ballarat.edu.au/login?url=http://ovidsp.ovid.com/ovidweb.cgi?T=JS&PAGE=booktext&NEWS=N&DF=bookdb&AN=01222988/5th_Edition/8&XPATH=/PG(0).)
- [11] R. Oostenveld and P. Praamstra. The five percent electrode system for high-resolution EEG and ERP measurements. *Clinical Neurophysiology* 112(4), pp. 713 <last_page> 719. 2001. . DOI: 10.1016/S1388-2457(00)00527-7.
- [12] Emotiv, "EMOTIV EPOC BRAIN COMPUTER INTERFACE & SCIENTIFIC CONTEXTUAL EEG," Emotiv, 2014.
- [13] American Diabetes Association. (). *Hyperglycemia - High Blood Glucose (Blood Sugar): American Diabetes Association®* URL: <http://www.diabetes.org/living-with-diabetes/treatment-and-care/blood-glucose-control/hyperglycemia.html> Accessed: 5/31/2015 .

- [14] P. E. Cryer, "Hypoglycemia" In Jefferson L. Cherrington A. Goodman H. eds. for the American Physiological Society, " *Handbook of Physiology*, pp. 1057-1092, 2001.
- [15] N. H. White, D. A. Skor, P. E. Cryer, L. A. Levandoski, D. M. Bier and J. V. Santiago. Identification of type I diabetic patients at increased risk for hypoglycemia during intensive therapy. *N. Engl. J. Med.* 308(9), pp. 485-491. 1983. Available: <http://dx.doi.org/10.1056/NEJM198303033080903>. DOI: 10.1056/NEJM198303033080903.
- [16] National Diabetes Information Clearinghouse. (). *Hypoglycemia - National Diabetes Information Clearinghouse* URL: <http://diabetes.niddk.nih.gov/dm/pubs/hypoglycemia/> Accessed: 5/31/2015 .
- [17] David Zieve. (). *Diabetes and nerve damage: MedlinePlus Medical Encyclopedia* URL: <http://www.nlm.nih.gov/medlineplus/ency/article/000693.htm> Accessed: 5/31/2015 .
- [18] Michael Dansinger. (). *Eye Problems and Diabetes: Cataracts, Glaucoma, and Diabetic Retinopathy* URL: <http://www.webmd.com/eye-health/eye-problems> Accessed: 5/31/2015 .
- [19] National Eye Institute. (). *Facts About Glaucoma | National Eye Institute* URL: https://www.nei.nih.gov/health/glaucoma/glaucoma_facts Accessed: 5/31/2015 .
- [20] Healthwise Staff. (). *Diabetic Foot Ulcer* URL: <http://www.webmd.com/diabetes/diabetic-foot-ulcer> Accessed: 5/31/2015 .
- [21] Vincent Lopez Rowe. (). *Diabetic Ulcers* URL: <http://emedicine.medscape.com/article/460282-overview> Accessed: 5/31/2015 .
- [22] HumaSens. (). *HumaSens: Easy self-checks with HumaSens* URL: <http://www.humasens.com/your-humasens-glucose-meter/diabetes-self-checks-with-humasens/> Accessed: 5/31/2015 .
- [23] (2016). *Checking Blood Sugar: Blood Glucose Meter Accuracy* URL: <http://www.diabeticlivingonline.com/monitoring/blood-sugar/checking-blood-sugar-blood-glucose-meter-accuracy?page=0%2C0> Accessed: 6/20/2016 Online .
- [24] (2012, September 5). *Why Meters Can't Tell Us Our Blood Sugar Levels* URL: <http://www.healthline.com/diabetesmine/why-meters-cant-tell-us-our-blood-sugar-levels#1> Accessed: 2016, June 20 Online .
- [25] (2014, January 22). *Glucose Meter Accuracy: FDA's New Dual Call for Tighter Standards* URL: <http://www.healthline.com/diabetesmine/glucose-meter-accuracy-fdas-new-dual-call-for-tighter-standards#1> Accessed: June 6, 2016 Online .
- [26] Mike Hoskins. (). *Glucose Glass? Google Glass technology mixes with diabetes data* URL: <http://www.healthline.com/diabetesmine/glimpsing-diabetes-data-through-glucose-glass> Accessed: 5/31/2015 .
- [27] Tim Locke. (). *Google Crafting Contact Lenses for Diabetes* URL: <http://www.webmd.com/diabetes/news/20140117/contact-lenses-diabetes> Accessed: 5/31/2015 .
- [28] C. O'Donnell and N. Efron. Diabetes and contact lens wear. *Clinical and Experimental Optometry* 95(3), pp. 328-337. 2012. . DOI: 10.1111/j.1444-0938.2012.00738.x.
- [29] Honor Whiteman. (). *New breathalyzer 'monitors blood glucose' in diabetics - Medical News Today* URL: <http://www.medicalnewstoday.com/articles/268726.php> Accessed: 5/31/2015 .

- [30] Lawrence Taylor. (). *Diabetes and the Counterfeit DUI - Avvo.com* URL: <http://www.avvo.com/legal-guides/ugc/diabetes-and-the-counterfeit-dui> Accessed: 5/31/2015 .
- [31] G. Hastings, N. Ghevondian and H. Nguyen. A self-organising fuzzy estimator for hypoglycaemia monitoring in diabetic patients. Presented at Engineering in Medicine and Biology Society, 1998. Proceedings of the 20th Annual International Conference of the IEEE. 1998, . DOI: 10.1109/IEMBS.1998.747135.
- [32] D. Lange, "EEG Spike Waves," 2005. <https://en.wikipedia.org/wiki/Spike-and-wave#/media/File:Spike-waves.png>
- [33] O. EEG-OLOFSSON. Hypoglycemia and neurological disturbances in children with diabetes mellitus. *Acta Paediatrica* 66(S270), pp. 91-96. 1977.
- [34] C. H. Vandyke, "Principle Fissures and Lobes of the Cerebellum Viewed Laterally," Before 1858.
- [35] (2016). *Frontal Lobes*. Available: <http://www.neuroskills.com/brain-injury/frontal-lobes.php>.
- [36] (2016). *Parietal Lobes*. Available: <http://www.neuroskills.com/brain-injury/parietal-lobes.php>.
- [37] J. Downs, R. F. Harrison and S. S. Cross, "A Decision Support Tool for the Diagnosis of Breast Cancer Based upon Fuzzy ARTMAP," *Neural Comput & Applic*, vol. 7, pp. 147-165, 1998.
- [38] L. Ohno-Machado and D. Bialek, "Diagnosing breast cancer from FNAs: variable relevance in neural network and logistic regression models." *Studies in Health Technology and Informatics*, pp. 6/20/2016, .
- [39] S. Karkanis, G. D. Magoulas and N. Theofanous, "Image Recognition and Neuronal Networks: Intelligent Systems for the Improvement of Imaging Information," *Minimal Invasive Therapy and Allied Technologies*, vol. 9, pp. 225-230, 2000.
- [40] P. Lasch, E. N. Lewis, L. H. Kiddler and D. Naumann, "Image reconstruction of FT-IR microspectrometric data," in *Spectral Imaging: Instrumentation, Applications, and Analysis*, San Jose, 2000, .
- [41] R. Cisi and E. Cabral, "Human gait analysed by an artificial neural network model," in *Brazilian Conference on Neural Networks*, São José dos Campos, 1999, pp. 148-151.
- [42] Y. Suzuki, "Self-organizing QRS-Wave Recognition in ECG Using Neural Networks," *IEEE Transactions on Neural Networks*, vol. 6, pp. 1469-1477, 1995.
- [43] Y. H. Hu, W. J. Tompkins and Q. Xue, "Artificial neural network for ECG arrhythmia monitoring," in *International Joint Conference on Neural Network*, 1992, pp. 987-991.
- [44] (2016). *McDonalds Nutrition Information*. Available: <https://www.mcdonalds.com/us/en-us/about-our-food/nutrition-calculator.html>.
- [45] (2016). *Coca-Cola Product Facts*. Available: <http://www.coca-colaproductfacts.com/en/coca-cola-products/coca-cola/>.

APPENDIX A: IRB Approval Letters



EAST CAROLINA UNIVERSITY
University & Medical Center Institutional Review Board Office
4N-70 Brody Medical Sciences Building · Mail Stop 682
600 Moye Boulevard · Greenville, NC 27834
Office 252-744-2914 · Fax 252-744-2284 · www.ecu.edu/irb

Notification of Initial Approval: Expedited

From: Biomedical IRB
To: [Bryce Cranwell](#)
CC: [Ricky Castles](#)
Date: 9/5/2015
Re: [UMCIRB 15-000125](#)
Utilizing Electroencephalography to Detect Changes in Brain Activity During Hyperglycemia

I am pleased to inform you that your Expedited Application was approved. Approval of the study and any consent form(s) is for the period of 9/5/2015 to 9/4/2016. The research study is eligible for review under expedited category #2,4. The Chairperson (or designee) deemed this study no more than minimal risk.

Changes to this approved research may not be initiated without UMCIRB review except when necessary to eliminate an apparent immediate hazard to the participant. All unanticipated problems involving risks to participants and others must be promptly reported to the UMCIRB. The investigator must submit a continuing review/closure application to the UMCIRB prior to the date of study expiration. The Investigator must adhere to all reporting requirements for this study.

Approved consent documents with the IRB approval date stamped on the document should be used to consent participants (consent documents with the IRB approval date stamp are found under the Documents tab in the study workspace).

The approval includes the following items:

Name	Description
EEG_Glucose Survey.docx	Surveys and Questionnaires
Informed Consent Document	Consent Forms
Thesis Pre-Proposal - Cranwell.docx	Study Protocol or Grant Application

The Chairperson (or designee) does not have a potential for conflict of interest on this study.



EAST CAROLINA UNIVERSITY
University & Medical Center Institutional Review Board Office
 4N-70 Brody Medical Sciences Building · Mail Stop 682
 600 Moyer Boulevard · Greenville, NC 27834
 Office 252-744-2914 · Fax 252-744-2284 · www.ecu.edu/irb

Notification of Amendment Approval

From: Biomedical IRB
 To: [Bryce Cranwell](#)
 CC: [Ricky Castles](#)
 Date: 12/14/2015
 Re: [Ame1_UMCIRB 15-000125](#)
[UMCIRB 15-000125](#)
 Utilizing Electroencephalography to Detect Changes in Brain Activity During Hyperglycemia

Your Amendment has been reviewed and approved using expedited review for the period of 12/11/2015 to 9/4/2016. It was the determination of the UMCIRB Chairperson (or designee) that this revision does not impact the overall risk/benefit ratio of the study and is appropriate for the population and procedures proposed.

Please note that any further changes to this approved research may not be initiated without UMCIRB review except when necessary to eliminate an apparent immediate hazard to the participant. All unanticipated problems involving risks to participants and others must be promptly reported to the UMCIRB. A continuing or final review must be submitted to the UMCIRB prior to the date of study expiration. The investigator must adhere to all reporting requirements for this study.

Approved consent documents with the IRB approval date stamped on the document should be used to consent participants (consent documents with the IRB approval date stamp are found under the Documents tab in the study workspace).

The approval includes the following items:

Document	Description
EEG_Glucose Survey.docx(0.05)	Surveys and Questionnaires
Informed Consent Document(0.07)	Consent Forms
Thesis Pre-Proposal - Cranwell.docx(0.03)	Study Protocol or Grant Application

The Chairperson (or designee) does not have a potential for conflict of interest on this study.

IRB00000705 East Carolina UIRB #1 (Biomedical) IORG000418
 IRB00003781 East Carolina UIRB #2 (Behavioral/SS) IORG000418

APPENDIX B: Participant Surveys Taken Before Each Data Recording Session

Bryce A. Cranwell

VSZ #1 10/21/15

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

25

2. What is your estimated height and weight?

6' 00" 160lbs

3. Please briefly describe your diet today before your participation in this experiment.

Did not eat

4. Have you exercised this morning before attending this recording session?

No.

5. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No.

6. Are you on any blood thinners?

No.

7. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

sleep 11:00 - 12:00 am 4x
pm

Awake 8:00 am

P12Q

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

25

2. What is your sex?

M

3. What is your estimated height and weight?

6'00" 165 lbs

4. Please briefly describe your diet today before your participation in this experiment.

Fasted

5. Have you exercised this morning before attending this recording session?

No, last night.

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

12:00am - 8:00am wake up 2-3 times.

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

25

2. What is your sex?

M

3. What is your estimated height and weight?

6'00 165lbs

4. Please briefly describe your diet today before your participation in this experiment.

Fasted

5. Have you exercised this morning before attending this recording session?

No.

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No.

7. Are you on any blood thinners?

No.

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

Sleep ~12:00 am Wake up ~7:30

Wake up 2-3 x per night.

2

Brain Activity and Hyperglycemia Survey Questions Visit #1

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you! 10/2/15

1. What is your age?

34

2. What is your estimated height and weight?

5'10" 290 lbs

3. Please briefly describe your diet today before your participation in this experiment.

Granola, 2% milk, banana, toast, cream cheese

4. Have you exercised this morning before attending this recording session?

Yes, lightly

5. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

6. Are you on any blood thinners?

No

7. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

Usually go to bed between midnight/1AM, up about 6:30-7AM. Might wake once during nights, sometimes not at all

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

34

2. What is your estimated height and weight?

5' 10" 290 lbs

3. Please briefly describe your diet today before your participation in this experiment.

eggs, sausage, water, granola bar

4. Have you exercised this morning before attending this recording session?

No

5. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

6. Are you on any blood thinners?

No

7. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

Varies, usually 6 or so hours of sleep per night with 1 wake up last night went to bed late (3AM) woke at 7.30AM

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

34

2. What is your sex?

M

3. What is your estimated height and weight?

5'10" 290 lbs

4. Please briefly describe your diet today before your participation in this experiment.

Eggs, turkey bacon, milk, bagel, Peanut Butter

5. Have you exercised this morning before attending this recording session?

No

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

Bed around midnight, up at 7AM

Participant 3 16/20/15

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

23

2. What is your estimated height and weight?

165 5'11"

3. Please briefly describe your diet today before your participation in this experiment.

Mc Donalds Mc Griddle & Hashbrown small sip of Dr. Pepper

4. Have you exercised this morning before attending this recording session?

No

5. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

6. Are you on any blood thinners?

No

7. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

Sleep from 12:00 am - 8:30 am 8.5 hrs. Heavy sleeper/No interruptions

P3VQ

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

23 yrs

2. What is your estimated height and weight?

5ft 11³/₄ in. 170 lbs

3. Please briefly describe your diet today before your participation in this experiment.

Fasting

4. Have you exercised this morning before attending this recording session?

No

5. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

6. Are you on any blood thinners?

No

7. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

12 AM - 8 AM No interruptions

P3V3

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

23

2. What is your sex?

Male

3. What is your estimated height and weight?

5'11" 171.6 lbs

4. Please briefly describe your diet today before your participation in this experiment.

plain oatmeal, O.V., asparagus

5. Have you exercised this morning before attending this recording session?

NO

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

NO

7. Are you on any blood thinners?

NO

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

sleep \approx 7.5hrs

12:00am - 9:00am off and on

wake up once a night

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

24

2. What is your sex?

Male

3. What is your estimated height and weight?

5'11" 175 lbs

4. Please briefly describe your diet today before your participation in this experiment.

Water, no food

5. Have you exercised this morning before attending this recording session?

No

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

10pm - 8am

P4V1

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age? sex?

32 , Female

2. What is your estimated height and weight?

5'2" , 110 lb

3. Please briefly describe your diet today before your participation in this experiment.

coffee w/ cream
1 banana
1 piece of toast

4. Have you exercised this morning before attending this recording session?

NO

5. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

NO

6. Are you on any blood thinners?

NO

7. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

10pm - go to sleep
6am - wake up
Don't typically wake up during the night

P2
P4V2

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

32

2. What is your sex?

Female

3. What is your estimated height and weight?

5' 2" 113

4. Please briefly describe your diet today before your participation in this experiment.

banana
cereal w/ almond milk

5. Have you exercised this morning before attending this recording session?

No

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

10pm sleep
6am wake
1 time

p24v3

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

32

2. What is your sex?

F

3. What is your estimated height and weight?

5'2" 113lb

4. Please briefly describe your diet today before your participation in this experiment.

plum, eggs, coffee w/cream

5. Have you exercised this morning before attending this recording session?

Negative

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

10pm 6am

once at most

P5V1

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age^{sex}?

23 Male

2. What is your estimated height and weight?

6'0" 225 lbs

3. Please briefly describe your diet today before your participation in this experiment.

Fasted

4. Have you exercised this morning before attending this recording session?

No

5. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

6. Are you on any blood thinners?

No

7. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

Midnight to 9 A.M. Yes 2-3 times

75V2

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

23

2. What is your sex?

M

3. What is your estimated height and weight?

6'0" 220 lbs

4. Please briefly describe your diet today before your participation in this experiment.

Water

5. Have you exercised this morning before attending this recording session?

No

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

12 a.m. → 8 a.m. 2-3 times

P5V3

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

23

2. What is your sex?

M

3. What is your estimated height and weight?

6' 10" 220 lbs

4. Please briefly describe your diet today before your participation in this experiment.

Water

5. Have you exercised this morning before attending this recording session?

No

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns -- what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

midnight to 8:00 A.M

wake up 2-3 times

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

24

2. What is your sex?

Female

3. What is your estimated height and weight?

5-06

4. Please briefly describe your diet today before your participation in this experiment.

13916

fasted

5. Have you exercised this morning before attending this recording session?

No

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

I go to sleep at around 11pm and wake up at around 7am. I usually don't wake up during the night.

~~P6V2~~ P6V2

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age? 25
2. What is your sex? female
3. What is your estimated height and weight?
5'4 142lb
4. Please briefly describe your diet today before your participation in this experiment. Had a muffin
5. Have you exercised this morning before attending this recording session?
No
6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?
No
7. Are you on any blood thinners?
No
8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?
Go to bed around 11p.m. Don't usually wake up during the night. Wake up around 7am

P7V1

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

27

2. What is your sex?

MALE

3. What is your estimated height and weight?

5'10" AND 207

4. Please briefly describe your diet today before your participation in this experiment.

PANCAKES, FRENCH EGGS, TOAST & 2 CUPS OF COFFEE

5. Have you exercised this morning before attending this recording session?

YES

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

NO

7. Are you on any blood thinners?

NO

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

10 PM - 4:30 PM, NO WAKE DURING SLEEP

P7VA2

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

27

2. What is your sex?

MALE

3. What is your estimated height and weight?

5'10" & 200

4. Please briefly describe your diet today before your participation in this experiment.

SMOOTHIE w/ SEEDS

5. Have you exercised this morning before attending this recording session?

NO

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

NO

7. Are you on any blood thinners?

NO

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

10 PM - 4:30 AM, NO WAKING UP
DURING THE NIGHT

P7V3

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

27

2. What is your sex?

MALE

3. What is your estimated height and weight?

5'10" & 208

4. Please briefly describe your diet today before your participation in this experiment.

FASTING SINCE 6 PM YESTERDAY

5. Have you exercised this morning before attending this recording session?

NO

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

NO

7. Are you on any blood thinners?

NO

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

go to sleep @ ~ 11 PM

wake up @ ~ 7:30 AM

↳ φ

P8V1

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age? 23
2. What is your sex? Male
3. What is your estimated height and weight?
5'8", 145
4. Please briefly describe your diet today before your participation in this experiment.
None
5. Have you exercised this morning before attending this recording session?
Yes,
6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?
No
7. Are you on any blood thinners?
No
8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?
Usually go to sleep around 12AM-1AM, wake up around 8AM

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

35

2. What is your sex?

F

3. What is your estimated height and weight?

5ft 3in, ~250lb

4. Please briefly describe your diet today before your participation in this experiment.

Taco salad, ~ 2.5 cups of coffee, Diet Coke, water

5. Have you exercised this morning before attending this recording session?

Nope

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

↳ ~10mg escitalopram (migraine preventative)

7. Are you on any blood thinners?

Nope.

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

usually sleep ~ 10³⁰/11⁰⁰ - 6³⁰/7⁰⁰

* yes - 1-3x/night

Last night: 5³⁰ - 8³⁰ pm sleep, awake until ~ midnight,
then slept until ~ 8¹⁵ am

P9V2

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

35

2. What is your sex?

Female

3. What is your estimated height and weight?

255, 63in tall

4. Please briefly describe your diet today before your participation in this experiment.

cereal, coffee, pizza, water

5. Have you exercised this morning before attending this recording session?

NO

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

escitalopram

7. Are you on any blood thinners?

NO

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

last night ~ 11 pm - 7:30 pm

P10V1

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

21

2. What is your sex?

Male

3. What is your estimated height and weight?

6'1" , 190 lbs

4. Please briefly describe your diet today before your participation in this experiment.

Coffee. ≈ 2 cups + creamer

5. Have you exercised this morning before attending this recording session?

No

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

12:00 pm → 6:00 am

P10V2

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

21

2. What is your sex?

Male

3. What is your estimated height and weight?

6'1" 200 lbs

4. Please briefly describe your diet today before your participation in this experiment.

Eggs - 2, coffee → 3 cups

5. Have you exercised this morning before attending this recording session?

No

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

12:00 pm → 6:30 am

P10V3

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

AA 21

2. What is your sex?

Male

3. What is your estimated height and weight?

6'1" 205 lb

4. Please briefly describe your diet today before your participation in this experiment.

Coffee

5. Have you exercised this morning before attending this recording session?

No

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

12:00 pm → 6:30 am

PII VI

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?

25

2. What is your sex?

Female

3. What is your estimated height and weight?

5' 4"

4. Please briefly describe your diet today before your participation in this experiment.

Breakfast → 1 scrambled egg w/ cheddar cheese, salsa, and a few bits of steak + water

5. Have you exercised this morning before attending this recording session?

No

6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?

No

7. Are you on any blood thinners?

No

8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

Go to sleep usually around 12 AM

Wake up at about 7 AM

I don't usually wake up at night.

Brain Activity and Hyperglycemia Survey Questions

Please read and answer the following questions. If you are uncomfortable answering any question on the survey please leave that question blank. Thank you!

1. What is your age?
2. What is your sex?
3. What is your estimated height and weight?
4. Please briefly describe your diet today before your participation in this experiment. Breakfast → 1 egg, 1 slice of cheddar cheese, 1 low-carb tortilla
water
5. Have you exercised this morning before attending this recording session?
No
6. Are you on any medication that you feel may influence your brain activity, such as antidepressants, or medication to help you focus?
7. Are you on any blood thinners?
8. Can you describe your sleep patterns – what time do you usually go to sleep and wake up? Do you wake up during the night and if so, how many times?

APPENDIX C: Utilizing Brain Activity to Detect Blood Glucose Levels

BGL Code For R Coefficient Preparation.m

```
% This Code Prepares data to be trained and analyzed

% Author: Bryce A. Cranwell

% Institution: East Carolina University

%%

% Loading participant data sets

clear; close all; clc

% Loading Participant # Recordings

Recording_1 = load('Participant_1_Visit_1.mat');
Recording_2 = load('Participant_2_Visit_2.mat');
Recording_3 = load('Participant_3_Visit_3.mat');

%%

% Setting up variables that will be called further down in the code

fs = 128; % Sampling Frequency

overlap = 0.9375; % Amount that the Spectrogram Overlaps the Previous Window .50 stands for 50% overlap

length1 = length(Recording_1.ED_AF3); % Length of Recording 1
length2 = length(Recording_2.ED_AF3); % Length of Recording 2
length3 = length(Recording_3.ED_AF3); % Length of Recording 3

NFFT_Length = 2^nextpow2(length1);

time_window = 16; % Length of time, in seconds, that the Spectrogram takes into consideration. 16 is 16 seconds.

m = 1; % Counter used for EEG data

M = 1; % Counter used for BGL data

n = 1:1:length1;

n_2 = 1:1:length2;

n_3 = 1:1:length3;
```

```

n_1 = 1:1:length1/fs;

EEG_Index = 1; %EEG Index

BGL_Index = 1; % BGL Index

Length_Window = (fs*time_window); %Length of the Window used for STFT

freq = fs*(0:(Length_Window/2))/Length_Window; %Used to change FFT from indices/amplitude to
freq/amplitude

freq = freq';

Recording_2.ED_P8 = Recording_2.ED_P7;

%Time vectors to graphically display X axis in time, rather than index
%points.

time_vector1 = 1:(1/(60*fs)):(length1*(1/(128*60))+(1-(1/(60*fs))));
time_vector2 = 1:(1/(60*fs)):(length2*(1/(128*60))+(1-(1/(60*fs))));
time_vector3 = 1:(1/(60*fs)):(length3*(1/(128*60))+(1-(1/(60*fs))));

%%

%Setting up the theoretical index points where BGL is supposedly taken.
%Each participant takes BGL in ~15 minute increments, that is what is
%assumed by this portion of the code.

%Indices where First BGL data is theoretically taken.

X_1 = 1;
X_5 = fs*5*60;
X_10 = fs*10*60;
X_15 = fs*15*60;
X_30 = fs*30*60;
X_45 = fs*45*60;
X_50 = fs*50*60;

```

\n
X_55 = fs*55*60;
X_END1 = length1;

%Indicies where Second BGL data is theoretically taken.

X_61 = 1;
X_65 = fs*5*60;
X_70 = fs*10*60;
X_75 = fs*15*60;
X_90 = fs*30*60;
X_105 = fs*45*60;
X_110 = fs*50*60;
X_115 = fs*55*60;
X_END2 = length2;

% Parts of this section of code is suppressed only

% because during this segment there was a lot of noise in the EEG

% recording for this particular participant. Typically this is unsurpressed.

%Indicies where Third BGL data is theoretically taken.

X_121 = 1;
X_125 = fs*5*60;
X_130 = fs*10*60;
X_135 = fs*15*60;
X_150 = fs*30*60;
X_165 = fs*45*60;
X_170 = fs*50*60;
X_175 = fs*55*60;

% X_135 = 1;

```

% X_150 = fs*15*60;
% X_165 = fs*30*60;
% X_170 = fs*35*60;
% X_175 = fs*40*60;
X_END3 = length3;
%%
%BGL Measurements from Freestyle Freedom Lite, These Need to be changed based on Individual Person, or
%changed to be prompted from user.
BGL_T_0 = 94;
BGL_T_5 = 94;
BGL_T_10 = 94;
BGL_T_15 = 94;
BGL_T_30 = 106;
BGL_T_45 = 96;
BGL_T_60 = 92;
BGL_T_50 = ((BGL_T_60 - BGL_T_45)/(X_END1 - X_45))*(X_50 - X_45) + BGL_T_45;
BGL_T_55 = ((BGL_T_60 - BGL_T_45)/(X_END1 - X_45))*(X_55 - X_45) + BGL_T_45;

%Second BGL data
BGL_T_61 = 91;
BGL_T_65 = 91;
BGL_T_70 = 91;
BGL_T_75 = 91;
BGL_T_90 = 102;
BGL_T_105 = 124;
BGL_T_120 = 102;
BGL_T_110 = ((BGL_T_120 - BGL_T_105)/(X_END2 - X_105))*(X_110 - X_105) + BGL_T_105;
BGL_T_115 = ((BGL_T_120 - BGL_T_105)/(X_END2 - X_105))*(X_115 - X_105) + BGL_T_105;

```

\

`%Third BGL data`

`BGL_T_121 = 88;`

`BGL_T_125 = 88;`

`BGL_T_130 = 88;`

`BGL_T_135 = 88;`

`BGL_T_150 = 111;`

`BGL_T_165 = 97;`

`BGL_T_180 = 86;`

`BGL_T_170 = ((BGL_T_180 - BGL_T_165)/(X_END3 - X_165))*(X_170 - X_165) + BGL_T_165;`

`BGL_T_175 = ((BGL_T_180 - BGL_T_165)/(X_END3 - X_165))*(X_175 - X_165) + BGL_T_165;`

`%%`

`%This section sets up the Polynomial curves that approximate a volunteers`

`%BGLs`

`%BGL Matricies to be used in polyval_1`

`BGL_T_M_1 = [BGL_T_0, BGL_T_5, BGL_T_10, BGL_T_15, BGL_T_30, BGL_T_45, BGL_T_50, BGL_T_55,`

`BGL_T_60];`

`X_M1 = [X_1, X_5, X_10, X_15, X_30, X_45, X_50, X_55, X_END1];`

`%Obtaining polynomial curve_1`

`polyval_1 = polyfit(X_M1, BGL_T_M_1, 7);`

`BGL1 = (polyval_1(1,1).*n.^7) + (polyval_1(1,2).*n.^6) + (polyval_1(1,3).*n.^5) + (polyval_1(1,4).*n.^4) +`

`(polyval_1(1,5).*n.^3) + (polyval_1(1,6).*n.^2) + (polyval_1(1,7).*n) + polyval_1(1,8);`

`BGL1 = BGL1';`

`%BGL Matricies to be used in polyval_2`

```
BGL_T_M_2 = [BGL_T_61, BGL_T_65, BGL_T_70, BGL_T_75, BGL_T_90, BGL_T_105, BGL_T_110,  
BGL_T_115, BGL_T_120];
```

```
X_M2 = [X_61, X_65, X_70, X_75, X_90, X_105, X_110, X_115, X_END2];
```

```
%Obtaining polynomial curve_2
```

```
polyval_2 = polyfit(X_M2, BGL_T_M_2, 7);
```

```
BGL2 = (polyval_2(1,1).*n_2.^7) + (polyval_2(1,2).*n_2.^6) + (polyval_2(1,3).*n_2.^5) +  
(polyval_2(1,4).*n_2.^4) + (polyval_2(1,5).*n_2.^3) + (polyval_2(1,6).*n_2.^2) + (polyval_2(1,7).*n_2) +  
polyval_2(1,8);
```

```
BGL2 = BGL2';
```

```
%BGL Matricies to be used in polyval_3
```

```
BGL_T_M_3 = [BGL_T_121, BGL_T_125, BGL_T_130, BGL_T_135, BGL_T_150, BGL_T_165, BGL_T_170,  
BGL_T_175, BGL_T_180];
```

```
X_M3 = [X_121, X_125, X_130, X_135, X_150, X_165, X_170, X_175, X_END3];
```

```
% BGL_T_M_3 = [BGL_T_135, BGL_T_150, BGL_T_165, BGL_T_170, BGL_T_175, BGL_T_180];
```

```
% X_M3 = [ X_135, X_150, X_165, X_170, X_175, X_END3];
```

```
%Obtaining polynomial curve_3
```

```
polyval_3 = polyfit(X_M3, BGL_T_M_3, 7);
```

```
BGL3 = (polyval_3(1,1).*n_3.^7) + (polyval_3(1,2).*n_3.^6) + (polyval_3(1,3).*n_3.^5) +  
(polyval_3(1,4).*n_3.^4) + (polyval_3(1,5).*n_3.^3) + (polyval_3(1,6).*n_3.^2) + (polyval_3(1,7).*n_3) +  
polyval_3(1,8);
```

```
% BGL3 = (polyval_3(1,1).*n_3.^4) + (polyval_3(1,2).*n_3.^3) + (polyval_3(1,3).*n_3.^2) +  
(polyval_3(1,4).*n_3.^1 + polyval_3(1,5));
```

```
BGL3 = BGL3';
```

```
%%
```

\

%This portion of the code sets up which points are actually recorded by the
%Freestyle freedom lite (measured) and which points are Interpolated
%(Inferred). This is in order to differentiate the points later when
%graphing.

%Measured Points

Meas_1 = [BGL_T_0, BGL_T_30, BGL_T_45, BGL_T_60];

Meas_Index_1 = [X_1, X_30, X_45, X_END1];

Meas_2 = [BGL_T_61, BGL_T_90, BGL_T_105, BGL_T_120];

Meas_Index_2 = [X_61, X_90, X_105, X_END2];

% Meas_3 = [BGL_T_121, BGL_T_150, BGL_T_165, BGL_T_180];

% Meas_Index_3 = [X_121, X_150, X_165, X_END3];

%For when Data is missing from third visit

Meas_3 = [BGL_T_135, BGL_T_150, BGL_T_165, BGL_T_180];

Meas_Index_3 = [X_135, X_150, X_165, X_END3];

%Interpolated Points

Int_1 = [BGL_T_5, BGL_T_10, BGL_T_15, BGL_T_50, BGL_T_55];

Int_Index_1 = [X_5, X_10, X_15, X_50, X_55];

Int_2 = [BGL_T_65, BGL_T_70, BGL_T_75, BGL_T_110, BGL_T_115];

Int_Index_2 = [X_65, X_70, X_75, X_110, X_115];

% Int_3 = [BGL_T_125, BGL_T_130, BGL_T_135, BGL_T_170, BGL_T_175];

% Int_Index_3 = [X_125, X_130, X_135, X_170, X_175];

```

Int_3 = [BGL_T_170, BGL_T_175];

Int_Index_3 = [X_170, X_175];

%%

%Quick Plot to Check that the polynomial fits are good, and no major
%irregularities, i.e spiking to 1000mg/dL of glucose.

%plot to check polynomial fit for Visit 1

figure;
plot(time_vector1, BGL1);
hold on
% plot((X_M1./(fs*60)), BGL_T_M_1, 'bo')
plot((Meas_Index_1 ./ (fs*60)) + 1, Meas_1, 'bo');
plot((Int_Index_1 ./ (fs*60)) + 1, Int_1, 'bx');
title('Interpolated BGL For Visit 1');
ylabel('Blood Glucose Level (mg/dL)')
xlabel('Time (Mins)');

%plot to check polynomial fit for Visit 2

figure;
plot(time_vector2, BGL2);
hold on
% plot((X_M2./(fs*60)), BGL_T_M_2, 'ro')
plot((Meas_Index_2 ./ (fs*60)) + 1, Meas_2, 'ro');
plot((Int_Index_2 ./ (fs*60)) + 1, Int_2, 'rx');
title('Interpolated BGL For Visit 2');
ylabel('Blood Glucose Level (mg/dL)')

```

```

xlabel('Time (Mins)');

%plot to check polynomial fit for Visit 3

figure;

plot(time_vector3, BGL3);

hold on

% plot((X_M3./(fs*60)), BGL_T_M_3, 'go')

plot((Meas_Index_3 ./ (fs*60)) + 1, Meas_3, 'go');

plot((Int_Index_3 ./ (fs*60)) + 1, Int_3, 'gx');

title('Interpolated BGL For Visit 3');

ylabel('Blood Glucose Level (mg/dL)')

xlabel('Time (Mins)');

%%

%This section concatenates all of the curve fit BGLS into one continuous

%recording. effectively making 3 hours of uninterrupted recording.

BGL_Combined = [BGL1 ; BGL2 ; BGL3];

bgl_combined_time_vector = 1:(1/(60*fs)):(length(BGL_Combined)*(1/(128*60))+(1-(1/(60*fs))));

%%

%Checking the polynomial fit for all 3 visits combined.

figure;

plot(bgl_combined_time_vector, BGL_Combined);

hold on

plot((Meas_Index_1 ./ (fs*60)) + 1, Meas_1, 'bo');

plot((Int_Index_1 ./ (fs*60)) + 1, Int_1, 'bx');

plot((X_END1/(fs*60)) + (Meas_Index_2 ./ (fs*60)) + 1, Meas_2, 'ro');

plot((X_END1/(fs*60)) + (Int_Index_2 ./ (fs*60)) + 1, Int_2, 'rx');

plot(((X_END1 + X_END2)/(fs*60)) + (Meas_Index_3 ./ (fs*60)) + 1, Meas_3, 'go');

```

```

plot(((X_END1 + X_END2)/(fs*60)) + (Int_Index_3 ./ (fs*60)) + 1, Int_3, 'gx');
title('Concatinated Blood Glucose Levels for 3 Trials');
xlabel('Time (Mins)');
ylabel('Blood Glucose Level (mg/dL)')

%%

%This portion of the code combines all three EEG recordings into a single 3
%hour long vector.

ED_AF3 = [Recording_1.ED_AF3 ; Recording_2.ED_AF3 ; Recording_3.ED_AF3 ];
ED_AF4 = [Recording_1.ED_AF4 ; Recording_2.ED_AF4 ; Recording_3.ED_AF4 ];
ED_F3 = [Recording_1.ED_F3 ; Recording_2.ED_F3 ; Recording_3.ED_F3 ];
ED_F4 = [Recording_1.ED_F4 ; Recording_2.ED_F4 ; Recording_3.ED_F4 ];
ED_F7 = [Recording_1.ED_F7 ; Recording_2.ED_F7 ; Recording_3.ED_F7 ];
ED_F8 = [Recording_1.ED_F8 ; Recording_2.ED_F8 ; Recording_3.ED_F8 ];
ED_FC5 = [Recording_1.ED_FC5 ; Recording_2.ED_FC5 ; Recording_3.ED_FC5 ];
ED_FC6 = [Recording_1.ED_FC6 ; Recording_2.ED_FC6 ; Recording_3.ED_FC6 ];
ED_O1 = [Recording_1.ED_O1 ; Recording_2.ED_O1 ; Recording_3.ED_O1 ];
ED_O2 = [Recording_1.ED_O2 ; Recording_2.ED_O2 ; Recording_3.ED_O2 ];
ED_P7 = [Recording_1.ED_P7 ; Recording_2.ED_P7 ; Recording_3.ED_P7 ];
ED_P8 = [Recording_1.ED_P8 ; Recording_2.ED_P8 ; Recording_3.ED_P8 ];
ED_T7 = [Recording_1.ED_T7 ; Recording_2.ED_T7 ; Recording_3.ED_T7 ];
ED_T8 = [Recording_1.ED_T8 ; Recording_2.ED_T8 ; Recording_3.ED_T8 ];

ED_AF3 = ED_AF3/max(ED_AF3);
ED_AF4 = ED_AF4/max(ED_AF4);
ED_F3 = ED_F3/max(ED_F3);
ED_F4 = ED_F4/max(ED_F4);
ED_F7 = ED_F7/max(ED_F7);
ED_F8 = ED_F8/max(ED_F8);

```

```

ED_FC5 = ED_FC5/max(ED_FC5);
ED_FC6 = ED_FC6/max(ED_FC6);
ED_O1 = ED_O1/max(ED_O1);
ED_O2 = ED_O2/max(ED_O2);
ED_P7 = ED_P7/max(ED_P7);
ED_P8 = ED_P8/max(ED_P8);
ED_T7 = ED_T7/max(ED_T7);
ED_T8 = ED_T8/max(ED_T8);

length_total = length(ED_AF3);

time_vector_total = 1:(1/(60*fs)):(length_total*(1/(128*60))+(1-(1/(60*fs))));

%%
%BGL Window
while M <= (length_total-(fs*time_window))
AF3_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
AF4_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
F3_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
F4_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
F7_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
F8_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
FC5_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
FC6_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
O1_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
O2_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
P7_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
P8_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);
T7_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window)))/max(BGL_Combined);

```

```

T8_BGL{BGL_Index,1} = BGL_Combined((M:M+(fs*time_window))/max(BGL_Combined));

M = M + (fs*(time_window*overlap));

BGL_Index = BGL_Index+1;

end

%%

%BGL Average

%Portion of Code averages all 16 seconds of BGL data into 1 point

for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))

AVG_AF3_BGL(y,1) = mean(AF3_BGL{y,1}(:,1));

AVG_AF4_BGL(y,1) = mean(AF4_BGL{y,1}(:,1));

AVG_F3_BGL(y,1) = mean(F3_BGL{y,1}(:,1));

AVG_F4_BGL(y,1) = mean(F4_BGL{y,1}(:,1));

AVG_F7_BGL(y,1) = mean(F7_BGL{y,1}(:,1));

AVG_F8_BGL(y,1) = mean(F8_BGL{y,1}(:,1));

AVG_FC5_BGL(y,1) = mean(FC5_BGL{y,1}(:,1));

AVG_FC6_BGL(y,1) = mean(FC6_BGL{y,1}(:,1));

AVG_O1_BGL(y,1) = mean(O1_BGL{y,1}(:,1));

AVG_O2_BGL(y,1) = mean(O2_BGL{y,1}(:,1));

AVG_P7_BGL(y,1) = mean(P7_BGL{y,1}(:,1));

AVG_P8_BGL(y,1) = mean(P8_BGL{y,1}(:,1));

AVG_T7_BGL(y,1) = mean(T7_BGL{y,1}(:,1));

AVG_T8_BGL(y,1) = mean(T8_BGL{y,1}(:,1));

BGL_Matrix = [AVG_AF3_BGL, AVG_AF4_BGL, AVG_F3_BGL, AVG_F4_BGL, AVG_F7_BGL,
AVG_F8_BGL, AVG_FC5_BGL, AVG_FC6_BGL, AVG_O1_BGL, AVG_O2_BGL, AVG_P7_BGL,
AVG_P8_BGL, AVG_T7_BGL, AVG_T8_BGL];

end

%%

```

\

% Median filter is employed to help remove noise artifacts from the

% recorded EEG

%Median Filtering EEG Data to remove Artifacts

f_af3 = medfilt1(ED_AF3, fs/4, 'truncate');

f_af4 = medfilt1(ED_AF4, fs/4, 'truncate');

f_f3 = medfilt1(ED_F3, fs/4, 'truncate');

f_f4 = medfilt1(ED_F4, fs/4, 'truncate');

f_f7 = medfilt1(ED_F7, fs/4, 'truncate');

f_f8 = medfilt1(ED_F8, fs/4, 'truncate');

f_fc5 = medfilt1(ED_FC5, fs/4, 'truncate');

f_fc6 = medfilt1(ED_FC6, fs/4, 'truncate');

f_o1 = medfilt1(ED_O1, fs/4, 'truncate');

f_o2 = medfilt1(ED_O2, fs/4, 'truncate');

f_p7 = medfilt1(ED_P7, fs/4, 'truncate');

f_p8 = medfilt1(ED_P8, fs/4, 'truncate');

f_t7 = medfilt1(ED_T7, fs/4, 'truncate');

f_t8 = medfilt1(ED_T8, fs/4, 'truncate');

%%

%DC component is removed from EEG to only look at frequency and amplitude

%changes within the EEG.

%Removing DC Component

f_af3 = f_af3 - mean(f_af3);

f_af4 = f_af4 - mean(f_af4);

f_f3 = f_f3 - mean(f_f3);

```

f_f4 = f_f4 - mean(f_f4);
f_f7 = f_f7 - mean(f_f7);
f_f8 = f_f8 - mean(f_f8);
f_fc5 = f_fc5 - mean(f_fc5);
f_fc6 = f_fc6 - mean(f_fc6);
f_o1 = f_o1 - mean(f_o1);
f_o2 = f_o2 - mean(f_o2);
f_p7 = f_p7 - mean(f_p7);
f_p8 = f_p8 - mean(f_p8);
f_t7 = f_t7 - mean(f_t7);
f_t8 = f_t8 - mean(f_t8);
%%
%Plot to Show effects of filtering between raw and filtered signals
figure
plot(time_vector_total, ED_O1 - mean(ED_O1), 'b')
hold on
plot(time_vector_total, f_o1, 'g')
title('Time Amplitude Data of O1 Channel')
ylabel('Voltage (mV)');
xlabel('Time (Mins)');
xlim([0 length_total*(1/(128*60))])
%%
%Because Brain activity data lies between ~1 Hz and 40 Hz a bandpass filter
%is employed to allow these components to move through, and to remove all
%other frequencies that are be contained within the signal.

%Band-Pass Filter to remove everything between 0.5Hz and 40Hz
[b, a] = butter(6, [0.0078 .625], 'bandpass');

```

```

f_af3 = filter(b,a,f_af3);
f_af4 = filter(b,a,f_af4);
f_f3 = filter(b,a,f_f3);
f_f4 = filter(b,a,f_f4);
f_f7 = filter(b,a,f_f7);
f_f8 = filter(b,a,f_f8);
f_fc5 = filter(b,a,f_fc5);
f_fc6 = filter(b,a,f_fc6);
f_o1 = filter(b,a,f_o1);
f_o2 = filter(b,a,f_o2);
f_p7 = filter(b,a,f_p7);
f_p8 = filter(b,a,f_p8);
f_t7 = filter(b,a,f_t7);
f_t8 = filter(b,a,f_t8);

%%

% 16 Seconds of Amplitude data
while m <= ((length_total)-(fs*time_window))
A_X1{EEG_Index,1} = (f_af3(m:m+(fs*time_window)));
A_X2{EEG_Index,1} = (f_af4(m:m+(fs*time_window)));
A_X3{EEG_Index,1} = (f_f3(m:m+(fs*time_window)));
A_X4{EEG_Index,1} = (f_f4(m:m+(fs*time_window)));
A_X5{EEG_Index,1} = (f_f7(m:m+(fs*time_window)));
A_X6{EEG_Index,1} = (f_f8(m:m+(fs*time_window)));
A_X7{EEG_Index,1} = (f_fc5(m:m+(fs*time_window)));
A_X8{EEG_Index,1} = (f_fc6(m:m+(fs*time_window)));
A_X9{EEG_Index,1} = (f_o1(m:m+(fs*time_window)));
A_X10{EEG_Index,1} = (f_o2(m:m+(fs*time_window)));

```

```

A_X11{EEG_Index,1} = (f_p7(m:m+(fs*time_window)));
A_X12{EEG_Index,1} = (f_p8(m:m+(fs*time_window)));
A_X13{EEG_Index,1} = (f_t7(m:m+(fs*time_window)));
A_X14{EEG_Index,1} = (f_t8(m:m+(fs*time_window)));
m = m + (fs*(time_window*overlap));
EEG_Index = EEG_Index+1;
end

% Averaged EEG Data
for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))
AVG_A_X1(y,1) = mean(A_X1{y,1}(:,1));
AVG_A_X2(y,1) = mean(A_X2{y,1}(:,1));
AVG_A_X3(y,1) = mean(A_X3{y,1}(:,1));
AVG_A_X4(y,1) = mean(A_X4{y,1}(:,1));
AVG_A_X5(y,1) = mean(A_X5{y,1}(:,1));
AVG_A_X6(y,1) = mean(A_X6{y,1}(:,1));
AVG_A_X7(y,1) = mean(A_X7{y,1}(:,1));
AVG_A_X8(y,1) = mean(A_X8{y,1}(:,1));
AVG_A_X9(y,1) = mean(A_X9{y,1}(:,1));
AVG_A_X10(y,1) = mean(A_X10{y,1}(:,1));
AVG_A_X11(y,1) = mean(A_X11{y,1}(:,1));
AVG_A_X12(y,1) = mean(A_X12{y,1}(:,1));
AVG_A_X13(y,1) = mean(A_X13{y,1}(:,1));
AVG_A_X14(y,1) = mean(A_X14{y,1}(:,1));
end

%%

%FFT of Fourier Transform

m = 1;

```

```

EEG_Index = 1;

%Fourier Transform of EEG data is performed to take data from
%Time/Amplitude to Frequency/Power

while m <= ((length_total)-(fs*time_window))

X1{EEG_Index,1} = fft(f_af3(m:m+(fs*time_window)));
X2{EEG_Index,1} = fft(f_af4(m:m+(fs*time_window)));
X3{EEG_Index,1} = fft(f_f3(m:m+(fs*time_window)));
X4{EEG_Index,1} = fft(f_f4(m:m+(fs*time_window)));
X5{EEG_Index,1} = fft(f_f7(m:m+(fs*time_window)));
X6{EEG_Index,1} = fft(f_f8(m:m+(fs*time_window)));
X7{EEG_Index,1} = fft(f_fc5(m:m+(fs*time_window)));
X8{EEG_Index,1} = fft(f_fc6(m:m+(fs*time_window)));
X9{EEG_Index,1} = fft(f_o1(m:m+(fs*time_window)));
X10{EEG_Index,1} = fft(f_o2(m:m+(fs*time_window)));
X11{EEG_Index,1} = fft(f_p7(m:m+(fs*time_window)));
X12{EEG_Index,1} = fft(f_p8(m:m+(fs*time_window)));
X13{EEG_Index,1} = fft(f_t7(m:m+(fs*time_window)));
X14{EEG_Index,1} = fft(f_t8(m:m+(fs*time_window)));

m = m + (fs*(time_window*overlap));

EEG_Index = EEG_Index+1;

end

%%

%Taking ABS/Length value of FFT

for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))

AF3_P2{y,1} =(abs((X1{y,1}(:,1)))/Length_Window);
AF4_P2{y,1} =(abs((X2{y,1}(:,1)))/Length_Window);
F3_P2{y,1} = (abs((X3{y,1}(:,1)))/Length_Window);
F4_P2{y,1} = (abs((X4{y,1}(:,1)))/Length_Window);

```

$F7_P2\{y,1\} = (\text{abs}((X5\{y,1\}(:,1))))/\text{Length_Window};$
 $F8_P2\{y,1\} = (\text{abs}((X6\{y,1\}(:,1))))/\text{Length_Window};$
 $FC5_P2\{y,1\} = (\text{abs}((X7\{y,1\}(:,1))))/\text{Length_Window};$
 $FC6_P2\{y,1\} = (\text{abs}((X8\{y,1\}(:,1))))/\text{Length_Window};$
 $O1_P2\{y,1\} = (\text{abs}((X9\{y,1\}(:,1))))/\text{Length_Window};$
 $O2_P2\{y,1\} = (\text{abs}((X10\{y,1\}(:,1))))/\text{Length_Window};$
 $P7_P2\{y,1\} = (\text{abs}((X11\{y,1\}(:,1))))/\text{Length_Window};$
 $P8_P2\{y,1\} = (\text{abs}((X12\{y,1\}(:,1))))/\text{Length_Window};$
 $T7_P2\{y,1\} = (\text{abs}((X13\{y,1\}(:,1))))/\text{Length_Window};$
 $T8_P2\{y,1\} = (\text{abs}((X14\{y,1\}(:,1))))/\text{Length_Window};$

$AF3_P1\{y,1\} = AF3_P2\{y,1\}(1:(\text{Length_Window}/2)+1);$
 $AF3_P1\{y,1\}(2:\text{end}-1) = 2*AF3_P1\{y,1\}(2:\text{end}-1);$

$AF4_P1\{y,1\} = AF4_P2\{y,1\}(1:(\text{Length_Window}/2)+1);$
 $AF4_P1\{y,1\}(2:\text{end}-1) = 2*AF4_P1\{y,1\}(2:\text{end}-1);$

$F3_P1\{y,1\} = F3_P2\{y,1\}(1:(\text{Length_Window}/2)+1);$
 $F3_P1\{y,1\}(2:\text{end}-1) = 2*F3_P1\{y,1\}(2:\text{end}-1);$

$F4_P1\{y,1\} = F4_P2\{y,1\}(1:(\text{Length_Window}/2)+1);$
 $F4_P1\{y,1\}(2:\text{end}-1) = 2*F4_P1\{y,1\}(2:\text{end}-1);$

$F7_P1\{y,1\} = F7_P2\{y,1\}(1:(\text{Length_Window}/2)+1);$
 $F7_P1\{y,1\}(2:\text{end}-1) = 2*F7_P1\{y,1\}(2:\text{end}-1);$

$F8_P1\{y,1\} = F8_P2\{y,1\}(1:(\text{Length_Window}/2)+1);$
 $F8_P1\{y,1\}(2:\text{end}-1) = 2*F8_P1\{y,1\}(2:\text{end}-1);$

FC5_P1{y,1} = FC5_P2{y,1}(1:(Length_Window/2)+1);

FC5_P1{y,1}(2:end-1) = 2*FC5_P1{y,1}(2:end-1);

FC6_P1{y,1} = FC6_P2{y,1}(1:(Length_Window/2)+1);

FC6_P1{y,1}(2:end-1) = 2*FC6_P1{y,1}(2:end-1);

O1_P1{y,1} = O1_P2{y,1}(1:(Length_Window/2)+1);

O1_P1{y,1}(2:end-1) = 2*O1_P1{y,1}(2:end-1);

O2_P1{y,1} = O2_P2{y,1}(1:(Length_Window/2)+1);

O2_P1{y,1}(2:end-1) = 2*O2_P1{y,1}(2:end-1);

P7_P1{y,1} = P7_P2{y,1}(1:(Length_Window/2)+1);

P7_P1{y,1}(2:end-1) = 2*P7_P1{y,1}(2:end-1);

P8_P1{y,1} = P8_P2{y,1}(1:(Length_Window/2)+1);

P8_P1{y,1}(2:end-1) = 2*P8_P1{y,1}(2:end-1);

T7_P1{y,1} = T7_P2{y,1}(1:(Length_Window/2)+1);

T7_P1{y,1}(2:end-1) = 2*T7_P1{y,1}(2:end-1);

T8_P1{y,1} = T8_P2{y,1}(1:(Length_Window/2)+1);

T8_P1{y,1}(2:end-1) = 2*T8_P1{y,1}(2:end-1);

end

```

FT = [AF3_P1, AF4_P1, F3_P1, F4_P1, F7_P1, F8_P1, FC5_P1, FC6_P1, O1_P1, O2_P1, P7_P1, P8_P1, T7_P1,
T8_P1];

%%

%Looking at the Frequency Spectrum for one specific channel

figure

plot(freq, O1_P1{1,:})

title('Fast Fourier Transform of O1 Channel');

ylabel('Power Spectrum')

xlabel('Frequency (Hz)')

xlim([0 40])

%%

%This portion of code "breaks" the entire frequency bands into the

%delta/theta/alpha/beta/gamma frequency bands.

%Grabbing Delta (Freq Value <4Hz)

for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))

Delta_AF3{y,1} = AF3_P1{y,1}(1:64);

Delta_AF4{y,1} = AF4_P1{y,1}(1:64);

Delta_F3{y,1} = F3_P1{y,1}(1:64);

Delta_F4{y,1} = F4_P1{y,1}(1:64);

Delta_F7{y,1} = F7_P1{y,1}(1:64);

Delta_F8{y,1} = F8_P1{y,1}(1:64);

Delta_FC5{y,1} = FC5_P1{y,1}(1:64);

Delta_FC6{y,1} = FC6_P1{y,1}(1:64);

Delta_O1{y,1} = O1_P1{y,1}(1:64);

Delta_O2{y,1} = O2_P1{y,1}(1:64);

Delta_P7{y,1} = P7_P1{y,1}(1:64);

```

```
Delta_P8{y,1} = P8_P1{y,1}(1:64);
```

```
Delta_T7{y,1} = T7_P1{y,1}(1:64);
```

```
Delta_T8{y,1} = T8_P1{y,1}(1:64);
```

```
end
```

```
%Grabbing Theta (Freq Value 4-7Hz)
```

```
for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))
```

```
Theta_AF3{y,1} = AF3_P1{y,1}(65:128);
```

```
Theta_AF4{y,1} = AF4_P1{y,1}(65:128);
```

```
Theta_F3{y,1} = F3_P1{y,1}(65:128);
```

```
Theta_F4{y,1} = F4_P1{y,1}(65:128);
```

```
Theta_F7{y,1} = F7_P1{y,1}(65:128);
```

```
Theta_F8{y,1} = F8_P1{y,1}(65:128);
```

```
Theta_FC5{y,1} = FC5_P1{y,1}(65:128);
```

```
Theta_FC6{y,1} = FC6_P1{y,1}(65:128);
```

```
Theta_O1{y,1} = O1_P1{y,1}(65:128);
```

```
Theta_O2{y,1} = O2_P1{y,1}(65:128);
```

```
Theta_P7{y,1} = P7_P1{y,1}(65:128);
```

```
Theta_P8{y,1} = P8_P1{y,1}(65:128);
```

```
Theta_T7{y,1} = T7_P1{y,1}(65:128);
```

```
Theta_T8{y,1} = T8_P1{y,1}(65:128);
```

```
end
```

```
%Grabbing Alpha (Freq Value 8-15Hz)
```

```
for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))
```

```
Alpha_AF3{y,1} = AF3_P1{y,1}(129:256);
```

```
Alpha_AF4{y,1} = AF4_P1{y,1}(129:256);
```

```
Alpha_F3{y,1} = F3_P1{y,1}(129:256);
```

```

Alpha_F4{y,1} = F4_P1{y,1}(129:256);
Alpha_F7{y,1} = F7_P1{y,1}(129:256);
Alpha_F8{y,1} = F8_P1{y,1}(129:256);
Alpha_FC5{y,1} = FC5_P1{y,1}(129:256);
Alpha_FC6{y,1} = FC6_P1{y,1}(129:256);
Alpha_O1{y,1} = O1_P1{y,1}(129:256);
Alpha_O2{y,1} = O2_P1{y,1}(129:256);
Alpha_P7{y,1} = P7_P1{y,1}(129:256);
Alpha_P8{y,1} = P8_P1{y,1}(129:256);
Alpha_T7{y,1} = T7_P1{y,1}(129:256);
Alpha_T8{y,1} = T8_P1{y,1}(129:256);
end

```

%Grabbing Beta (Freq Value 16-31Hz)

```

for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))
Beta_AF3{y,1} = AF3_P1{y,1}(257:512);
Beta_AF4{y,1} = AF4_P1{y,1}(257:512);
Beta_F3{y,1} = F3_P1{y,1}(257:512);
Beta_F4{y,1} = F4_P1{y,1}(257:512);
Beta_F7{y,1} = F7_P1{y,1}(257:512);
Beta_F8{y,1} = F8_P1{y,1}(257:512);
Beta_FC5{y,1} = FC5_P1{y,1}(257:512);
Beta_FC6{y,1} = FC6_P1{y,1}(257:512);
Beta_O1{y,1} = O1_P1{y,1}(257:512);
Beta_O2{y,1} = O2_P1{y,1}(257:512);
Beta_P7{y,1} = P7_P1{y,1}(257:512);
Beta_P8{y,1} = P8_P1{y,1}(257:512);
Beta_T7{y,1} = T7_P1{y,1}(257:512);

```

```
Beta_T8{y,1} = T8_P1{y,1}(257:512);
```

```
end
```

```
%Grabbing Gamma (Freq Value 31+Hz) Note that currently only extends from
```

```
%31-40Hz Anything above 40Hz is discarded as noise.
```

```
for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))
```

```
Gamma_AF3{y,1} = AF3_P1{y,1}(513:641);
```

```
Gamma_AF4{y,1} = AF4_P1{y,1}(513:641);
```

```
Gamma_F3{y,1} = F3_P1{y,1}(513:641);
```

```
Gamma_F4{y,1} = F4_P1{y,1}(513:641);
```

```
Gamma_F7{y,1} = F7_P1{y,1}(513:641);
```

```
Gamma_F8{y,1} = F8_P1{y,1}(513:641);
```

```
Gamma_FC5{y,1} = FC5_P1{y,1}(513:641);
```

```
Gamma_FC6{y,1} = FC6_P1{y,1}(513:641);
```

```
Gamma_O1{y,1} = O1_P1{y,1}(513:641);
```

```
Gamma_O2{y,1} = O2_P1{y,1}(513:641);
```

```
Gamma_P7{y,1} = P7_P1{y,1}(513:641);
```

```
Gamma_P8{y,1} = P8_P1{y,1}(513:641);
```

```
Gamma_T7{y,1} = T7_P1{y,1}(513:641);
```

```
Gamma_T8{y,1} = T8_P1{y,1}(513:641);
```

```
end
```

```
%%
```

```
%Average Delta
```

```
for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))
```

```
AVG_Delta_AF3(y,1) = mean(Delta_AF3{y,1});
```

```
AVG_Delta_AF4(y,1) = mean(Delta_AF4{y,1});
```

```
AVG_Delta_F3(y,1) = mean(Delta_F3{y,1});
```

```
AVG_Delta_F4(y,1) = mean(Delta_F4{y,1});
```

```

AVG_Delta_F7(y,1) = mean(Delta_F7{y,1});
AVG_Delta_F8(y,1) = mean(Delta_F8{y,1});
AVG_Delta_FC5(y,1) = mean(Delta_FC5{y,1});
AVG_Delta_FC6(y,1) = mean(Delta_FC6{y,1});
AVG_Delta_O1(y,1) = mean(Delta_O1{y,1});
AVG_Delta_O2(y,1) = mean(Delta_O2{y,1});
AVG_Delta_P7(y,1) = mean(Delta_P7{y,1});
AVG_Delta_P8(y,1) = mean(Delta_P8{y,1});
AVG_Delta_T7(y,1) = mean(Delta_T7{y,1});
AVG_Delta_T8(y,1) = mean(Delta_T8{y,1});
end

```

% Average Theta

```

for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))
AVG_Theta_AF3(y,1) = mean(Theta_AF3{y,1});
AVG_Theta_AF4(y,1) = mean(Theta_AF4{y,1});
AVG_Theta_F3(y,1) = mean(Theta_F3{y,1});
AVG_Theta_F4(y,1) = mean(Theta_F4{y,1});
AVG_Theta_F7(y,1) = mean(Theta_F7{y,1});
AVG_Theta_F8(y,1) = mean(Theta_F8{y,1});
AVG_Theta_FC5(y,1) = mean(Theta_FC5{y,1});
AVG_Theta_FC6(y,1) = mean(Theta_FC6{y,1});
AVG_Theta_O1(y,1) = mean(Theta_O1{y,1});
AVG_Theta_O2(y,1) = mean(Theta_O2{y,1});
AVG_Theta_P7(y,1) = mean(Theta_P7{y,1});
AVG_Theta_P8(y,1) = mean(Theta_P8{y,1});
AVG_Theta_T7(y,1) = mean(Theta_T7{y,1});
AVG_Theta_T8(y,1) = mean(Theta_T8{y,1});
end

```

end

% Average Alpha

```
for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))
```

```
AVG_Alpha_AF3(y,1) = mean(Alpha_AF3{y,1});
```

```
AVG_Alpha_AF4(y,1) = mean(Alpha_AF4{y,1});
```

```
AVG_Alpha_F3(y,1) = mean(Alpha_F3{y,1});
```

```
AVG_Alpha_F4(y,1) = mean(Alpha_F4{y,1});
```

```
AVG_Alpha_F7(y,1) = mean(Alpha_F7{y,1});
```

```
AVG_Alpha_F8(y,1) = mean(Alpha_F8{y,1});
```

```
AVG_Alpha_FC5(y,1) = mean(Alpha_FC5{y,1});
```

```
AVG_Alpha_FC6(y,1) = mean(Alpha_FC6{y,1});
```

```
AVG_Alpha_O1(y,1) = mean(Alpha_O1{y,1});
```

```
AVG_Alpha_O2(y,1) = mean(Alpha_O2{y,1});
```

```
AVG_Alpha_P7(y,1) = mean(Alpha_P7{y,1});
```

```
AVG_Alpha_P8(y,1) = mean(Alpha_P8{y,1});
```

```
AVG_Alpha_T7(y,1) = mean(Alpha_T7{y,1});
```

```
AVG_Alpha_T8(y,1) = mean(Alpha_T8{y,1});
```

end

% Average Beta

```
for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))
```

```
AVG_Beta_AF3(y,1) = mean(Beta_AF3{y,1});
```

```
AVG_Beta_AF4(y,1) = mean(Beta_AF4{y,1});
```

```
AVG_Beta_F3(y,1) = mean(Beta_F3{y,1});
```

```
AVG_Beta_F4(y,1) = mean(Beta_F4{y,1});
```

```
AVG_Beta_F7(y,1) = mean(Beta_F7{y,1});
```

```
AVG_Beta_F8(y,1) = mean(Beta_F8{y,1});
```

```

AVG_Beta_FC5(y,1) = mean(Beta_FC5{y,1});
AVG_Beta_FC6(y,1) = mean(Beta_FC6{y,1});
AVG_Beta_O1(y,1) = mean(Beta_O1{y,1});
AVG_Beta_O2(y,1) = mean(Beta_O2{y,1});
AVG_Beta_P7(y,1) = mean(Beta_P7{y,1});
AVG_Beta_P8(y,1) = mean(Beta_P8{y,1});
AVG_Beta_T7(y,1) = mean(Beta_T7{y,1});
AVG_Beta_T8(y,1) = mean(Beta_T8{y,1});

```

```
end
```

```
%Average Gamma
```

```
for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))
```

```

AVG_Gamma_AF3(y,1) = mean(Gamma_AF3{y,1});
AVG_Gamma_AF4(y,1) = mean(Gamma_AF4{y,1});
AVG_Gamma_F3(y,1) = mean(Gamma_F3{y,1});
AVG_Gamma_F4(y,1) = mean(Gamma_F4{y,1});
AVG_Gamma_F7(y,1) = mean(Gamma_F7{y,1});
AVG_Gamma_F8(y,1) = mean(Gamma_F8{y,1});
AVG_Gamma_FC5(y,1) = mean(Gamma_FC5{y,1});
AVG_Gamma_FC6(y,1) = mean(Gamma_FC6{y,1});
AVG_Gamma_O1(y,1) = mean(Gamma_O1{y,1});
AVG_Gamma_O2(y,1) = mean(Gamma_O2{y,1});
AVG_Gamma_P7(y,1) = mean(Gamma_P7{y,1});
AVG_Gamma_P8(y,1) = mean(Gamma_P8{y,1});
AVG_Gamma_T7(y,1) = mean(Gamma_T7{y,1});
AVG_Gamma_T8(y,1) = mean(Gamma_T8{y,1});

```

```
end
```

```
%%
```

%The Percent Average of the Different Frequency Bands, rather than looking
%at the power of each of the bands, it is rather a % of all the activity
%that is occurring.

% % Average Delta

for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))

Percent_Delta_AF3(y,1) = mean(Delta_AF3{y,1})/(mean(Delta_AF3{y,1} + mean(Delta_AF4{y,1}) +
mean(Delta_F3{y,1}) + mean(Delta_F4{y,1}) + mean(Delta_F7{y,1}) + mean(Delta_F8{y,1}) +
mean(Delta_FC5{y,1}) + mean(Delta_FC6{y,1}) + mean(Delta_O1{y,1}) + mean(Delta_O2{y,1}) +
mean(Delta_P7{y,1}) + mean(Delta_T7{y,1}) + mean(Delta_T8{y,1})));

Percent_Delta_AF4(y,1) = mean(Delta_AF4{y,1})/(mean(Delta_AF3{y,1} + mean(Delta_AF4{y,1}) +
mean(Delta_F3{y,1}) + mean(Delta_F4{y,1}) + mean(Delta_F7{y,1}) + mean(Delta_F8{y,1}) +
mean(Delta_FC5{y,1}) + mean(Delta_FC6{y,1}) + mean(Delta_O1{y,1}) + mean(Delta_O2{y,1}) +
mean(Delta_P7{y,1}) + mean(Delta_T7{y,1}) + mean(Delta_T8{y,1})));

Percent_Delta_F3(y,1) = mean(Delta_F3{y,1})/(mean(Delta_AF3{y,1} + mean(Delta_AF4{y,1}) +
mean(Delta_F3{y,1}) + mean(Delta_F4{y,1}) + mean(Delta_F7{y,1}) + mean(Delta_F8{y,1}) +
mean(Delta_FC5{y,1}) + mean(Delta_FC6{y,1}) + mean(Delta_O1{y,1}) + mean(Delta_O2{y,1}) +
mean(Delta_P7{y,1}) + mean(Delta_T7{y,1}) + mean(Delta_T8{y,1})));

Percent_Delta_F4(y,1) = mean(Delta_F4{y,1})/(mean(Delta_AF3{y,1} + mean(Delta_AF4{y,1}) +
mean(Delta_F3{y,1}) + mean(Delta_F4{y,1}) + mean(Delta_F7{y,1}) + mean(Delta_F8{y,1}) +
mean(Delta_FC5{y,1}) + mean(Delta_FC6{y,1}) + mean(Delta_O1{y,1}) + mean(Delta_O2{y,1}) +
mean(Delta_P7{y,1}) + mean(Delta_T7{y,1}) + mean(Delta_T8{y,1})));

Percent_Delta_F7(y,1) = mean(Delta_F7{y,1})/(mean(Delta_AF3{y,1} + mean(Delta_AF4{y,1}) +
mean(Delta_F3{y,1}) + mean(Delta_F4{y,1}) + mean(Delta_F7{y,1}) + mean(Delta_F8{y,1}) +
mean(Delta_FC5{y,1}) + mean(Delta_FC6{y,1}) + mean(Delta_O1{y,1}) + mean(Delta_O2{y,1}) +
mean(Delta_P7{y,1}) + mean(Delta_T7{y,1}) + mean(Delta_T8{y,1})));

Percent_Delta_T7(y,1) = mean(Delta_T7{y,1})/(mean(Delta_AF3{y,1} + mean(Delta_AF4{y,1}) +
 mean(Delta_F3{y,1}) + mean(Delta_F4{y,1}) + mean(Delta_F7{y,1}) + mean(Delta_F8{y,1}) +
 mean(Delta_FC5{y,1}) + mean(Delta_FC6{y,1}) + mean(Delta_O1{y,1}) + mean(Delta_O2{y,1}) +
 mean(Delta_P7{y,1}) + mean(Delta_T7{y,1}) + mean(Delta_T8{y,1})));

Percent_Delta_T8(y,1) = mean(Delta_T8{y,1})/(mean(Delta_AF3{y,1} + mean(Delta_AF4{y,1}) +
 mean(Delta_F3{y,1}) + mean(Delta_F4{y,1}) + mean(Delta_F7{y,1}) + mean(Delta_F8{y,1}) +
 mean(Delta_FC5{y,1}) + mean(Delta_FC6{y,1}) + mean(Delta_O1{y,1}) + mean(Delta_O2{y,1}) +
 mean(Delta_P7{y,1}) + mean(Delta_T7{y,1}) + mean(Delta_T8{y,1})));

end

% % Average Theta

for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))

Percent_Theta_AF3(y,1) = mean(Theta_AF3{y,1})/(mean(Theta_AF3{y,1} + mean(Theta_AF4{y,1}) +
 mean(Theta_F3{y,1}) + mean(Theta_F4{y,1}) + mean(Theta_F7{y,1}) + mean(Theta_F8{y,1}) +
 mean(Theta_FC5{y,1}) + mean(Theta_FC6{y,1}) + mean(Theta_O1{y,1}) + mean(Theta_O2{y,1}) +
 mean(Theta_P7{y,1}) + mean(Theta_T7{y,1}) + mean(Theta_T8{y,1})));

Percent_Theta_AF4(y,1) = mean(Theta_AF4{y,1})/(mean(Theta_AF3{y,1} + mean(Theta_AF4{y,1}) +
 mean(Theta_F3{y,1}) + mean(Theta_F4{y,1}) + mean(Theta_F7{y,1}) + mean(Theta_F8{y,1}) +
 mean(Theta_FC5{y,1}) + mean(Theta_FC6{y,1}) + mean(Theta_O1{y,1}) + mean(Theta_O2{y,1}) +
 mean(Theta_P7{y,1}) + mean(Theta_T7{y,1}) + mean(Theta_T8{y,1})));

Percent_Theta_F3(y,1) = mean(Theta_F3{y,1})/(mean(Theta_AF3{y,1} + mean(Theta_AF4{y,1}) +
 mean(Theta_F3{y,1}) + mean(Theta_F4{y,1}) + mean(Theta_F7{y,1}) + mean(Theta_F8{y,1}) +
 mean(Theta_FC5{y,1}) + mean(Theta_FC6{y,1}) + mean(Theta_O1{y,1}) + mean(Theta_O2{y,1}) +
 mean(Theta_P7{y,1}) + mean(Theta_T7{y,1}) + mean(Theta_T8{y,1})));

Percent_Theta_F4(y,1) = mean(Theta_F4{y,1})/(mean(Theta_AF3{y,1} + mean(Theta_AF4{y,1}) +
 mean(Theta_F3{y,1}) + mean(Theta_F4{y,1}) + mean(Theta_F7{y,1}) + mean(Theta_F8{y,1}) +
 mean(Theta_FC5{y,1}) + mean(Theta_FC6{y,1}) + mean(Theta_O1{y,1}) + mean(Theta_O2{y,1}) +
 mean(Theta_P7{y,1}) + mean(Theta_T7{y,1}) + mean(Theta_T8{y,1})));

Percent_Theta_P8(y,1) = mean(Theta_P8{y,1})/(mean(Theta_AF3{y,1} + mean(Theta_AF4{y,1}) +
mean(Theta_F3{y,1}) + mean(Theta_F4{y,1}) + mean(Theta_F7{y,1}) + mean(Theta_F8{y,1}) +
mean(Theta_FC5{y,1}) + mean(Theta_FC6{y,1}) + mean(Theta_O1{y,1}) + mean(Theta_O2{y,1}) +
mean(Theta_P7{y,1}) + mean(Theta_T7{y,1}) + mean(Theta_T8{y,1})));

Percent_Theta_T7(y,1) = mean(Theta_T7{y,1})/(mean(Theta_AF3{y,1} + mean(Theta_AF4{y,1}) +
mean(Theta_F3{y,1}) + mean(Theta_F4{y,1}) + mean(Theta_F7{y,1}) + mean(Theta_F8{y,1}) +
mean(Theta_FC5{y,1}) + mean(Theta_FC6{y,1}) + mean(Theta_O1{y,1}) + mean(Theta_O2{y,1}) +
mean(Theta_P7{y,1}) + mean(Theta_T7{y,1}) + mean(Theta_T8{y,1})));

Percent_Theta_T8(y,1) = mean(Theta_T8{y,1})/(mean(Theta_AF3{y,1} + mean(Theta_AF4{y,1}) +
mean(Theta_F3{y,1}) + mean(Theta_F4{y,1}) + mean(Theta_F7{y,1}) + mean(Theta_F8{y,1}) +
mean(Theta_FC5{y,1}) + mean(Theta_FC6{y,1}) + mean(Theta_O1{y,1}) + mean(Theta_O2{y,1}) +
mean(Theta_P7{y,1}) + mean(Theta_T7{y,1}) + mean(Theta_T8{y,1})));

end

% % Average Alpha

for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))

Percent_Alpha_AF3(y,1) = mean(Alpha_AF3{y,1})/(mean(Alpha_AF3{y,1} + mean(Alpha_AF4{y,1}) +
mean(Alpha_F3{y,1}) + mean(Alpha_F4{y,1}) + mean(Alpha_F7{y,1}) + mean(Alpha_F8{y,1}) +
mean(Alpha_FC5{y,1}) + mean(Alpha_FC6{y,1}) + mean(Alpha_O1{y,1}) + mean(Alpha_O2{y,1}) +
mean(Alpha_P7{y,1}) + mean(Alpha_T7{y,1}) + mean(Alpha_T8{y,1})));

Percent_Alpha_AF4(y,1) = mean(Alpha_AF4{y,1})/(mean(Alpha_AF3{y,1} + mean(Alpha_AF4{y,1}) +
mean(Alpha_F3{y,1}) + mean(Alpha_F4{y,1}) + mean(Alpha_F7{y,1}) + mean(Alpha_F8{y,1}) +
mean(Alpha_FC5{y,1}) + mean(Alpha_FC6{y,1}) + mean(Alpha_O1{y,1}) + mean(Alpha_O2{y,1}) +
mean(Alpha_P7{y,1}) + mean(Alpha_T7{y,1}) + mean(Alpha_T8{y,1})));

Percent_Alpha_F3(y,1) = mean(Alpha_F3{y,1})/(mean(Alpha_AF3{y,1} + mean(Alpha_AF4{y,1}) +
mean(Alpha_F3{y,1}) + mean(Alpha_F4{y,1}) + mean(Alpha_F7{y,1}) + mean(Alpha_F8{y,1}) +
mean(Alpha_FC5{y,1}) + mean(Alpha_FC6{y,1}) + mean(Alpha_O1{y,1}) + mean(Alpha_O2{y,1}) +
mean(Alpha_P7{y,1}) + mean(Alpha_T7{y,1}) + mean(Alpha_T8{y,1})));


```
Percent_Alpha_P7(y,1) = mean(Alpha_P7{y,1})/(mean(Alpha_AF3{y,1} + mean(Alpha_AF4{y,1}) +
mean(Alpha_F3{y,1}) + mean(Alpha_F4{y,1}) + mean(Alpha_F7{y,1}) + mean(Alpha_F8{y,1}) +
mean(Alpha_FC5{y,1}) + mean(Alpha_FC6{y,1}) + mean(Alpha_O1{y,1}) + mean(Alpha_O2{y,1}) +
mean(Alpha_P7{y,1}) + mean(Alpha_T7{y,1}) + mean(Alpha_T8{y,1})));
```

```
Percent_Alpha_P8(y,1) = mean(Alpha_P8{y,1})/(mean(Alpha_AF3{y,1} + mean(Alpha_AF4{y,1}) +
mean(Alpha_F3{y,1}) + mean(Alpha_F4{y,1}) + mean(Alpha_F7{y,1}) + mean(Alpha_F8{y,1}) +
mean(Alpha_FC5{y,1}) + mean(Alpha_FC6{y,1}) + mean(Alpha_O1{y,1}) + mean(Alpha_O2{y,1}) +
mean(Alpha_P7{y,1}) + mean(Alpha_T7{y,1}) + mean(Alpha_T8{y,1})));
```

```
Percent_Alpha_T7(y,1) = mean(Alpha_T7{y,1})/(mean(Alpha_AF3{y,1} + mean(Alpha_AF4{y,1}) +
mean(Alpha_F3{y,1}) + mean(Alpha_F4{y,1}) + mean(Alpha_F7{y,1}) + mean(Alpha_F8{y,1}) +
mean(Alpha_FC5{y,1}) + mean(Alpha_FC6{y,1}) + mean(Alpha_O1{y,1}) + mean(Alpha_O2{y,1}) +
mean(Alpha_P7{y,1}) + mean(Alpha_T7{y,1}) + mean(Alpha_T8{y,1})));
```

```
Percent_Alpha_T8(y,1) = mean(Alpha_T8{y,1})/(mean(Alpha_AF3{y,1} + mean(Alpha_AF4{y,1}) +
mean(Alpha_F3{y,1}) + mean(Alpha_F4{y,1}) + mean(Alpha_F7{y,1}) + mean(Alpha_F8{y,1}) +
mean(Alpha_FC5{y,1}) + mean(Alpha_FC6{y,1}) + mean(Alpha_O1{y,1}) + mean(Alpha_O2{y,1}) +
mean(Alpha_P7{y,1}) + mean(Alpha_T7{y,1}) + mean(Alpha_T8{y,1})));
```

```
end
```

```
% % Average Beta
```

```
for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))
```

```
Percent_Beta_AF3(y,1) = mean(Beta_AF3{y,1})/(mean(Beta_AF3{y,1} + mean(Beta_AF4{y,1}) +
mean(Beta_F3{y,1}) + mean(Beta_F4{y,1}) + mean(Beta_F7{y,1}) + mean(Beta_F8{y,1}) +
mean(Beta_FC5{y,1}) + mean(Beta_FC6{y,1}) + mean(Beta_O1{y,1}) + mean(Beta_O2{y,1}) +
mean(Beta_P7{y,1}) + mean(Beta_T7{y,1}) + mean(Beta_T8{y,1})));
```

```
Percent_Beta_AF4(y,1) = mean(Beta_AF4{y,1})/(mean(Beta_AF3{y,1} + mean(Beta_AF4{y,1}) +
mean(Beta_F3{y,1}) + mean(Beta_F4{y,1}) + mean(Beta_F7{y,1}) + mean(Beta_F8{y,1}) +
mean(Beta_FC5{y,1}) + mean(Beta_FC6{y,1}) + mean(Beta_O1{y,1}) + mean(Beta_O2{y,1}) +
mean(Beta_P7{y,1}) + mean(Beta_T7{y,1}) + mean(Beta_T8{y,1})));
```


Percent_Beta_O2(y,1) = mean(Beta_O2{y,1})/(mean(Beta_AF3{y,1} + mean(Beta_AF4{y,1}) +
 mean(Beta_F3{y,1}) + mean(Beta_F4{y,1}) + mean(Beta_F7{y,1}) + mean(Beta_F8{y,1}) +
 mean(Beta_FC5{y,1}) + mean(Beta_FC6{y,1}) + mean(Beta_O1{y,1}) + mean(Beta_O2{y,1}) +
 mean(Beta_P7{y,1}) + mean(Beta_T7{y,1}) + mean(Beta_T8{y,1})));

Percent_Beta_P7(y,1) = mean(Beta_P7{y,1})/(mean(Beta_AF3{y,1} + mean(Beta_AF4{y,1}) +
 mean(Beta_F3{y,1}) + mean(Beta_F4{y,1}) + mean(Beta_F7{y,1}) + mean(Beta_F8{y,1}) +
 mean(Beta_FC5{y,1}) + mean(Beta_FC6{y,1}) + mean(Beta_O1{y,1}) + mean(Beta_O2{y,1}) +
 mean(Beta_P7{y,1}) + mean(Beta_T7{y,1}) + mean(Beta_T8{y,1})));

Percent_Beta_P8(y,1) = mean(Beta_P8{y,1})/(mean(Beta_AF3{y,1} + mean(Beta_AF4{y,1}) +
 mean(Beta_F3{y,1}) + mean(Beta_F4{y,1}) + mean(Beta_F7{y,1}) + mean(Beta_F8{y,1}) +
 mean(Beta_FC5{y,1}) + mean(Beta_FC6{y,1}) + mean(Beta_O1{y,1}) + mean(Beta_O2{y,1}) +
 mean(Beta_P7{y,1}) + mean(Beta_T7{y,1}) + mean(Beta_T8{y,1})));

Percent_Beta_T7(y,1) = mean(Beta_T7{y,1})/(mean(Beta_AF3{y,1} + mean(Beta_AF4{y,1}) +
 mean(Beta_F3{y,1}) + mean(Beta_F4{y,1}) + mean(Beta_F7{y,1}) + mean(Beta_F8{y,1}) +
 mean(Beta_FC5{y,1}) + mean(Beta_FC6{y,1}) + mean(Beta_O1{y,1}) + mean(Beta_O2{y,1}) +
 mean(Beta_P7{y,1}) + mean(Beta_T7{y,1}) + mean(Beta_T8{y,1})));

Percent_Beta_T8(y,1) = mean(Beta_T8{y,1})/(mean(Beta_AF3{y,1} + mean(Beta_AF4{y,1}) +
 mean(Beta_F3{y,1}) + mean(Beta_F4{y,1}) + mean(Beta_F7{y,1}) + mean(Beta_F8{y,1}) +
 mean(Beta_FC5{y,1}) + mean(Beta_FC6{y,1}) + mean(Beta_O1{y,1}) + mean(Beta_O2{y,1}) +
 mean(Beta_P7{y,1}) + mean(Beta_T7{y,1}) + mean(Beta_T8{y,1})));

end

% % Average Gamma

for y = 1:ceil((length_total-(fs*time_window))/(fs*(time_window*overlap)))

Percent_Gamma_AF3(y,1) = mean(Gamma_AF3{y,1})/(mean(Gamma_AF3{y,1} + mean(Gamma_AF4{y,1}) +
 mean(Gamma_F3{y,1}) + mean(Gamma_F4{y,1}) + mean(Gamma_F7{y,1}) + mean(Gamma_F8{y,1}) +
 mean(Gamma_FC5{y,1}) + mean(Gamma_FC6{y,1}) + mean(Gamma_O1{y,1}) + mean(Gamma_O2{y,1}) +
 mean(Gamma_P7{y,1}) + mean(Gamma_T7{y,1}) + mean(Gamma_T8{y,1})));

Percent_Gamma_O1(y,1) = mean(Gamma_O1{y,1})/(mean(Gamma_AF3{y,1} + mean(Gamma_AF4{y,1}) + mean(Gamma_F3{y,1}) + mean(Gamma_F4{y,1}) + mean(Gamma_F7{y,1}) + mean(Gamma_F8{y,1}) + mean(Gamma_FC5{y,1}) + mean(Gamma_FC6{y,1}) + mean(Gamma_O1{y,1}) + mean(Gamma_O2{y,1}) + mean(Gamma_P7{y,1}) + mean(Gamma_T7{y,1}) + mean(Gamma_T8{y,1})));

Percent_Gamma_O2(y,1) = mean(Gamma_O2{y,1})/(mean(Gamma_AF3{y,1} + mean(Gamma_AF4{y,1}) + mean(Gamma_F3{y,1}) + mean(Gamma_F4{y,1}) + mean(Gamma_F7{y,1}) + mean(Gamma_F8{y,1}) + mean(Gamma_FC5{y,1}) + mean(Gamma_FC6{y,1}) + mean(Gamma_O1{y,1}) + mean(Gamma_O2{y,1}) + mean(Gamma_P7{y,1}) + mean(Gamma_T7{y,1}) + mean(Gamma_T8{y,1})));

Percent_Gamma_P7(y,1) = mean(Gamma_P7{y,1})/(mean(Gamma_AF3{y,1} + mean(Gamma_AF4{y,1}) + mean(Gamma_F3{y,1}) + mean(Gamma_F4{y,1}) + mean(Gamma_F7{y,1}) + mean(Gamma_F8{y,1}) + mean(Gamma_FC5{y,1}) + mean(Gamma_FC6{y,1}) + mean(Gamma_O1{y,1}) + mean(Gamma_O2{y,1}) + mean(Gamma_P7{y,1}) + mean(Gamma_T7{y,1}) + mean(Gamma_T8{y,1})));

Percent_Gamma_P8(y,1) = mean(Gamma_P8{y,1})/(mean(Gamma_AF3{y,1} + mean(Gamma_AF4{y,1}) + mean(Gamma_F3{y,1}) + mean(Gamma_F4{y,1}) + mean(Gamma_F7{y,1}) + mean(Gamma_F8{y,1}) + mean(Gamma_FC5{y,1}) + mean(Gamma_FC6{y,1}) + mean(Gamma_O1{y,1}) + mean(Gamma_O2{y,1}) + mean(Gamma_P7{y,1}) + mean(Gamma_T7{y,1}) + mean(Gamma_T8{y,1})));

Percent_Gamma_T7(y,1) = mean(Gamma_T7{y,1})/(mean(Gamma_AF3{y,1} + mean(Gamma_AF4{y,1}) + mean(Gamma_F3{y,1}) + mean(Gamma_F4{y,1}) + mean(Gamma_F7{y,1}) + mean(Gamma_F8{y,1}) + mean(Gamma_FC5{y,1}) + mean(Gamma_FC6{y,1}) + mean(Gamma_O1{y,1}) + mean(Gamma_O2{y,1}) + mean(Gamma_P7{y,1}) + mean(Gamma_T7{y,1}) + mean(Gamma_T8{y,1})));

Percent_Gamma_T8(y,1) = mean(Gamma_T8{y,1})/(mean(Gamma_AF3{y,1} + mean(Gamma_AF4{y,1}) + mean(Gamma_F3{y,1}) + mean(Gamma_F4{y,1}) + mean(Gamma_F7{y,1}) + mean(Gamma_F8{y,1}) + mean(Gamma_FC5{y,1}) + mean(Gamma_FC6{y,1}) + mean(Gamma_O1{y,1}) + mean(Gamma_O2{y,1}) + mean(Gamma_P7{y,1}) + mean(Gamma_T7{y,1}) + mean(Gamma_T8{y,1})));

end

%%

%Building Matrices that will then be passed into another program to be

%utilized.

% Total Matrix

```
EEG_Matrix = [Percent_Delta_AF3, Percent_Theta_AF3, Percent_Alpha_AF3, Percent_Beta_AF3,  
Percent_Gamma_AF3, Percent_Delta_AF4, Percent_Theta_AF4, Percent_Alpha_AF4, Percent_Beta_AF4,  
Percent_Gamma_AF4, Percent_Delta_F3, Percent_Theta_F3, Percent_Alpha_F3, Percent_Beta_F3,  
Percent_Gamma_F3, Percent_Delta_F4, Percent_Theta_F4, Percent_Alpha_F4, Percent_Beta_F4,  
Percent_Gamma_F4, Percent_Delta_F7, Percent_Theta_F7, Percent_Alpha_F7, Percent_Beta_F7,  
Percent_Gamma_F7, Percent_Delta_F8, Percent_Theta_F8, Percent_Alpha_F8, Percent_Beta_F8,  
Percent_Gamma_F8, Percent_Delta_FC5, Percent_Theta_FC5, Percent_Alpha_FC5, Percent_Beta_FC5,  
Percent_Gamma_FC5, Percent_Delta_FC6, Percent_Theta_FC6, Percent_Alpha_FC6, Percent_Beta_FC6,  
Percent_Gamma_FC6, Percent_Delta_O1, Percent_Theta_O1, Percent_Alpha_O1, Percent_Beta_O1,  
Percent_Gamma_O1, Percent_Delta_O2, Percent_Theta_O2, Percent_Alpha_O2, Percent_Beta_O2,  
Percent_Gamma_O2, Percent_Delta_P7, Percent_Theta_P7, Percent_Alpha_P7, Percent_Beta_P7,  
Percent_Gamma_P7, Percent_Delta_P8, Percent_Theta_P8, Percent_Alpha_P8, Percent_Beta_P8,  
Percent_Gamma_P8, Percent_Delta_T7, Percent_Theta_T7, Percent_Alpha_T7, Percent_Beta_T7,  
Percent_Gamma_T7, Percent_Delta_T8, Percent_Theta_T8, Percent_Alpha_T8, Percent_Beta_T8,  
Percent_Gamma_T8];
```

% Individual Matrices

```
AF3_Matrix = [Percent_Delta_AF3, Percent_Theta_AF3, Percent_Alpha_AF3, Percent_Beta_AF3,  
Percent_Gamma_AF3, AVG_A_X1];  
AF4_Matrix = [Percent_Delta_AF4, Percent_Theta_AF4, Percent_Alpha_AF4, Percent_Beta_AF4,  
Percent_Gamma_AF4, AVG_A_X2];  
F3_Matrix = [Percent_Delta_F3, Percent_Theta_F3, Percent_Alpha_F3, Percent_Beta_F3, Percent_Gamma_F3,  
AVG_A_X3];  
F4_Matrix = [Percent_Delta_F4, Percent_Theta_F4, Percent_Alpha_F4, Percent_Beta_F4, Percent_Gamma_F4,  
AVG_A_X4];
```

```

F7_Matrix = [Percent_Delta_F7, Percent_Theta_F7, Percent_Alpha_F7, Percent_Beta_F7, Percent_Gamma_F7,
AVG_A_X5];

F8_Matrix = [Percent_Delta_F8, Percent_Theta_F8, Percent_Alpha_F8, Percent_Beta_F8, Percent_Gamma_F8,
AVG_A_X6];

FC5_Matrix = [Percent_Delta_FC5, Percent_Theta_FC5, Percent_Alpha_FC5, Percent_Beta_FC5,
Percent_Gamma_FC5, AVG_A_X7];

FC6_Matrix = [Percent_Delta_FC6, Percent_Theta_FC6, Percent_Alpha_FC6, Percent_Beta_FC6,
Percent_Gamma_FC6, AVG_A_X8];

O1_Matrix = [Percent_Delta_O1, Percent_Theta_O1, Percent_Alpha_O1, Percent_Beta_O1, Percent_Gamma_O1,
AVG_A_X9];

O2_Matrix = [Percent_Delta_O2, Percent_Theta_O2, Percent_Alpha_O2, Percent_Beta_O2, Percent_Gamma_O2,
AVG_A_X10];

P7_Matrix = [Percent_Delta_P7, Percent_Theta_P7, Percent_Alpha_P7, Percent_Beta_P7, Percent_Gamma_P7,
AVG_A_X11];

P8_Matrix = [Percent_Delta_P8, Percent_Theta_P8, Percent_Alpha_P8, Percent_Beta_P8, Percent_Gamma_P8,
AVG_A_X12];

T7_Matrix = [Percent_Delta_T7, Percent_Theta_T7, Percent_Alpha_T7, Percent_Beta_T7, Percent_Gamma_T7,
AVG_A_X13];

T8_Matrix = [Percent_Delta_T8, Percent_Theta_T8, Percent_Alpha_T8, Percent_Beta_T8, Percent_Gamma_T8,
AVG_A_X14];

filename = input('Please choose fileName to save results, leave empty to not save file: ', 's');

if isempty(filename) == 0;

    col_Header_1 = {'Percent_Delta_AF3', 'Percent_Theta_AF3', 'Percent_Alpha_AF3', 'Percent_Beta_AF3',
'Percent_Gamma_AF3', 'AVG_Amplitude_AF3'};

```

```

col_Header_2 = {'Percent_Delta_AF4', 'Percent_Theta_AF4', 'Percent_Alpha_AF4', 'Percent_Beta_AF4',
'Percent_Gamma_AF4', 'AVG_Amplitude_AF4'};

col_Header_3 = {'Percent_Delta_F3', 'Percent_Theta_F3', 'Percent_Alpha_F3', 'Percent_Beta_F3',
'Percent_Gamma_F3', 'AVG_Amplitude_F3'};

col_Header_4 = {'Percent_Delta_F4', 'Percent_Theta_F4', 'Percent_Alpha_F4', 'Percent_Beta_F4',
'Percent_Gamma_F4', 'AVG_Amplitude_F4'};

col_Header_5 = {'Percent_Delta_F7', 'Percent_Theta_F7', 'Percent_Alpha_F7', 'Percent_Beta_F7',
'Percent_Gamma_F7', 'AVG_Amplitude_F7'};

col_Header_6 = {'Percent_Delta_F8', 'Percent_Theta_F8', 'Percent_Alpha_F8', 'Percent_Beta_F8',
'Percent_Gamma_F8', 'AVG_Amplitude_F8'};

col_Header_7 = {'Percent_Delta_FC5', 'Percent_Theta_FC5', 'Percent_Alpha_FC5', 'Percent_Beta_FC5',
'Percent_Gamma_FC5', 'AVG_Amplitude_FC5'};

col_Header_8 = {'Percent_Delta_FC6', 'Percent_Theta_FC6', 'Percent_Alpha_FC6', 'Percent_Beta_FC6',
'Percent_Gamma_FC6', 'AVG_Amplitude_FC6'};

col_Header_9 = {'Percent_Delta_O1', 'Percent_Theta_O1', 'Percent_Alpha_O1', 'Percent_Beta_O1',
'Percent_Gamma_O1', 'AVG_Amplitude_O1'};

col_Header_10 = {'Percent_Delta_O2', 'Percent_Theta_O2', 'Percent_Alpha_O2', 'Percent_Beta_O2',
'Percent_Gamma_O2', 'AVG_Amplitude_O2'};

col_Header_11 = {'Percent_Delta_P7', 'Percent_Theta_P7', 'Percent_Alpha_P7', 'Percent_Beta_P7',
'Percent_Gamma_P7', 'AVG_Amplitude_P7'};

col_Header_12 = {'Percent_Delta_P8', 'Percent_Theta_P8', 'Percent_Alpha_P8', 'Percent_Beta_P8',
'Percent_Gamma_P8', 'AVG_Amplitude_P8'};

col_Header_13 = {'Percent_Delta_T7', 'Percent_Theta_T7', 'Percent_Alpha_T7', 'Percent_Beta_T7',
'Percent_Gamma_T7', 'AVG_Amplitude_T7'};

col_Header_14 = {'Percent_Delta_T8', 'Percent_Theta_T8', 'Percent_Alpha_T8', 'Percent_Beta_T8',
'Percent_Gamma_T8', 'AVG_Amplitude_T8'};

row = [1:length(Percent_Delta_T8),1];

```

```
row = row';
```

```
row_Header_1 = (row);
```

```
%AF3
```

```
xlswrite(filename, AF3_Matrix, 'Sheet1', 'B2');
```

```
xlswrite(filename, col_Header_1, 'Sheet1', 'B1');
```

```
xlswrite(filename, row_Header_1, 'Sheet1', 'A2');
```

```
xlswrite(filename, {'Blood Glucose Level'}, 'Sheet1', 'I1');
```

```
xlswrite(filename, AVG_AF3_BGL, 'Sheet1', 'I2');
```

```
%AF4
```

```
xlswrite(filename, AF4_Matrix, 'Sheet2', 'B2');
```

```
xlswrite(filename, col_Header_2, 'Sheet2', 'B1');
```

```
xlswrite(filename, row_Header_1, 'Sheet2', 'A2');
```

```
xlswrite(filename, {'Blood Glucose Level'}, 'Sheet1', 'I1');
```

```
xlswrite(filename, AVG_AF3_BGL, 'Sheet2', 'I2');
```

```
%F3
```

```
xlswrite(filename, F3_Matrix, 'Sheet3', 'B2');
```

```
xlswrite(filename, col_Header_3, 'Sheet3', 'B1');
```

```
xlswrite(filename, row_Header_1, 'Sheet3', 'A2');
```

```
xlswrite(filename, {'Blood Glucose Level'}, 'Sheet3', 'I1');
```

```
xlswrite(filename, AVG_AF3_BGL, 'Sheet3', 'I2');
```

```
%F4
```

```
xlswrite(filename, F4_Matrix, 'Sheet4', 'B2');
```

```
xlswrite(filename, col_Header_4, 'Sheet4', 'B1');
```

```
xlswrite(filename, row_Header_1, 'Sheet4', 'A2');
```

```
xlswrite(filename, {'Blood Glucose Level'}, 'Sheet4', 'I1');
```

```
xlswrite(filename, AVG_AF3_BGL, 'Sheet4', 'I2');
```

```
%F7
```

```

xlswrite(filename, F7_Matrix, 'Sheet5', 'B2');
xlswrite(filename, col_Header_5, 'Sheet5', 'B1');
xlswrite(filename, row_Header_1, 'Sheet5', 'A2');
xlswrite(filename, {'Blood Glucose Level'}, 'Sheet5', 'I1');
xlswrite(filename, AVG_AF3_BGL, 'Sheet5', 'I2');
%F8
xlswrite(filename, F8_Matrix, 'Sheet6', 'B2');
xlswrite(filename, col_Header_6, 'Sheet6', 'B1');
xlswrite(filename, row_Header_1, 'Sheet6', 'A2');
xlswrite(filename, {'Blood Glucose Level'}, 'Sheet6', 'I1');
xlswrite(filename, AVG_AF3_BGL, 'Sheet6', 'I2');
%FC5
xlswrite(filename, FC5_Matrix, 'Sheet7', 'B2');
xlswrite(filename, col_Header_7, 'Sheet7', 'B1');
xlswrite(filename, row_Header_1, 'Sheet7', 'A2');
xlswrite(filename, {'Blood Glucose Level'}, 'Sheet7', 'I1');
xlswrite(filename, AVG_AF3_BGL, 'Sheet7', 'I2');
%FC6
xlswrite(filename, FC6_Matrix, 'Sheet8', 'B2');
xlswrite(filename, col_Header_8, 'Sheet8', 'B1');
xlswrite(filename, row_Header_1, 'Sheet8', 'A2');
xlswrite(filename, {'Blood Glucose Level'}, 'Sheet8', 'I1');
xlswrite(filename, AVG_AF3_BGL, 'Sheet8', 'I2');
%O1
xlswrite(filename, O1_Matrix, 'Sheet9', 'B2');
xlswrite(filename, col_Header_9, 'Sheet9', 'B1');
xlswrite(filename, row_Header_1, 'Sheet9', 'A2');
xlswrite(filename, {'Blood Glucose Level'}, 'Sheet9', 'I1');

```

```

xlswrite(filename, AVG_AF3_BGL, 'Sheet9', 'I2');

%O2

xlswrite(filename, O2_Matrix, 'Sheet10', 'B2');

xlswrite(filename, col_Header_10, 'Sheet10', 'B1');

xlswrite(filename, row_Header_1, 'Sheet10', 'A2');

xlswrite(filename, {'Blood Glucose Level'}, 'Sheet10', 'I1');

xlswrite(filename, AVG_AF3_BGL, 'Sheet10', 'I2');

%P7

xlswrite(filename, P7_Matrix, 'Sheet11', 'B2');

xlswrite(filename, col_Header_11, 'Sheet11', 'B1');

xlswrite(filename, row_Header_1, 'Sheet11', 'A2');

xlswrite(filename, {'Blood Glucose Level'}, 'Sheet11', 'I1');

xlswrite(filename, AVG_AF3_BGL, 'Sheet11', 'I2');

%P8

xlswrite(filename, P8_Matrix, 'Sheet12', 'B2');

xlswrite(filename, col_Header_12, 'Sheet12', 'B1');

xlswrite(filename, row_Header_1, 'Sheet12', 'A2');

xlswrite(filename, {'Blood Glucose Level'}, 'Sheet12', 'I1');

xlswrite(filename, AVG_AF3_BGL, 'Sheet12', 'I2');

%T7

xlswrite(filename, T7_Matrix, 'Sheet13', 'B2');

xlswrite(filename, col_Header_13, 'Sheet13', 'B1');

xlswrite(filename, row_Header_1, 'Sheet13', 'A2');

xlswrite(filename, {'Blood Glucose Level'}, 'Sheet13', 'I1');

xlswrite(filename, AVG_AF3_BGL, 'Sheet13', 'I2');

%T8

xlswrite(filename, T8_Matrix, 'Sheet14', 'B2');

xlswrite(filename, col_Header_14, 'Sheet14', 'B1');

```

```

\
    xlswrite(filename, row_Header_1, 'Sheet14', 'A2');
    xlswrite(filename, {'Blood Glucose Level'}, 'Sheet14', 'I1');
    xlswrite(filename, AVG_AF3_BGL, 'Sheet14', 'I2');

elseif isempty(filename) == 1;
end

%%
%Setting up inputs
inputs_Delta_AF3 = Percent_Delta_AF3;
inputs_Delta_AF4 = Percent_Delta_AF4;
inputs_Delta_F3 = Percent_Delta_F3;
inputs_Delta_F4 = Percent_Delta_F4;
inputs_Delta_F7 = Percent_Delta_F7;
inputs_Delta_F8 = Percent_Delta_F8;
inputs_Delta_FC5 = Percent_Delta_FC5;
inputs_Delta_FC6 = Percent_Delta_FC6;
inputs_Delta_O1 = Percent_Delta_O1;
inputs_Delta_O2 = Percent_Delta_O2;
inputs_Delta_P7 = Percent_Delta_P7;
inputs_Delta_P8 = Percent_Delta_P8;
inputs_Delta_T7 = Percent_Delta_T7;
inputs_Delta_T8 = Percent_Delta_T8;
inputs_Theta_AF3 = Percent_Theta_AF3;
inputs_Theta_AF4 = Percent_Theta_AF4;
inputs_Theta_F3 = Percent_Theta_F3;
inputs_Theta_F4 = Percent_Theta_F4;
inputs_Theta_F7 = Percent_Theta_F7;

```

inputs_Theta_F8 = Percent_Theta_F8;
inputs_Theta_FC5 = Percent_Theta_FC5;
inputs_Theta_FC6 = Percent_Theta_FC6;
inputs_Theta_O1 = Percent_Theta_O1;
inputs_Theta_O2= Percent_Theta_O2;
inputs_Theta_P7 = Percent_Theta_P7;
inputs_Theta_P8 = Percent_Theta_P8;
inputs_Theta_T7 = Percent_Theta_T7;
inputs_Theta_T8 = Percent_Theta_T8;
inputs_Alpha_AF3 = Percent_Alpha_AF3;
inputs_Alpha_AF4 = Percent_Alpha_AF4;
inputs_Alpha_F3 = Percent_Alpha_F3;
inputs_Alpha_F4 = Percent_Alpha_F4;
inputs_Alpha_F7 = Percent_Alpha_F7;
inputs_Alpha_F8 = Percent_Alpha_F8;
inputs_Alpha_FC5 = Percent_Alpha_FC5;
inputs_Alpha_FC6 = Percent_Alpha_FC6;
inputs_Alpha_O1 = Percent_Alpha_O1;
inputs_Alpha_O2 = Percent_Alpha_O2;
inputs_Alpha_P7 = Percent_Alpha_P7;
inputs_Alpha_P8 = Percent_Alpha_P8;
inputs_Alpha_T7 = Percent_Alpha_T7;
inputs_Alpha_T8 = Percent_Alpha_T8;
inputs_Beta_AF3 = Percent_Beta_AF3;
inputs_Beta_AF4 = Percent_Beta_AF4;
inputs_Beta_F3 = Percent_Beta_F3;
inputs_Beta_F4 = Percent_Beta_F4;
inputs_Beta_F7 = Percent_Beta_F7;

```

inputs_Beta_F8 = Percent_Beta_F8;
inputs_Beta_FC5 = Percent_Beta_FC5;
inputs_Beta_FC6 = Percent_Beta_FC6;
inputs_Beta_O1 = Percent_Beta_O1;
inputs_Beta_O2 = Percent_Beta_O2;
inputs_Beta_P7 = Percent_Beta_P7;
inputs_Beta_P8 = Percent_Beta_P8;
inputs_Beta_T7 = Percent_Beta_T7;
inputs_Beta_T8 = Percent_Beta_T8;
inputs_Gamma_AF3 = Percent_Gamma_AF3;
inputs_Gamma_AF4 = Percent_Gamma_AF4;
inputs_Gamma_F3 = Percent_Gamma_F3;
inputs_Gamma_F4 = Percent_Gamma_F4;
inputs_Gamma_F7 = Percent_Gamma_F7;
inputs_Gamma_F8 = Percent_Gamma_F8;
inputs_Gamma_FC5 = Percent_Gamma_FC5;
inputs_Gamma_FC6 = Percent_Gamma_FC6;
inputs_Gamma_O1 = Percent_Gamma_O1;
inputs_Gamma_O2 = Percent_Gamma_O2;
inputs_Gamma_P7 = Percent_Gamma_P7;
inputs_Gamma_P8 = Percent_Gamma_P8;
inputs_Gamma_T7 = Percent_Gamma_T7;
inputs_Gamma_T8 = Percent_Gamma_T8;

```

% All Inputs

```

input_matrix = [inputs_Delta_AF4, inputs_Theta_AF4, inputs_Alpha_AF4, inputs_Beta_AF4,
inputs_Gamma_AF4, inputs_Delta_F4, inputs_Theta_F4, inputs_Alpha_F4, inputs_Beta_F4, inputs_Gamma_F4,
inputs_Delta_F8, inputs_Theta_F8, inputs_Alpha_F8, inputs_Beta_F8, inputs_Gamma_F8, inputs_Delta_FC5,

```



```

\

%

% This script MUST be run after BGL_Code_For_R_Coefficient_Preparation

%%

close all; clc;

%%

%File Name to Save Results

FileName = input('Please choose file name to results: ', 's');

%%

%Setting up inputs

inputs_Delta_AF3 = AVG_Delta_AF3';
inputs_Delta_AF4 = AVG_Delta_AF4';
inputs_Delta_F3 = AVG_Delta_F3';
inputs_Delta_F4 = AVG_Delta_F4';
inputs_Delta_F7 = AVG_Delta_F7';
inputs_Delta_F8 = AVG_Delta_F8';
inputs_Delta_FC5 = AVG_Delta_FC5';
inputs_Delta_FC6 = AVG_Delta_FC6';
inputs_Delta_O1 = AVG_Delta_O1';
inputs_Delta_O2 = AVG_Delta_O2';
inputs_Delta_P7 = AVG_Delta_P7';
inputs_Delta_P8 = AVG_Delta_P8';
inputs_Delta_T7 = AVG_Delta_T7';
inputs_Delta_T8 = AVG_Delta_T8';
inputs_Theta_AF3 = AVG_Theta_AF3';
inputs_Theta_AF4 = AVG_Theta_AF4';
inputs_Theta_F3 = AVG_Theta_F3';
inputs_Theta_F4 = AVG_Theta_F4';

```

inputs_Theta_F7 = AVG_Theta_F7';
inputs_Theta_F8 = AVG_Theta_F8';
inputs_Theta_FC5 = AVG_Theta_FC5';
inputs_Theta_FC6 = AVG_Theta_FC6';
inputs_Theta_O1 = AVG_Theta_O1';
inputs_Theta_O2 = AVG_Theta_O2';
inputs_Theta_P7 = AVG_Theta_P7';
inputs_Theta_P8 = AVG_Theta_P8';
inputs_Theta_T7 = AVG_Theta_T7';
inputs_Theta_T8 = AVG_Theta_T8';
inputs_Alpha_AF3 = AVG_Alpha_AF3';
inputs_Alpha_AF4 = AVG_Alpha_AF4';
inputs_Alpha_F3 = AVG_Alpha_F3';
inputs_Alpha_F4 = AVG_Alpha_F4';
inputs_Alpha_F7 = AVG_Alpha_F7';
inputs_Alpha_F8 = AVG_Alpha_F8';
inputs_Alpha_FC5 = AVG_Alpha_FC5';
inputs_Alpha_FC6 = AVG_Alpha_FC6';
inputs_Alpha_O1 = AVG_Alpha_O1';
inputs_Alpha_O2 = AVG_Alpha_O2';
inputs_Alpha_P7 = AVG_Alpha_P7';
inputs_Alpha_P8 = AVG_Alpha_P8';
inputs_Alpha_T7 = AVG_Alpha_T7';
inputs_Alpha_T8 = AVG_Alpha_T8';
inputs_Beta_AF3 = AVG_Beta_AF3';
inputs_Beta_AF4 = AVG_Beta_AF4';
inputs_Beta_F3 = AVG_Beta_F3';
inputs_Beta_F4 = AVG_Beta_F4';

```
\n\ninputs_Beta_F7 = AVG_Beta_F7';\ninputs_Beta_F8 = AVG_Beta_F8';\ninputs_Beta_FC5 = AVG_Beta_FC5';\ninputs_Beta_FC6 = AVG_Beta_FC6';\ninputs_Beta_O1 = AVG_Beta_O1';\ninputs_Beta_O2 = AVG_Beta_O2';\ninputs_Beta_P7 = AVG_Beta_P7';\ninputs_Beta_P8 = AVG_Beta_P8';\ninputs_Beta_T7 = AVG_Beta_T7';\ninputs_Beta_T8 = AVG_Beta_T8';\n\ninputs_Gamma_AF3 = AVG_Gamma_AF3';\ninputs_Gamma_AF4 = AVG_Gamma_AF4';\ninputs_Gamma_F3 = AVG_Gamma_F3';\ninputs_Gamma_F4 = AVG_Gamma_F4';\ninputs_Gamma_F7 = AVG_Gamma_F7';\ninputs_Gamma_F8 = AVG_Gamma_F8';\ninputs_Gamma_FC5 = AVG_Gamma_FC5';\ninputs_Gamma_FC6 = AVG_Gamma_FC6';\ninputs_Gamma_O1 = AVG_Gamma_O1';\ninputs_Gamma_O2 = AVG_Gamma_O2';\ninputs_Gamma_P7 = AVG_Gamma_P7';\ninputs_Gamma_P8 = AVG_Gamma_P8';\ninputs_Gamma_T7 = AVG_Gamma_T7';\ninputs_Gamma_T8 = AVG_Gamma_T8';
```

```
%Setting up targets
```

```
targets = AVG_AF4_BGL';
```

```

\

% Create a Fitting Network

hiddenLayerSize = 30;

net = fitnet(hiddenLayerSize);

% Set up Division of Data for Training, Validation, Testing

net.divideParam.trainRatio = 75/100;

net.divideParam.valRatio = 5/100;

net.divideParam.testRatio = 20/100;

%%

% Training Neural Networks for All Delta

%Delta

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_AF3,targets);

% Test the Network

outputs = net(inputs_Delta_AF3);

errors = gsubtract(outputs,targets);

performance_Delta_AF3(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

```

```

\

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Delta_AF3(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_AF4,targets);

% Test the Network

outputs = net(inputs_Delta_AF4);

errors = gsubtract(outputs,targets);

performance_Delta_AF4(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

```

```

close figure 1

% figure, ploterrhist(errors)

r_Delta_AF4(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_F3,targets);

% Test the Network

outputs = net(inputs_Delta_F3);

errors = gsubtract(outputs,targets);

performance_Delta_F3(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

```

```

r_Delta_F3(n) = getappdata(0, 'r');
end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Delta_F4,targets);

% Test the Network
outputs = net(inputs_Delta_F4);
errors = gsubtract(outputs,targets);
performance_Delta_F4(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

r_Delta_F4(n) = getappdata(0, 'r');
end

```

```

\

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_F7,targets);

% Test the Network

outputs = net(inputs_Delta_F7);

errors = gsubtract(outputs,targets);

performance_Delta_F7(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Delta_F7(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

```

```

\

[net,tr] = trainbr(net,inputs_Delta_F8,targets);

% Test the Network

outputs = net(inputs_Delta_F8);
errors = gsubtract(outputs,targets);
performance_Delta_F8(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1

% figure, ploterrhist(errors)

r_Delta_F8(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_FC5,targets);

% Test the Network

```

```

\

outputs = net(inputs_Delta_FC5);
errors = gsubtract(outputs,targets);
performance_Delta_FC5(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

r_Delta_FC5(n) = getappdata(0, 'r');

end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Delta_FC6,targets);

% Test the Network
outputs = net(inputs_Delta_FC6);
errors = gsubtract(outputs,targets);
performance_Delta_FC6(n) = perform(net,targets,outputs);

```

```

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Delta_FC6(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_O1,targets);

% Test the Network

outputs = net(inputs_Delta_O1);

errors = gsubtract(outputs,targets);

performance_Delta_O1(n) = perform(net,targets,outputs);

% View the Network

% view(net)

```

```

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Delta_O1(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_O2,targets);

% Test the Network

outputs = net(inputs_Delta_O2);

errors = gsubtract(outputs,targets);

performance_Delta_O2(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

```

```

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Delta_O2(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_P7,targets);

% Test the Network

outputs = net(inputs_Delta_P7);

errors = gsubtract(outputs,targets);

performance_Delta_P7(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

```

```

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Delta_P7(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_P8,targets);

% Test the Network

outputs = net(inputs_Delta_P8);

errors = gsubtract(outputs,targets);

performance_Delta_P8(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

```

```

r_Delta_P8(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_T7,targets);

% Test the Network

outputs = net(inputs_Delta_T7);

errors = gsubtract(outputs,targets);

performance_Delta_T7(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Delta_T7(n) = getappdata(0, 'r');

end

```

```

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_T8,targets);

% Test the Network

outputs = net(inputs_Delta_T8);

errors = gsubtract(outputs,targets);

performance_Delta_T8(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Delta_T8(n) = getappdata(0, 'r');

end

%%

% Training Neural Networks for Theta

```

```

\

%Theta

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Theta_AF3,targets);

% Test the Network

outputs = net(inputs_Theta_AF3);

errors = gsubtract(outputs,targets);

performance_D_Theta_AF3(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Theta_AF3(n) = getappdata(0, 'r');

end

for n = 1:50

```

```

\

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_AF4,targets);

% Test the Network

outputs = net(inputs_Delta_AF4);

errors = gsubtract(outputs,targets);

performance_Theta_AF4(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Theta_AF4(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Theta_F3,targets);

```

```

\

% Test the Network

outputs = net(inputs_Theta_F3);

errors = gsubtract(outputs,targets);

performance_Theta_F3(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Theta_F3(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Theta_F4,targets);

% Test the Network

outputs = net(inputs_Theta_F4);

```

```

errors = gsubtract(outputs,targets);

performance_Theta_F4(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

r_Theta_F4(n) = getappdata(0, 'r');

end

for n = 1:50
% Train the Network
net = init(net);

[net,tr] = trainbr(net,inputs_Theta_F7,targets);

% Test the Network
outputs = net(inputs_Theta_F7);
errors = gsubtract(outputs,targets);
performance_Theta_F7(n) = perform(net,targets,outputs);

```

```

\

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Theta_F7(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Theta_F8,targets);

% Test the Network

outputs = net(inputs_Theta_F8);

errors = gsubtract(outputs,targets);

performance_Theta_F8(n) = perform(net,targets,outputs);

% View the Network

% view(net)

```

```

\

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Theta_F8(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Theta_FC5,targets);

% Test the Network

outputs = net(inputs_Theta_FC5);

errors = gsubtract(outputs,targets);

performance_Theta_FC5(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

```

```

\

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Theta_FC5(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Theta_FC6,targets);

% Test the Network

outputs = net(inputs_Theta_FC6);

errors = gsubtract(outputs,targets);

performance_Theta_FC6(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

```

```

close figure 1

% figure, ploterrhist(errors)

r_Theta_FC6(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Theta_O1,targets);

% Test the Network

outputs = net(inputs_Theta_O1);

errors = gsubtract(outputs,targets);

performance_Theta_O1(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

```

```

r_Theta_O1(n) = getappdata(0, 'r');
end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Theta_O2,targets);

% Test the Network
outputs = net(inputs_Theta_O2);
errors = gsubtract(outputs,targets);
performance_Theta_O2(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

r_Theta_O2(n) = getappdata(0, 'r');
end

```

```

\

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Theta_P7,targets);

% Test the Network

outputs = net(inputs_Theta_P7);

errors = gsubtract(outputs,targets);

performance_Theta_P7(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Theta_P7(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

```

```

\

[net,tr] = trainbr(net,inputs_Theta_P8,targets);

% Test the Network

outputs = net(inputs_Theta_P8);
errors = gsubtract(outputs,targets);
performance_Theta_P8(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Theta_P8(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Theta_T7,targets);

% Test the Network

```

```

\

outputs = net(inputs_Theta_T7);
errors = gsubtract(outputs,targets);
performance_Theta_T7(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

r_Theta_T7(n) = getappdata(0, 'r');

end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Theta_T8,targets);

% Test the Network
outputs = net(inputs_Theta_T8);
errors = gsubtract(outputs,targets);
performance_Theta_T8(n) = perform(net,targets,outputs);

```

```

\

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Theta_T8(n) = getappdata(0, 'r');

end

%%

% Training Neural Networks for Alpha

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Alpha_AF3,targets);

% Test the Network

outputs = net(inputs_Alpha_AF3);

errors = gsubtract(outputs,targets);

performance_D_Alpha_AF3(n) = perform(net,targets,outputs);

```

```

\

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Alpha_AF3(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_AF4,targets);

% Test the Network

outputs = net(inputs_Delta_AF4);

errors = gsubtract(outputs,targets);

performance_Alpha_AF4(n) = perform(net,targets,outputs);

% View the Network

% view(net)

```

```

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Alpha_AF4(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Alpha_F3,targets);

% Test the Network

outputs = net(inputs_Alpha_F3);

errors = gsubtract(outputs,targets);

performance_Alpha_F3(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

```

```

\

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Alpha_F3(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Alpha_F4,targets);

% Test the Network

outputs = net(inputs_Alpha_F4);

errors = gsubtract(outputs,targets);

performance_Alpha_F4(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

```

```

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Alpha_F4(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Alpha_F7,targets);

% Test the Network

outputs = net(inputs_Alpha_F7);

errors = gsubtract(outputs,targets);

performance_Alpha_F7(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

```

```

r_Alpha_F7(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Alpha_F8,targets);

% Test the Network

outputs = net(inputs_Alpha_F8);

errors = gsubtract(outputs,targets);

performance_Alpha_F8(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Alpha_F8(n) = getappdata(0, 'r');

end

```

```

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Alpha_FC5,targets);

% Test the Network

outputs = net(inputs_Alpha_FC5);

errors = gsubtract(outputs,targets);

performance_Alpha_FC5(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Alpha_FC5(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

```

```

net = init(net);

[net,tr] = trainbr(net,inputs_Alpha_FC6,targets);

% Test the Network

outputs = net(inputs_Alpha_FC6);
errors = gsubtract(outputs,targets);
performance_Alpha_FC6(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Alpha_FC6(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Alpha_O1,targets);

```

```

\

% Test the Network

outputs = net(inputs_Alpha_O1);

errors = gsubtract(outputs,targets);

performance_Alpha_O1(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Alpha_O1(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Alpha_O2,targets);

% Test the Network

outputs = net(inputs_Alpha_O2);

errors = gsubtract(outputs,targets);

```

```
\nperformance_Alpha_O2(n) = perform(net,targets,outputs);
```

```
% View the Network
```

```
% view(net)
```

```
% Plots
```

```
% Uncomment these lines to enable various plots.
```

```
% figure, plotperform(tr)
```

```
% figure, plottrainstate(tr)
```

```
% figure, plotfit(targets,outputs)
```

```
figure, plotregression_1(targets,outputs);
```

```
close figure 1
```

```
% figure, ploterrhist(errors)
```

```
r_Alpha_O2(n) = getappdata(0, 'r');
```

```
end
```

```
for n = 1:50
```

```
% Train the Network
```

```
net = init(net);
```

```
[net,tr] = trainbr(net,inputs_Alpha_P7,targets);
```

```
% Test the Network
```

```
outputs = net(inputs_Alpha_P7);
```

```
errors = gsubtract(outputs,targets);
```

```
performance_Alpha_P7(n) = perform(net,targets,outputs);
```

```
% View the Network
```

```

\

% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

r_Alpha_P7(n) = getappdata(0, 'r');
end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Alpha_P8,targets);

% Test the Network
outputs = net(inputs_Alpha_P8);
errors = gsubtract(outputs,targets);
performance_Alpha_P8(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots

```

```

\

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Alpha_P8(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Alpha_T7,targets);

% Test the Network

outputs = net(inputs_Alpha_T7);

errors = gsubtract(outputs,targets);

performance_Alpha_T7(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

```

```

\

% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1

% figure, ploterrhist(errors)

r_Alpha_T7(n) = getappdata(0, 'r');
end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Alpha_T8,targets);

% Test the Network
outputs = net(inputs_Alpha_T8);
errors = gsubtract(outputs,targets);
performance_Alpha_T8(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1

```

```

\

% figure, ploterrhist(errors)

r_Alpha_T8(n) = getappdata(0, 'r');

end

%%

% Training Neural Networks for Beta

%Beta
for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Beta_AF3,targets);

% Test the Network
outputs = net(inputs_Beta_AF3);
errors = gsubtract(outputs,targets);
performance_D_Beta_AF3(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);

```

```

close figure 1

% figure, ploterrhist(errors)

r_Beta_AF3(n) = getappdata(0, 'r');
end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Delta_AF4,targets);

% Test the Network
outputs = net(inputs_Delta_AF4);
errors = gsubtract(outputs,targets);
performance_Beta_AF4(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

```

```

r_Beta_AF4(n) = getappdata(0, 'r');
end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Beta_F3,targets);

% Test the Network
outputs = net(inputs_Beta_F3);
errors = gsubtract(outputs,targets);
performance_Beta_F3(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

r_Beta_F3(n) = getappdata(0, 'r');
end

```

```

\

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Beta_F4,targets);

% Test the Network

outputs = net(inputs_Beta_F4);

errors = gsubtract(outputs,targets);

performance_Beta_F4(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Beta_F4(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

```

```

[net,tr] = trainbr(net,inputs_Beta_F7,targets);

% Test the Network

outputs = net(inputs_Beta_F7);
errors = gsubtract(outputs,targets);
performance_Beta_F7(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1

% figure, ploterrhist(errors)

r_Beta_F7(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Beta_F8,targets);

% Test the Network

```

```

\

outputs = net(inputs_Beta_F8);
errors = gsubtract(outputs,targets);
performance_Beta_F8(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

r_Beta_F8(n) = getappdata(0, 'r');
end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Beta_FC5,targets);

% Test the Network
outputs = net(inputs_Beta_FC5);
errors = gsubtract(outputs,targets);
performance_Beta_FC5(n) = perform(net,targets,outputs);

```

```

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Beta_FC5(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Beta_FC6,targets);

% Test the Network

outputs = net(inputs_Beta_FC6);

errors = gsubtract(outputs,targets);

performance_Beta_FC6(n) = perform(net,targets,outputs);

% View the Network

% view(net)

```

```

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Beta_FC6(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Beta_O1,targets);

% Test the Network

outputs = net(inputs_Beta_O1);

errors = gsubtract(outputs,targets);

performance_Beta_O1(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

```

```

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Beta_O1(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Beta_O2,targets);

% Test the Network

outputs = net(inputs_Beta_O2);

errors = gsubtract(outputs,targets);

performance_Beta_O2(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

```

```

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Beta_O2(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Beta_P7,targets);

% Test the Network

outputs = net(inputs_Beta_P7);

errors = gsubtract(outputs,targets);

performance_Beta_P7(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

```

```

r_Beta_P7(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Beta_P8,targets);

% Test the Network

outputs = net(inputs_Beta_P8);

errors = gsubtract(outputs,targets);

performance_Beta_P8(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Beta_P8(n) = getappdata(0, 'r');

end

```

```

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Beta_T7,targets);

% Test the Network

outputs = net(inputs_Beta_T7);

errors = gsubtract(outputs,targets);

performance_Beta_T7(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Beta_T7(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

```

```

net = init(net);

[net,tr] = trainbr(net,inputs_Beta_T8,targets);

% Test the Network

outputs = net(inputs_Beta_T8);
errors = gsubtract(outputs,targets);
performance_Beta_T8(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1

% figure, ploterrhist(errors)

r_Beta_T8(n) = getappdata(0, 'r');

end

%%

% Training Neural Networks for Gamma

%Gamma

for n = 1:50

```

```

\

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Gamma_AF3,targets);

% Test the Network

outputs = net(inputs_Gamma_AF3);

errors = gsubtract(outputs,targets);

performance_D_Gamma_AF3(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Gamma_AF3(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Delta_AF4,targets);

```

```

\

% Test the Network

outputs = net(inputs_Delta_AF4);

errors = gsubtract(outputs,targets);

performance_Gamma_AF4(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Gamma_AF4(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Gamma_F3,targets);

% Test the Network

outputs = net(inputs_Gamma_F3);

```

```

errors = gsubtract(outputs,targets);

performance_Gamma_F3(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Gamma_F3(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Gamma_F4,targets);

% Test the Network

outputs = net(inputs_Gamma_F4);

errors = gsubtract(outputs,targets);

performance_Gamma_F4(n) = perform(net,targets,outputs);

```

```

\

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Gamma_F4(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Gamma_F7,targets);

% Test the Network

outputs = net(inputs_Gamma_F7);

errors = gsubtract(outputs,targets);

performance_Gamma_F7(n) = perform(net,targets,outputs);

% View the Network

% view(net)

```

```

\

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Gamma_F7(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Gamma_F8,targets);

% Test the Network

outputs = net(inputs_Gamma_F8);

errors = gsubtract(outputs,targets);

performance_Gamma_F8(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

```

```

\

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Gamma_F8(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Gamma_FC5,targets);

% Test the Network

outputs = net(inputs_Gamma_FC5);

errors = gsubtract(outputs,targets);

performance_Gamma_FC5(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

```

```

close figure 1

% figure, ploterrhist(errors)

r_Gamma_FC5(n) = getappdata(0, 'r');
end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Gamma_FC6,targets);

% Test the Network
outputs = net(inputs_Gamma_FC6);
errors = gsubtract(outputs,targets);
performance_Gamma_FC6(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

```

```

r_Gamma_FC6(n) = getappdata(0, 'r');
end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Gamma_O1,targets);

% Test the Network
outputs = net(inputs_Gamma_O1);
errors = gsubtract(outputs,targets);
performance_Gamma_O1(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

r_Gamma_O1(n) = getappdata(0, 'r');
end

```

```

\

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Gamma_O2,targets);

% Test the Network

outputs = net(inputs_Gamma_O2);

errors = gsubtract(outputs,targets);

performance_Gamma_O2(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Gamma_O2(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

```

```

[net,tr] = trainbr(net,inputs_Gamma_P7,targets);

% Test the Network

outputs = net(inputs_Gamma_P7);
errors = gsubtract(outputs,targets);
performance_Gamma_P7(n) = perform(net,targets,outputs);

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Gamma_P7(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Gamma_P8,targets);

% Test the Network

```

```

\

outputs = net(inputs_Gamma_P8);
errors = gsubtract(outputs,targets);
performance_Gamma_P8(n) = perform(net,targets,outputs);

% View the Network
% view(net)

% Plots
% Uncomment these lines to enable various plots.
% figure, plotperform(tr)
% figure, plottrainstate(tr)
% figure, plotfit(targets,outputs)
figure, plotregression_1(targets,outputs);
close figure 1
% figure, ploterrhist(errors)

r_Gamma_P8(n) = getappdata(0, 'r');
end

for n = 1:50
% Train the Network
net = init(net);
[net,tr] = trainbr(net,inputs_Gamma_T7,targets);

% Test the Network
outputs = net(inputs_Gamma_T7);
errors = gsubtract(outputs,targets);
performance_Gamma_T7(n) = perform(net,targets,outputs);

```

```

% View the Network

% view(net)

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Gamma_T7(n) = getappdata(0, 'r');

end

for n = 1:50

% Train the Network

net = init(net);

[net,tr] = trainbr(net,inputs_Gamma_T8,targets);

% Test the Network

outputs = net(inputs_Gamma_T8);

errors = gsubtract(outputs,targets);

performance_Gamma_T8(n) = perform(net,targets,outputs);

% View the Network

% view(net)

```

```

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets,outputs);

close figure 1

% figure, ploterrhist(errors)

r_Gamma_T8(n) = getappdata(0, 'r');

end

%%

%Placing Results within Matrix

r_Matrix = [r_Delta_AF3', r_Delta_AF4', r_Delta_F3', r_Delta_F4', r_Delta_F7', r_Delta_F8', r_Delta_FC5',
r_Delta_FC6', r_Delta_O1', r_Delta_O2', r_Delta_P7', r_Delta_P8', r_Delta_T7', r_Delta_T8', r_Theta_AF3',
r_Theta_AF4', r_Theta_F3', r_Theta_F4', r_Theta_F7', r_Theta_F8', r_Theta_FC5', r_Theta_FC6', r_Theta_O1',
r_Theta_O2', r_Theta_P7', r_Theta_P8', r_Theta_T7', r_Theta_T8', r_Alpha_AF3', r_Alpha_AF4', r_Alpha_F3',
r_Alpha_F4', r_Alpha_F7', r_Alpha_F8', r_Alpha_FC5', r_Alpha_FC6', r_Alpha_O1', r_Alpha_O2', r_Alpha_P7',
r_Alpha_P8', r_Alpha_T7', r_Alpha_T8', r_Beta_AF3', r_Beta_AF4', r_Beta_F3', r_Beta_F4', r_Beta_F7',
r_Beta_F8', r_Beta_FC5', r_Beta_FC6', r_Beta_O1', r_Beta_O2', r_Beta_P7', r_Beta_P8', r_Beta_T7', r_Beta_T8',
r_Gamma_AF3', r_Gamma_AF4', r_Gamma_F3', r_Gamma_F4', r_Gamma_F7', r_Gamma_F8', r_Gamma_FC5',
r_Gamma_FC6', r_Gamma_O1', r_Gamma_O2', r_Gamma_P7', r_Gamma_P8', r_Gamma_T7', r_Gamma_T8'];

%Writing Data to Excel File

xlswrite(FileName, r_Matrix);

```

BGL Code For R Number Of Hidden Neuron Determination.m

```
%Neural Network Modified Example Code

% Solve an Input-Output Fitting problem with a Neural Network
% Bryce A. Cranwell
%
% This script MUST be run after BGL_Code_For_R_Coefficient_Preparation

%%
close all; clc;
%%
%File Name to Save Results
% FileName = input('Please choose file name to results: ', 's');
%%
%Setting up inputs
inputs_Delta_AF3 = Percent_Delta_AF3;
inputs_Delta_AF4 = Percent_Delta_AF4;
inputs_Delta_F3 = Percent_Delta_F3;
inputs_Delta_F4 = Percent_Delta_F4;
inputs_Delta_F7 = Percent_Delta_F7;
inputs_Delta_F8 = Percent_Delta_F8;
inputs_Delta_FC5 = Percent_Delta_FC5;
inputs_Delta_FC6 = Percent_Delta_FC6;
inputs_Delta_O1 = Percent_Delta_O1;
inputs_Delta_O2 = Percent_Delta_O2;
inputs_Delta_P7 = Percent_Delta_P7;
inputs_Delta_P8 = Percent_Delta_P8;
inputs_Delta_T7 = Percent_Delta_T7;
```

inputs_Delta_T8 = Percent_Delta_T8;
inputs_Theta_AF3 = Percent_Theta_AF3;
inputs_Theta_AF4 = Percent_Theta_AF4;
inputs_Theta_F3 = Percent_Theta_F3;
inputs_Theta_F4 = Percent_Theta_F4;
inputs_Theta_F7 = Percent_Theta_F7;
inputs_Theta_F8 = Percent_Theta_F8;
inputs_Theta_FC5 = Percent_Theta_FC5;
inputs_Theta_FC6 = Percent_Theta_FC6;
inputs_Theta_O1 = Percent_Theta_O1;
inputs_Theta_O2 = Percent_Theta_O2;
inputs_Theta_P7 = Percent_Theta_P7;
inputs_Theta_P8 = Percent_Theta_P8;
inputs_Theta_T7 = Percent_Theta_T7;
inputs_Theta_T8 = Percent_Theta_T8;
inputs_Alpha_AF3 = Percent_Alpha_AF3;
inputs_Alpha_AF4 = Percent_Alpha_AF4;
inputs_Alpha_F3 = Percent_Alpha_F3;
inputs_Alpha_F4 = Percent_Alpha_F4;
inputs_Alpha_F7 = Percent_Alpha_F7;
inputs_Alpha_F8 = Percent_Alpha_F8;
inputs_Alpha_FC5 = Percent_Alpha_FC5;
inputs_Alpha_FC6 = Percent_Alpha_FC6;
inputs_Alpha_O1 = Percent_Alpha_O1;
inputs_Alpha_O2 = Percent_Alpha_O2;
inputs_Alpha_P7 = Percent_Alpha_P7;
inputs_Alpha_P8 = Percent_Alpha_P8;
inputs_Alpha_T7 = Percent_Alpha_T7;

inputs_Alpha_T8 = Percent_Alpha_T8;
inputs_Beta_AF3 = Percent_Beta_AF3;
inputs_Beta_AF4 = Percent_Beta_AF4;
inputs_Beta_F3 = Percent_Beta_F3;
inputs_Beta_F4 = Percent_Beta_F4;
inputs_Beta_F7 = Percent_Beta_F7;
inputs_Beta_F8 = Percent_Beta_F8;
inputs_Beta_FC5 = Percent_Beta_FC5;
inputs_Beta_FC6 = Percent_Beta_FC6;
inputs_Beta_O1 = Percent_Beta_O1;
inputs_Beta_O2 = Percent_Beta_O2;
inputs_Beta_P7 = Percent_Beta_P7;
inputs_Beta_P8 = Percent_Beta_P8;
inputs_Beta_T7 = Percent_Beta_T7;
inputs_Beta_T8 = Percent_Beta_T8;
inputs_Gamma_AF3 = Percent_Gamma_AF3;
inputs_Gamma_AF4 = Percent_Gamma_AF4;
inputs_Gamma_F3 = Percent_Gamma_F3;
inputs_Gamma_F4 = Percent_Gamma_F4;
inputs_Gamma_F7 = Percent_Gamma_F7;
inputs_Gamma_F8 = Percent_Gamma_F8;
inputs_Gamma_FC5 = Percent_Gamma_FC5;
inputs_Gamma_FC6 = Percent_Gamma_FC6;
inputs_Gamma_O1 = Percent_Gamma_O1;
inputs_Gamma_O2 = Percent_Gamma_O2;
inputs_Gamma_P7 = Percent_Gamma_P7;
inputs_Gamma_P8 = Percent_Gamma_P8;
inputs_Gamma_T7 = Percent_Gamma_T7;


```

%
for hiddenLayerSize = 1:100
% Create a Fitting Network
net = fitnet(hiddenLayerSize);

% Set up Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 75/100;
net.divideParam.valRatio = 5/100;
net.divideParam.testRatio = 20/100;

for n = 1:3

% Train the Network
net = init(net);
[net,tr] = trainbr(net,input_matrix, targets);

% Test the Network
outputs = net(input_matrix);
trOut = outputs(tr.trainInd);
tsOut = outputs(tr.testInd);
trTarg = targets(tr.trainInd);
tsTarg = targets(tr.testInd);
errors = gsubtract(outputs,targets);
performance(n) = perform(net,targets,outputs);

% View the Network
% view(net)

```

```

% Plots

% Uncomment these lines to enable various plots.

% figure, plotperform(tr)

% figure, plottrainstate(tr)

% figure, plotfit(targets,outputs)

figure, plotregression_1(targets, outputs,'Train') %The setappdata is done under this code

close figure 1

% figure, plotregression_2(tsTarg,tsOut,'Testing') %The setappdata is done under this code

% close figure 1

% figure, ploterrhist(errors)

r_Training(n, hiddenLayerSize) = getappdata(0, 'r_train');

% r_Testing(n, hiddenLayerSize) = getappdata(0, 'r_test');

mse_train(n, hiddenLayerSize) = immse(trOut, trTarg);

mse_test(n, hiddenLayerSize) = immse(tsOut, tsTarg);

end

end

%%

%Placing Results within Matrix

% r_Matrix = [r_Delta_AF3', r_Delta_AF4', r_Delta_F3', r_Delta_F4', r_Delta_F7', r_Delta_F8', r_Delta_FC5',
r_Delta_FC6', r_Delta_O1', r_Delta_O2', r_Delta_P7', r_Delta_P8', r_Delta_T7', r_Delta_T8', r_Theta_AF3',
r_Theta_AF4', r_Theta_F3', r_Theta_F4', r_Theta_F7', r_Theta_F8', r_Theta_FC5', r_Theta_FC6', r_Theta_O1',
r_Theta_O2', r_Theta_P7', r_Theta_P8', r_Theta_T7', r_Theta_T8', r_Alpha_AF3', r_Alpha_AF4', r_Alpha_F3',
r_Alpha_F4', r_Alpha_F7', r_Alpha_F8', r_Alpha_FC5', r_Alpha_FC6', r_Alpha_O1', r_Alpha_O2', r_Alpha_P7',
r_Alpha_P8', r_Alpha_T7', r_Alpha_T8', r_Beta_AF3', r_Beta_AF4', r_Beta_F3', r_Beta_F4', r_Beta_F7',
r_Beta_F8', r_Beta_FC5', r_Beta_FC6', r_Beta_O1', r_Beta_O2', r_Beta_P7', r_Beta_P8', r_Beta_T7', r_Beta_T8',
r_Gamma_AF3', r_Gamma_AF4', r_Gamma_F3', r_Gamma_F4', r_Gamma_F7', r_Gamma_F8', r_Gamma_FC5',
r_Gamma_FC6', r_Gamma_O1', r_Gamma_O2', r_Gamma_P7', r_Gamma_P8', r_Gamma_T7', r_Gamma_T8'];

```

\

`%Writing Data to Excel File`

```
xlswrite(FileName, r_Training, 'Sheet1', 'B2');
```

```
xlswrite(FileName, mse_train, 'Sheet2', 'B2');
```

```
xlswrite(FileName, mse_test, 'Sheet3', 'B2');
```

Graphs Participant X

`%Participant 1 First 2 Hours Graph`

```
clc; close all; clear;
```

```
load('net_participant_5');
```

```
load('net_participant_5_10NBR');
```

```
load('input_matrix_participant_5_first_2');
```

```
load('target_matrix_participant_5_first_2');
```

```
load('input_matrix_participant_5_3');
```

```
load('target_matrix_participant_5_3');
```

```
%%
```

```
BGL_Max = 110;
```

```
%%
```

```
net_output = net_participant_5(input_matrix)*BGL_Max;
```

```
x_time = 1:214:105;
```

```
Upper_Limit = (targets + targets*0.2)*BGL_Max;
```

```
Lower_Limit = (targets - targets*0.2)*BGL_Max;
```

```
Upper_Limit_1 = (targets_test + targets_test*0.2)*BGL_Max;
```

```
Lower_Limit_1 = (targets_test - targets_test*0.2)*BGL_Max;
```

```
figure
```

```
plot(x_time, targets*BGL_Max, 'LineWidth', 2);
```

```
hold on
```

```
plot(x_time, sgolayfilt(net_output, 1, 51), 'LineWidth', 2);
```

```
plot(x_time, Upper_Limit, 'k--');
```

```
plot(x_time, Lower_Limit, 'k--');
```

```
xlabel('Time (mins)');
```

```
ylabel('BGL (mg/dl)');
```

```
title('Network Predicting Data Trained On (Participant 5)');
```

```
% legend('Target BGL Data', 'BGL Data Predicted from Network', '+- Boundaries');
```

```
set(gcf, 'color', 'w');
```

```
set(gca, 'FontSize', 15);
```

```
net_output = net_participant_5(input_matrix_test)*BGL_Max;
```

```
x_time_1 = 1:.27:45;
```

```
figure
```

```
plot(x_time_1, targets_test*BGL_Max, 'LineWidth', 2);
```

```
hold on
```

```
plot(x_time_1, sgolayfilt(net_output, 1, 51), 'LineWidth', 2);
```

```
plot(x_time_1, Upper_Limit_1, 'k--');
```

```
plot(x_time_1, Lower_Limit_1, 'k--');
```

```
xlabel('Time (mins)');
```

```
ylabel('BGL (mg/dl)');
```

```
title('Network Predicting Data never seen (Participant 5)');
```

```
% legend('Target BGL Data', 'BGL Data Predicted from Network', '+- Boundaries');
```

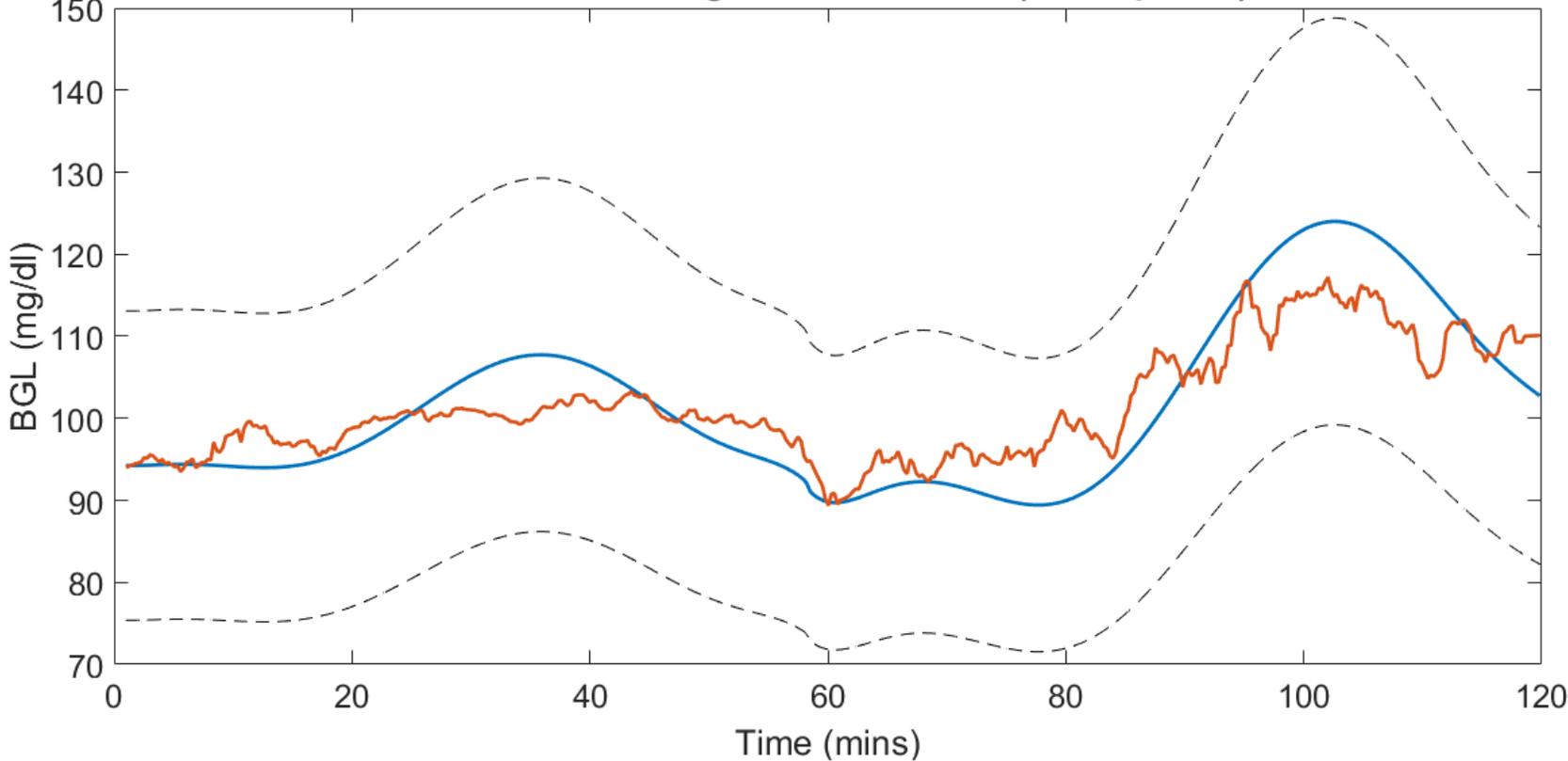
、

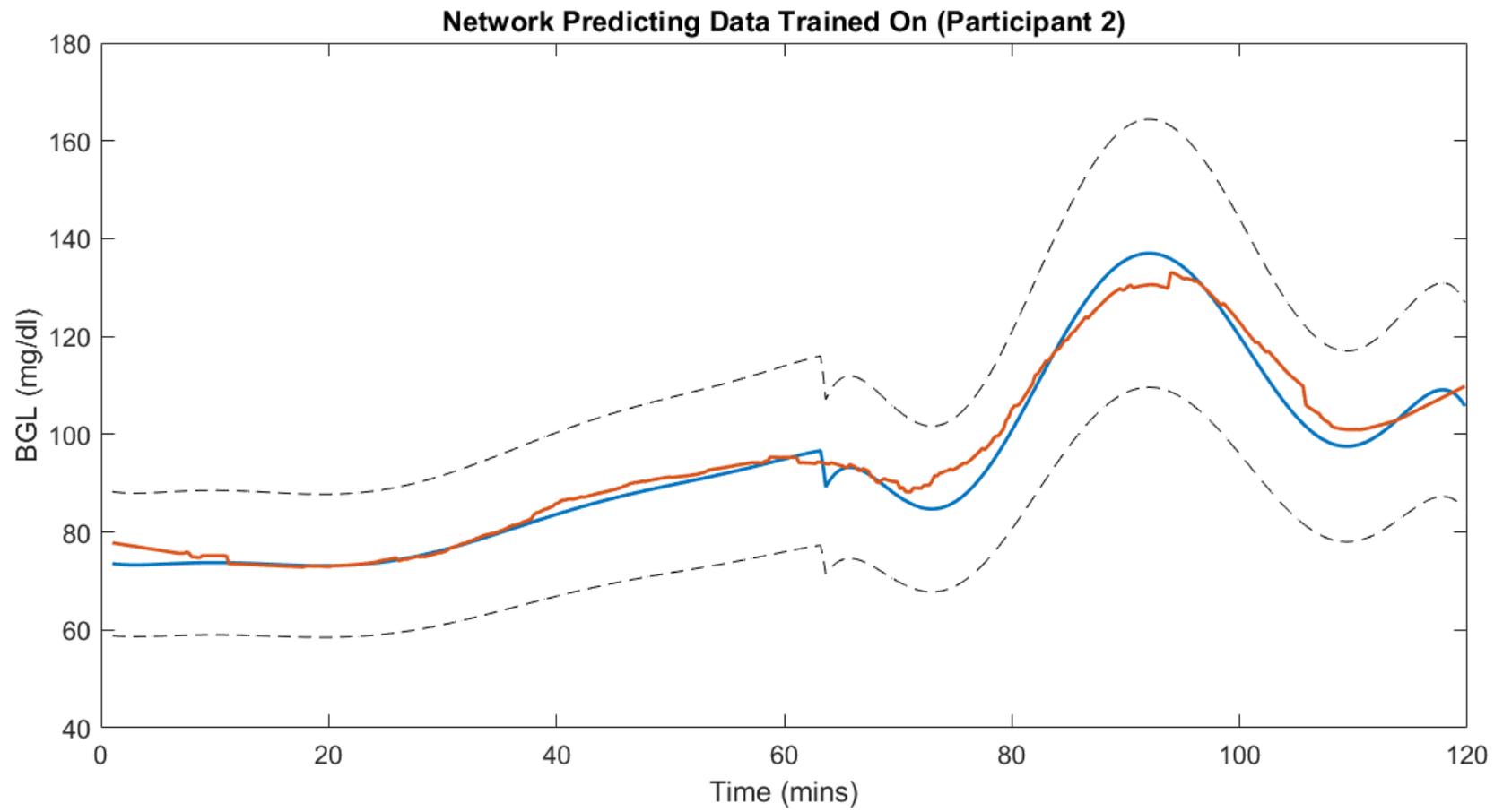
```
set(gcf,'color','w');
```

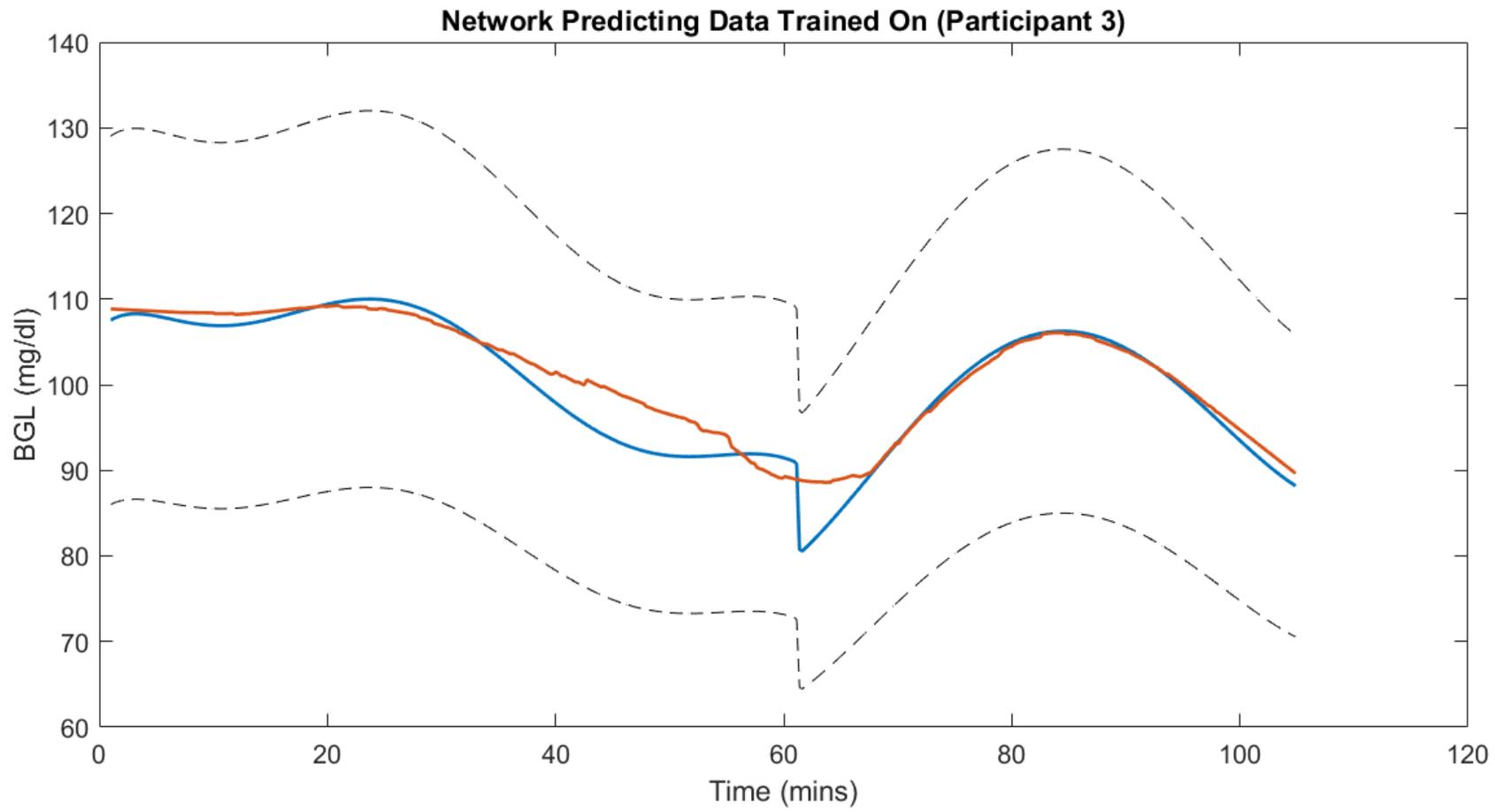
```
set(gca,'FontSize', 15);
```

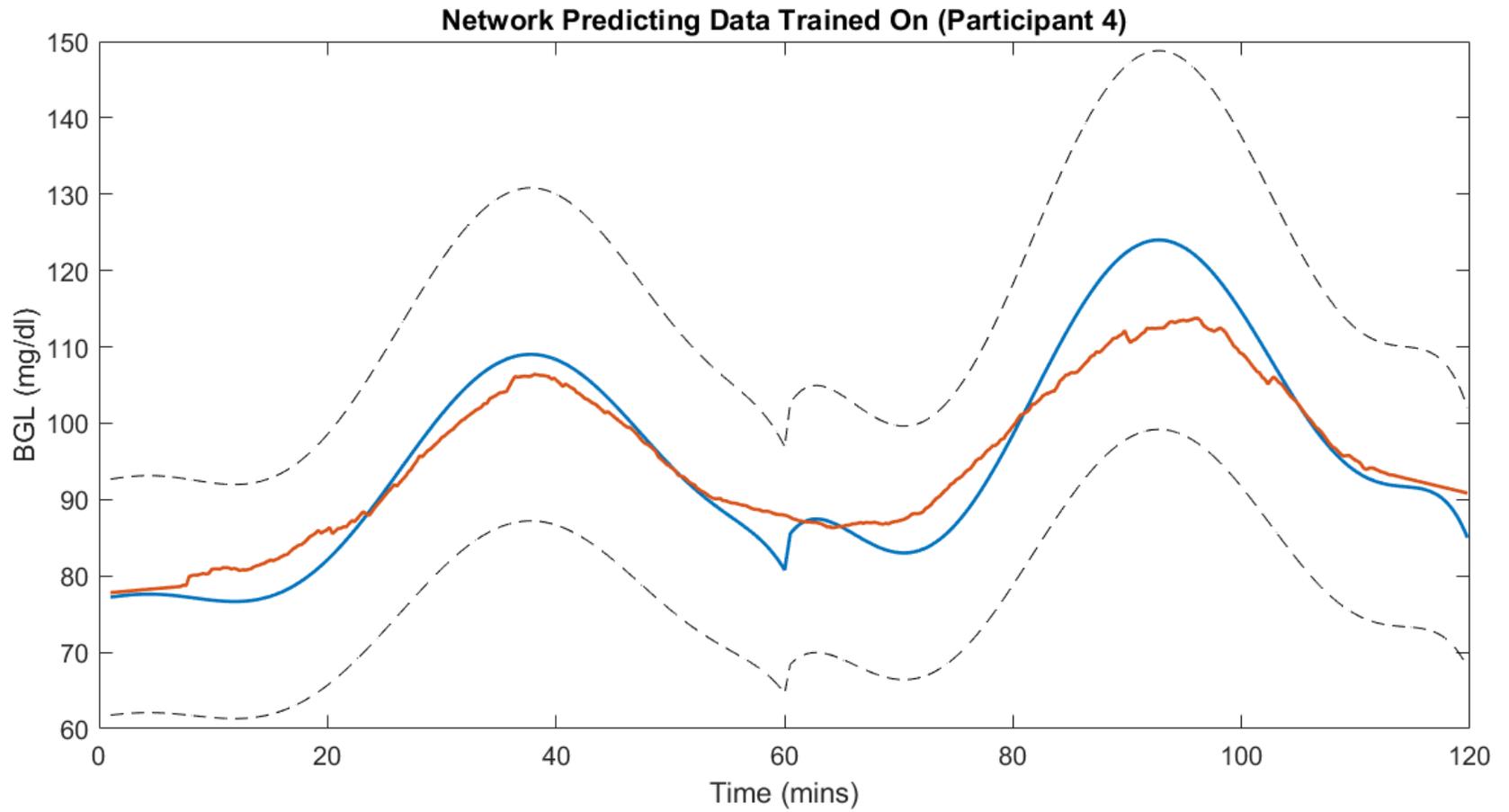
APPENDIX D: Network Figures Depicting Training/Testing Graphs

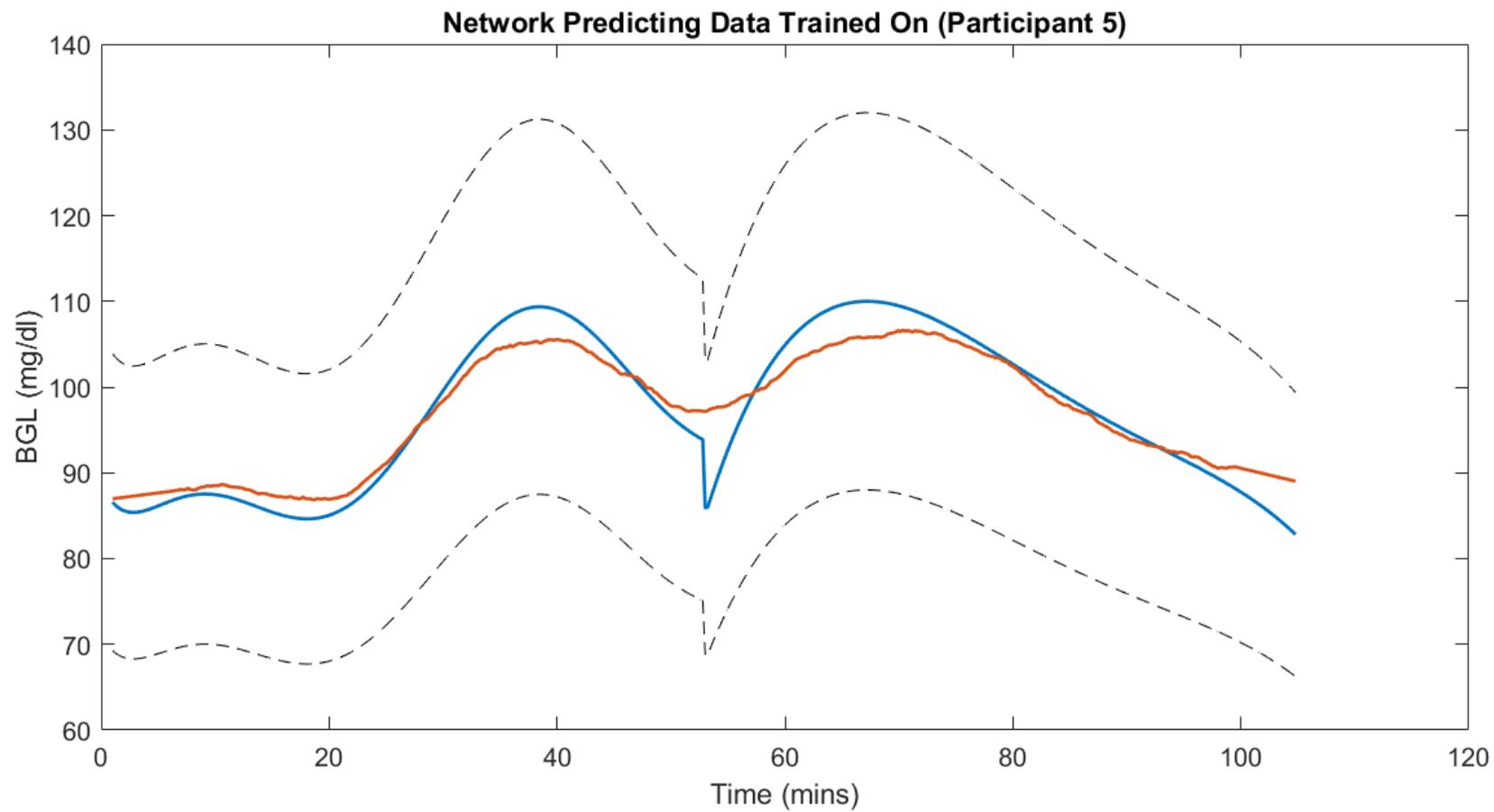
Network Predicting Data Trained On (Participant 1)

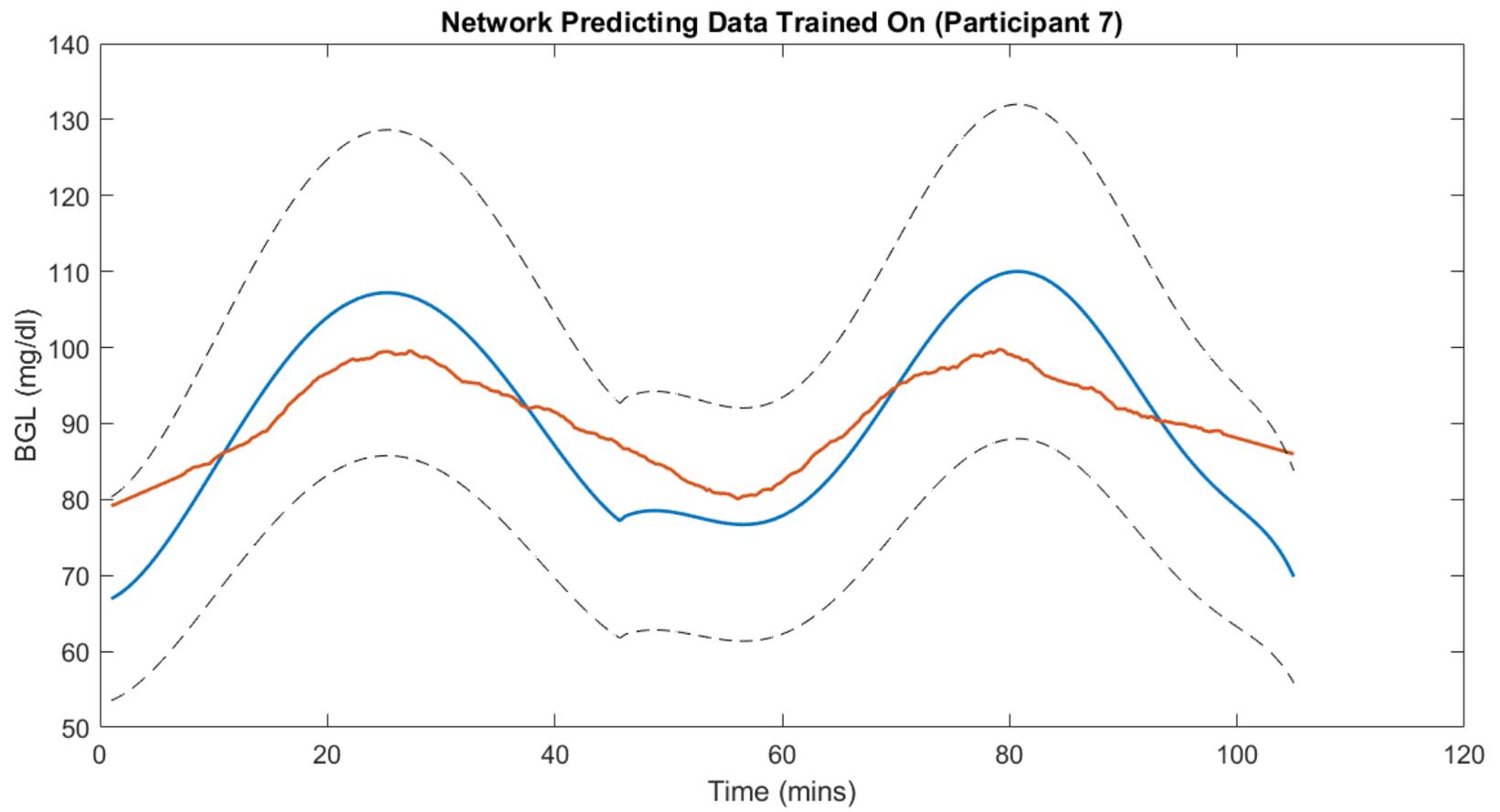


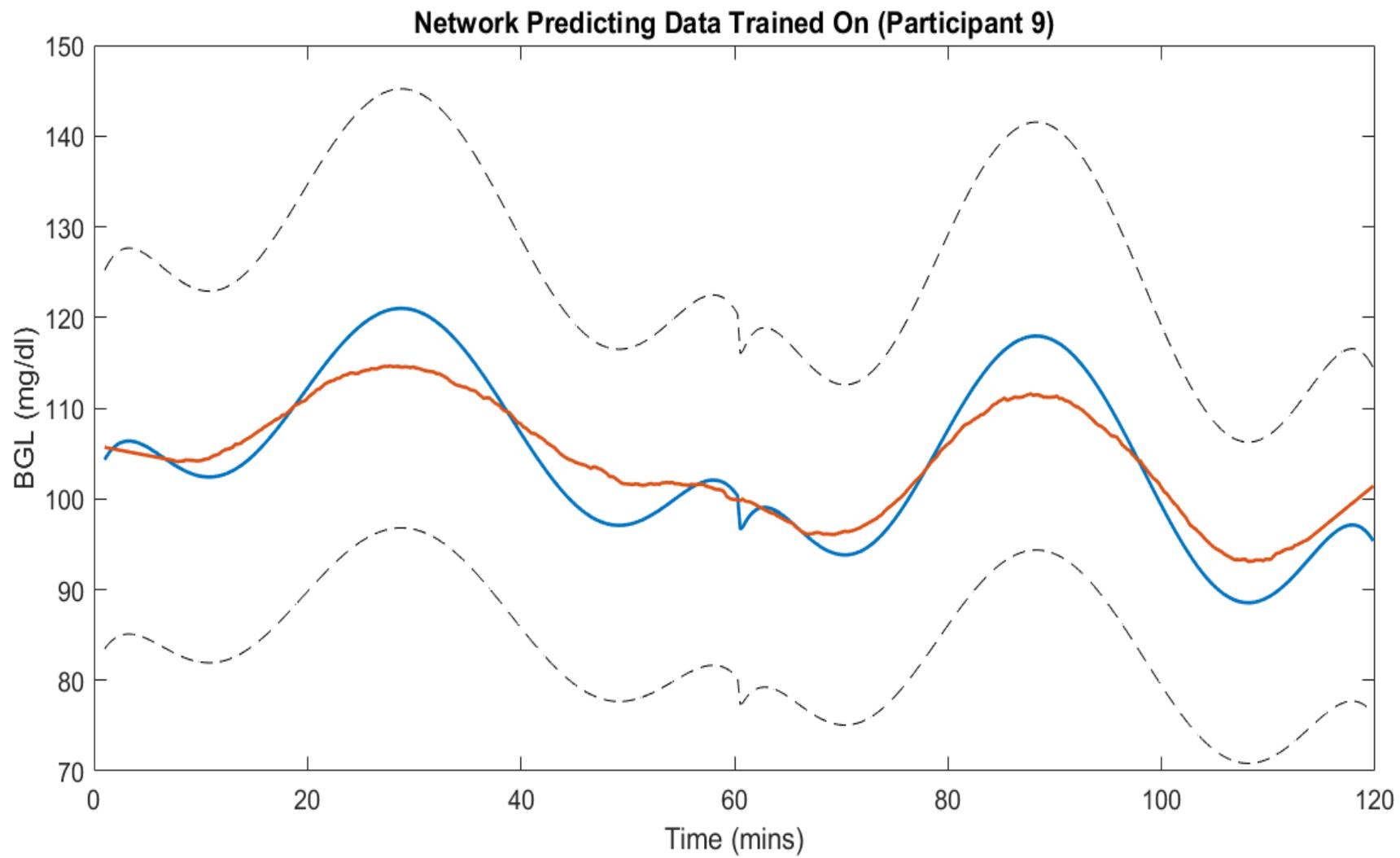


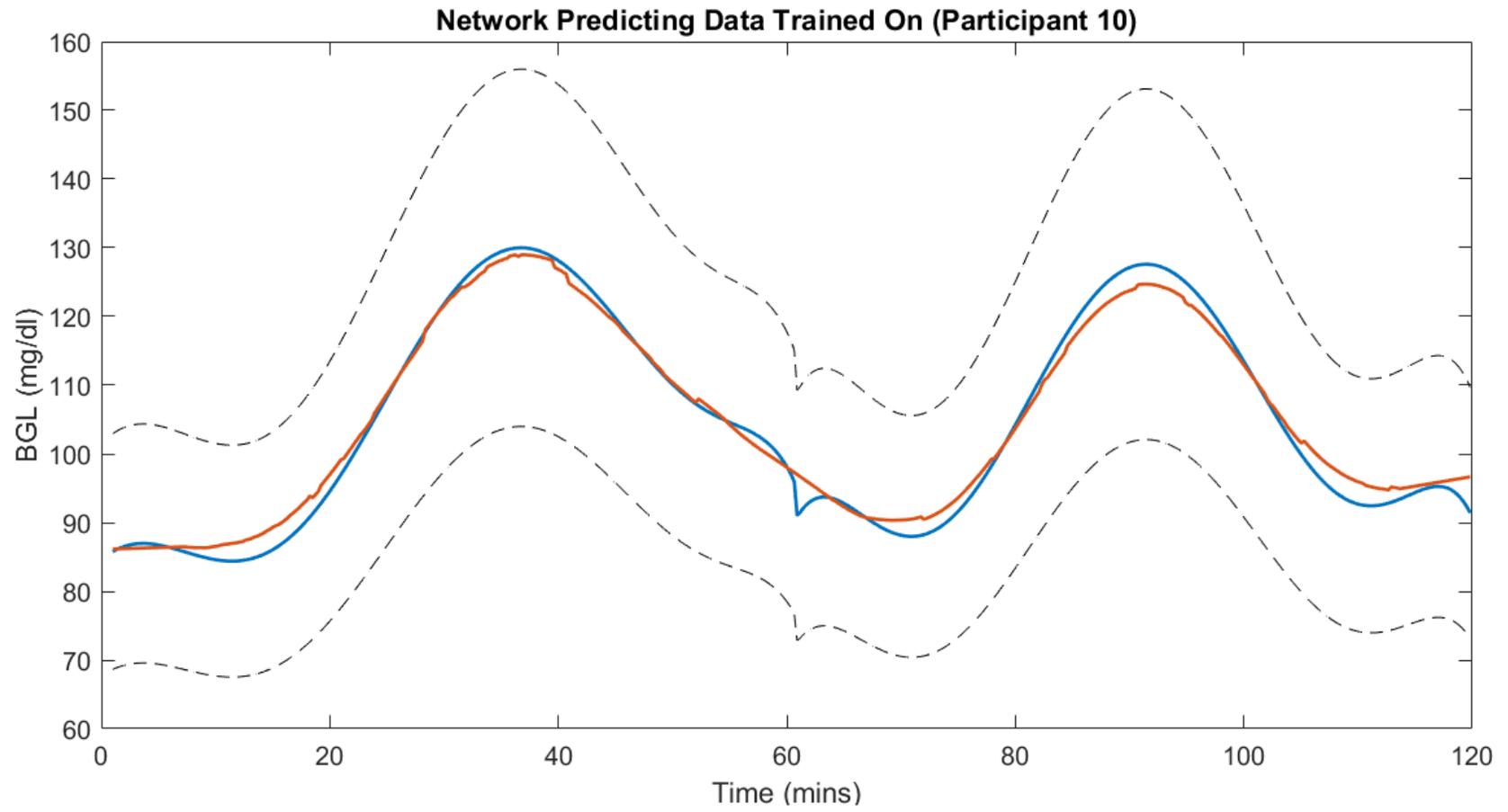




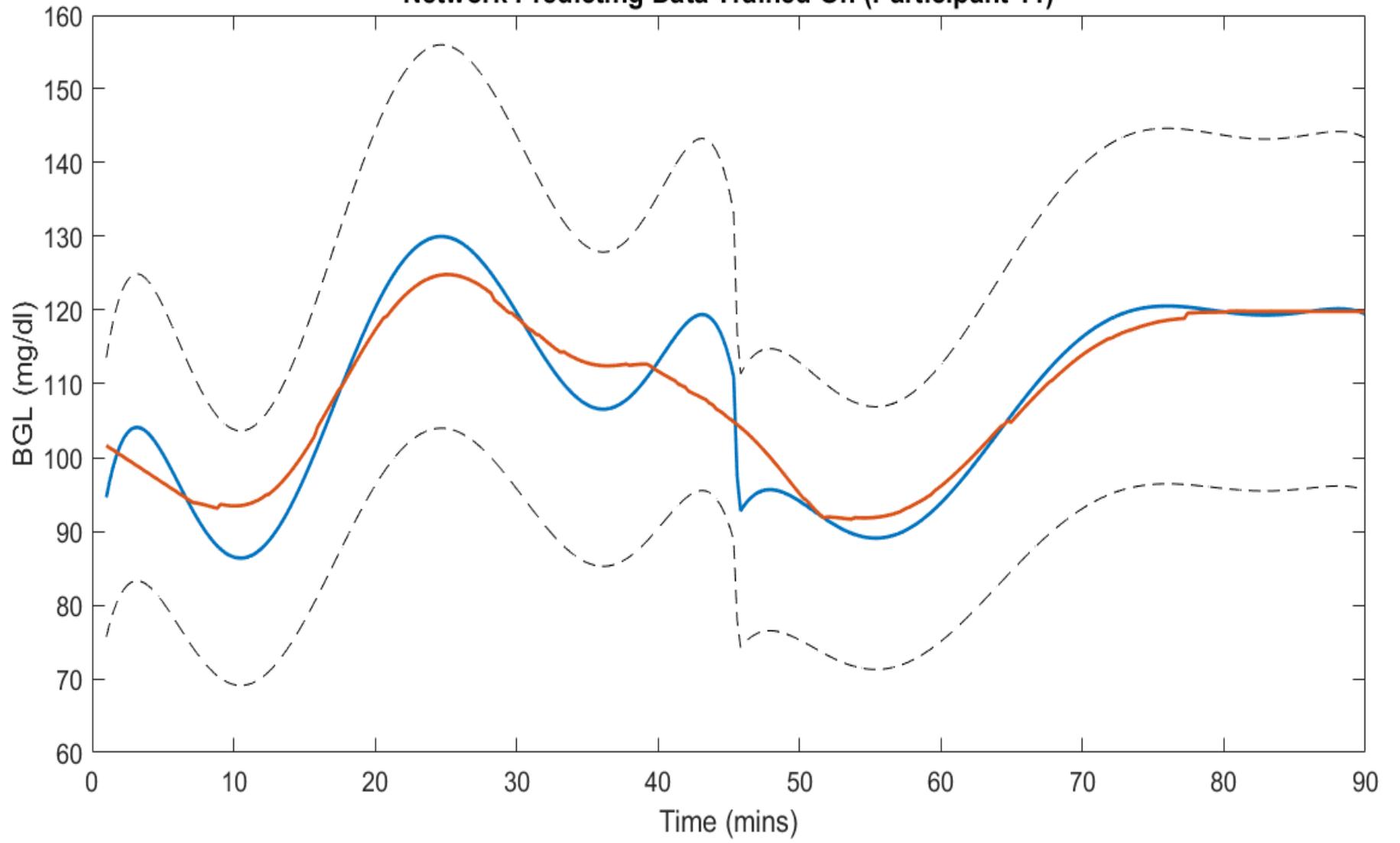




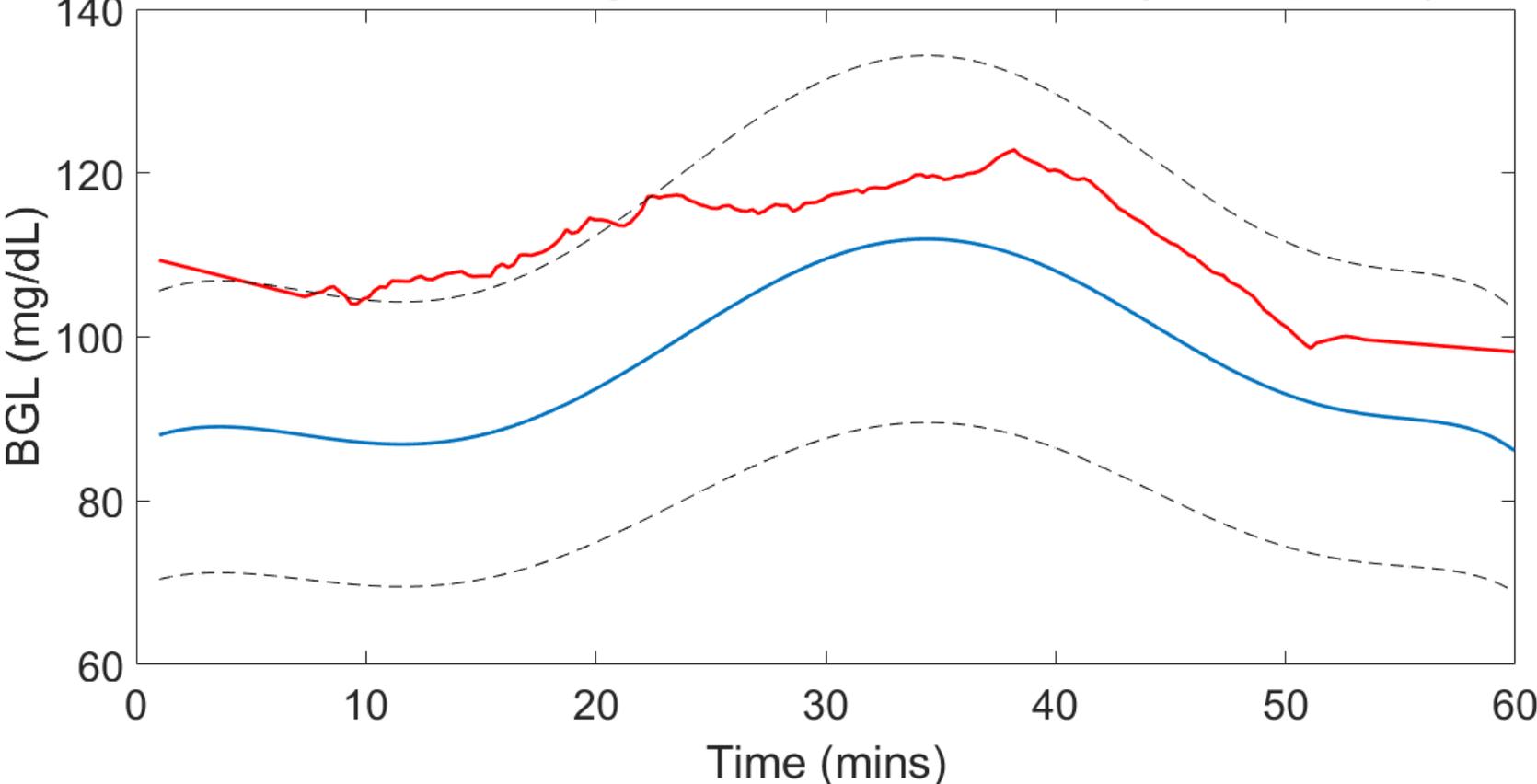




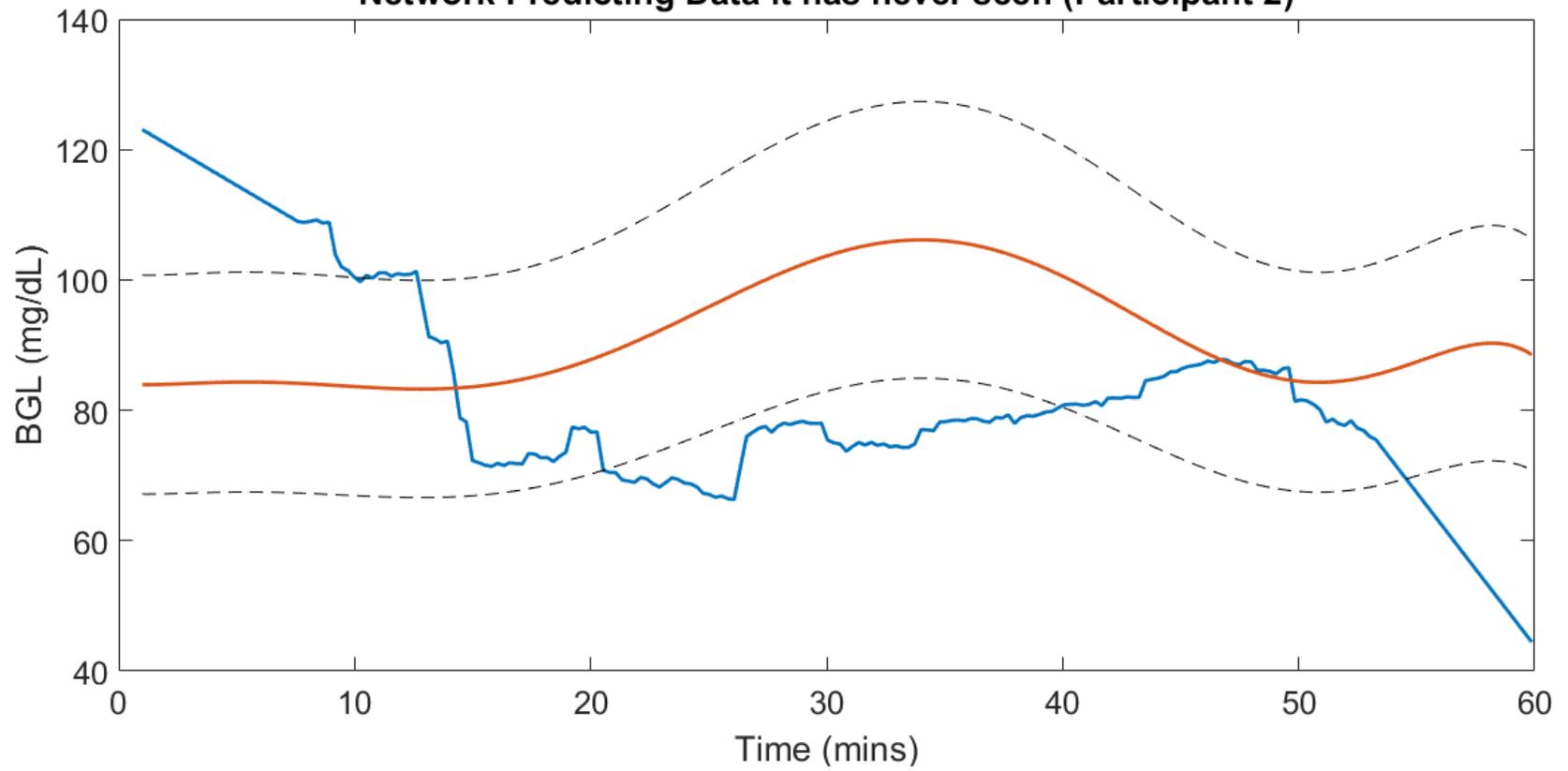
Network Predicting Data Trained On (Participant 11)



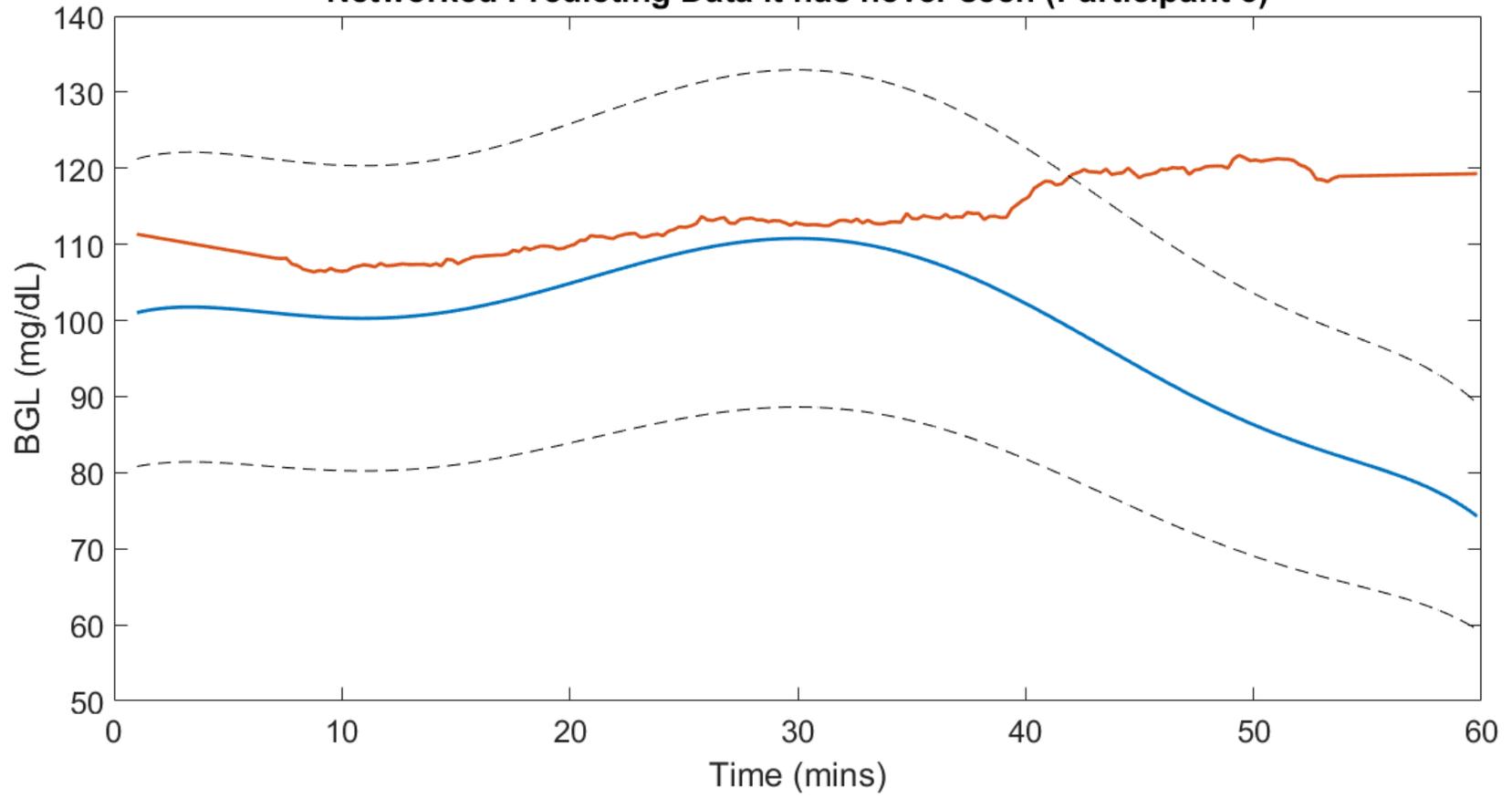
Network Predicting Data It has never seen (Participant 1)

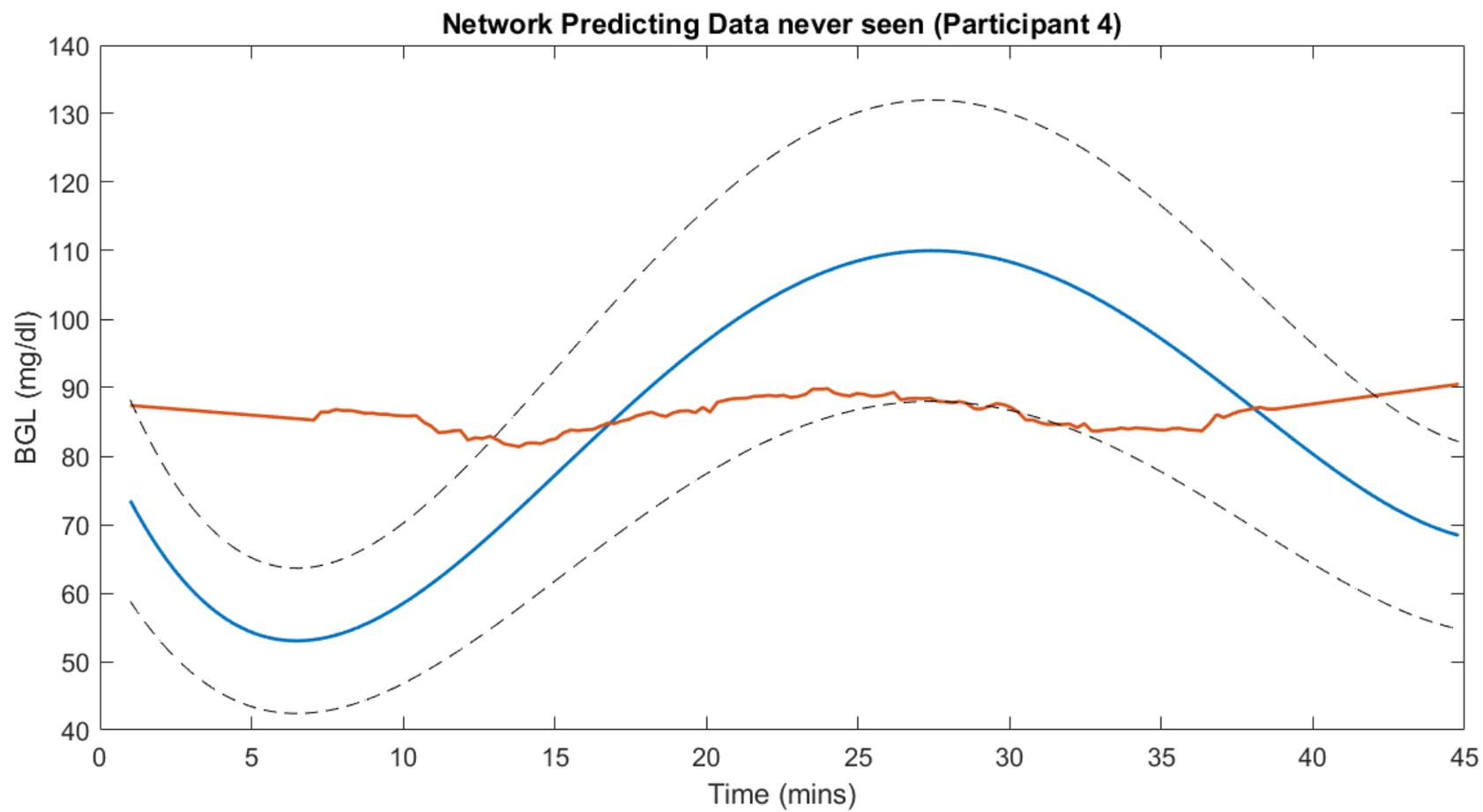


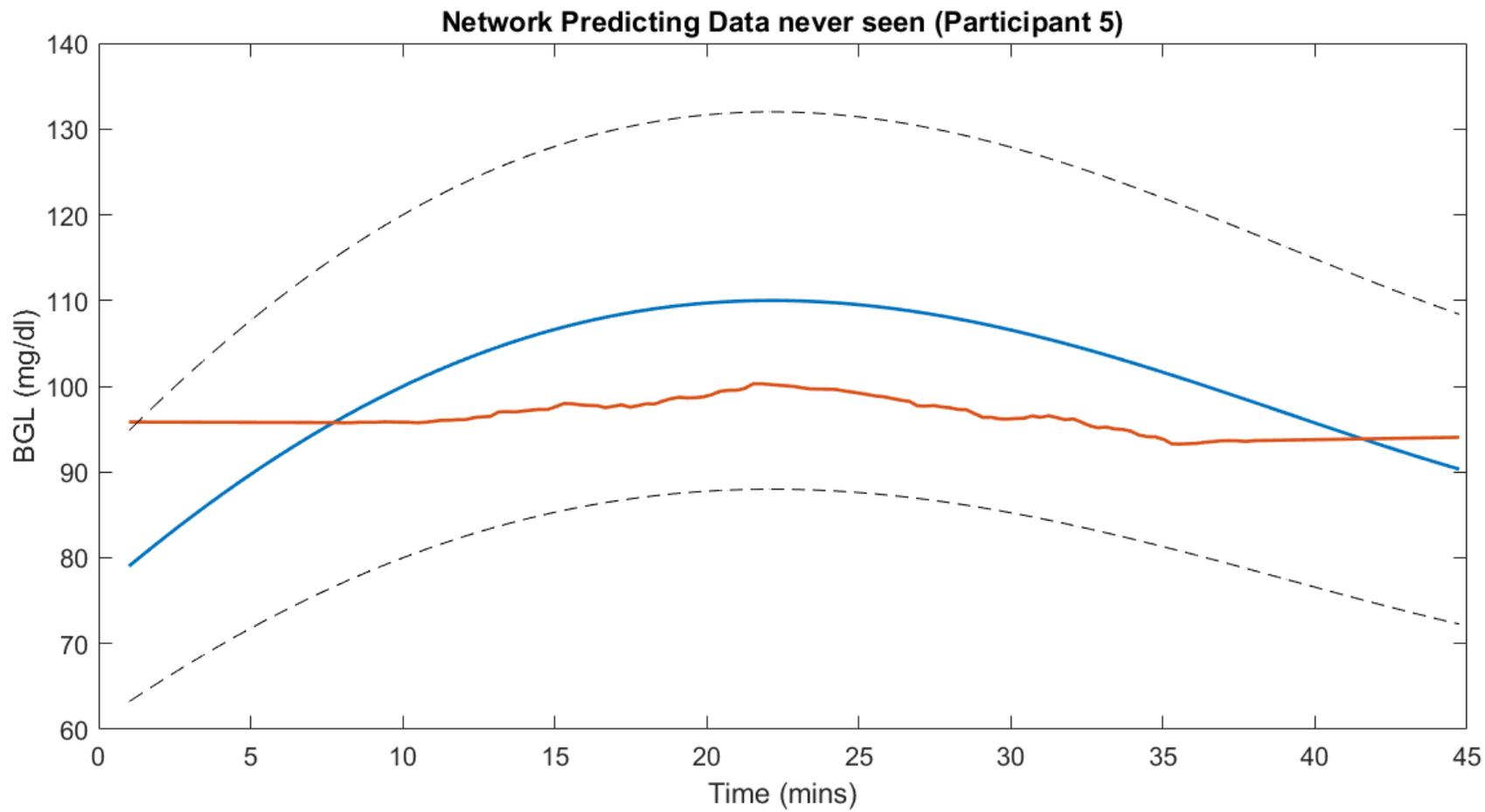
Network Predicting Data it has never seen (Participant 2)

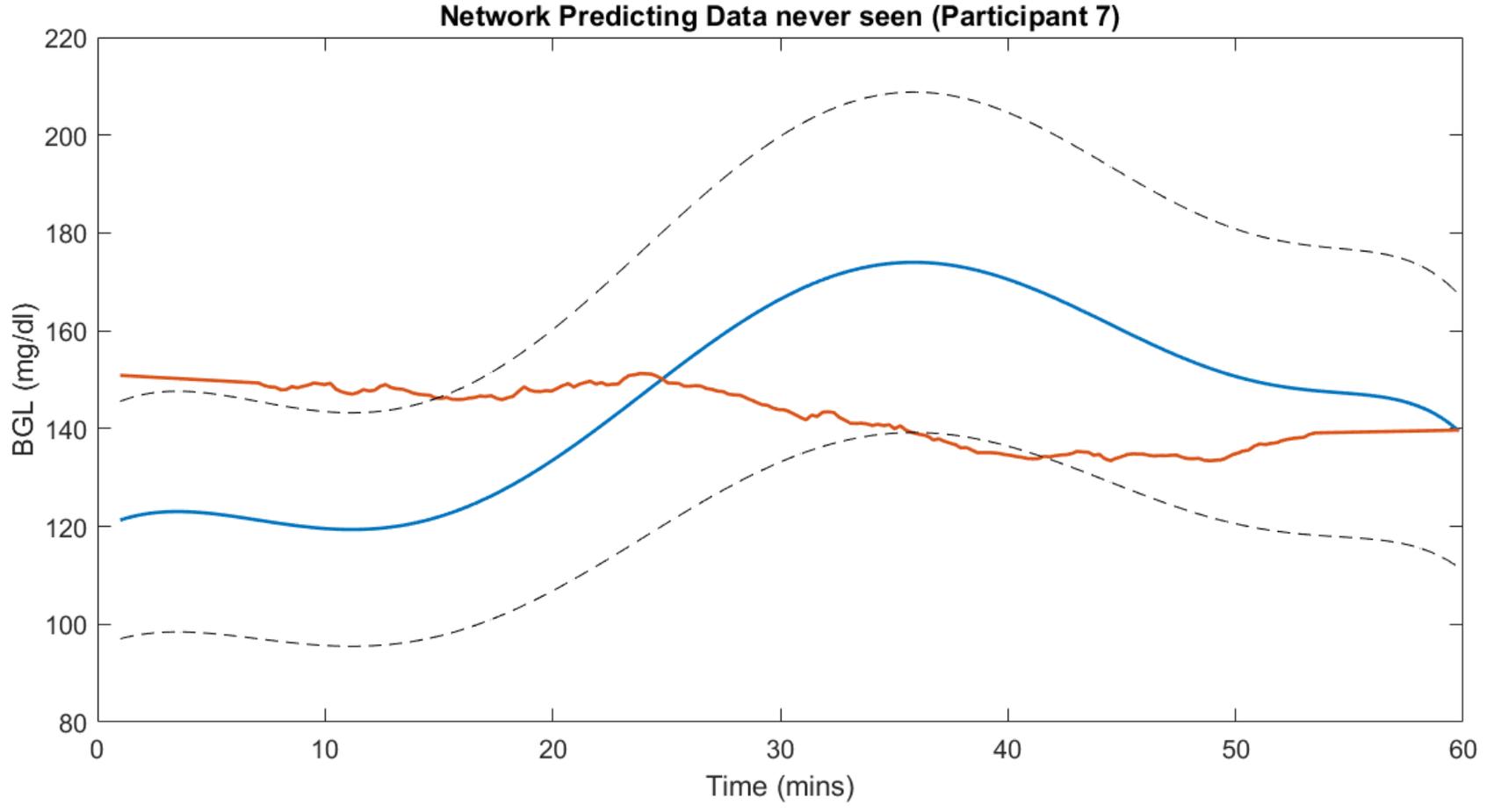


Networked Predicting Data it has never seen (Participant 3)

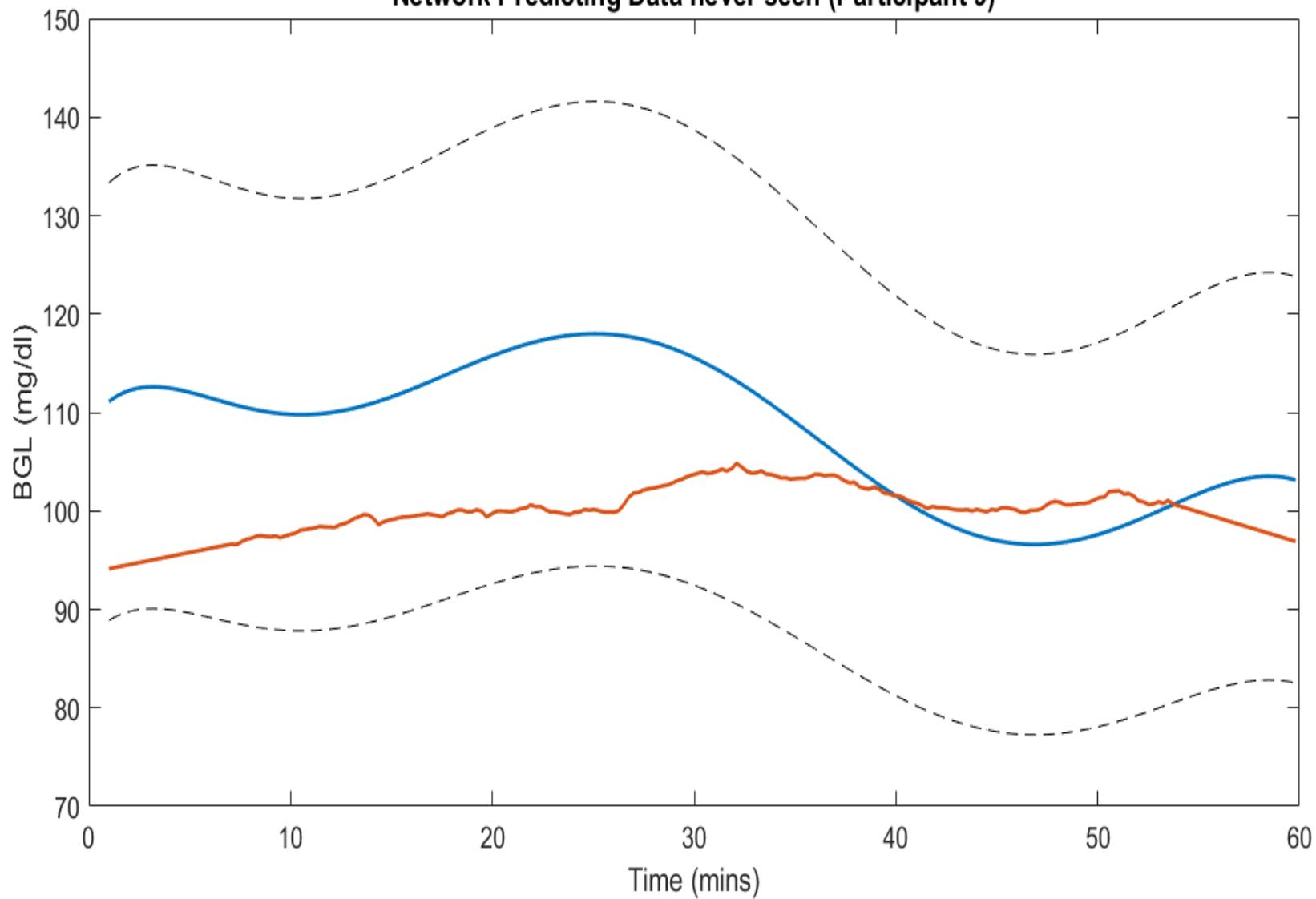


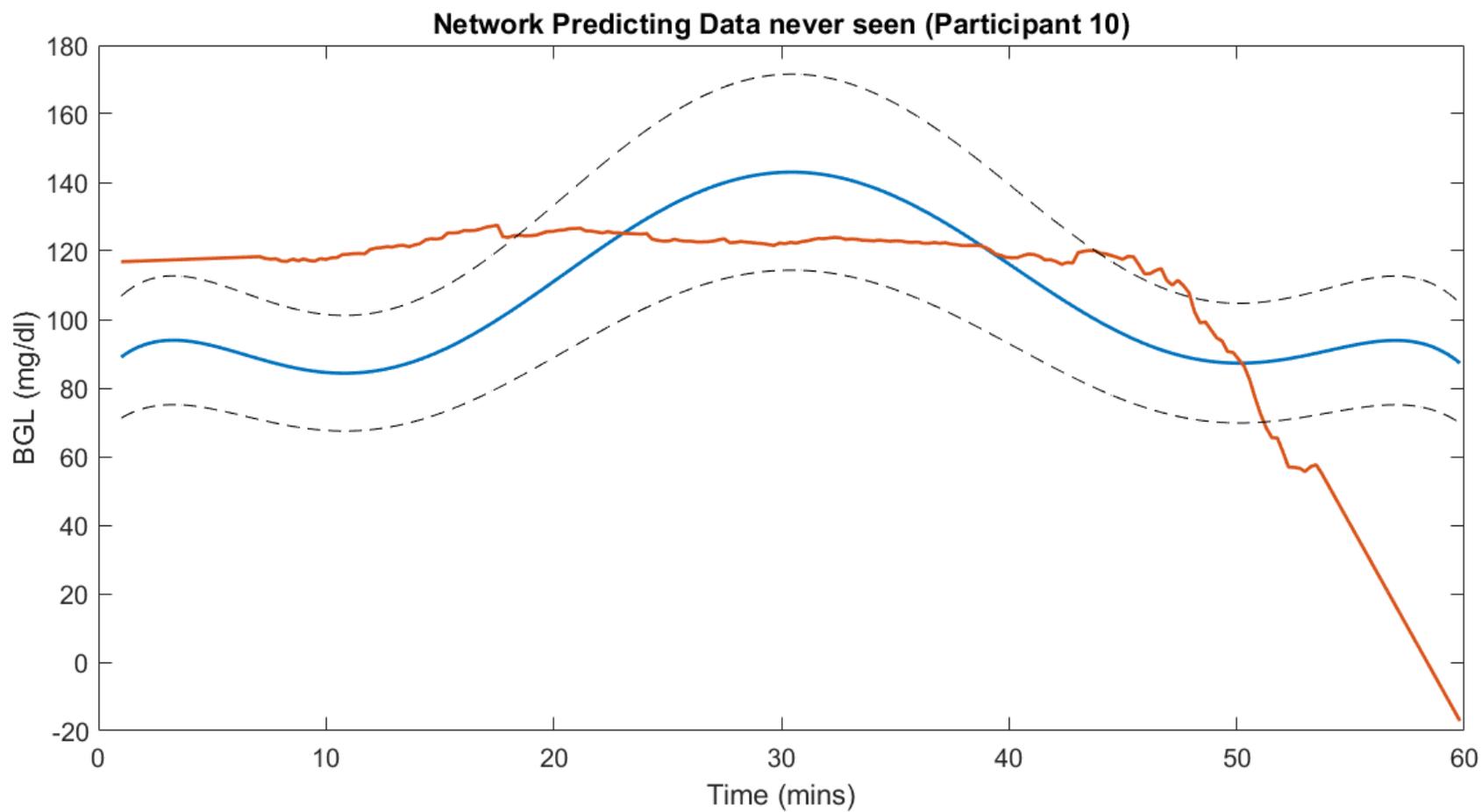






Network Predicting Data never seen (Participant 9)





Network Predicting Data never seen (Participant 11)

