EATING AND EXERCISE DETECTION WITH CONTINUOUS GLUCOSE MONITORS

An Undergraduate Research Scholars Thesis

by

TONY YANG

Submitted to the LAUNCH: Undergraduate Research office at
Texas A&M University
in partial fulfillment of requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Faculty Research Advisor:	Dr. Bobak Mortazavi

May 2022

Major:	Computer Science

RESEARCH COMPLIANCE CERTIFICATION

Research activities involving the use of human subjects, vertebrate animals, and/or biohazards must be reviewed and approved by the appropriate Texas A&M University regulatory research committee (i.e., IRB, IACUC, IBC) before the activity can commence. This requirement applies to activities conducted at Texas A&M and to activities conducted at non-Texas A&M facilities or institutions. In both cases, students are responsible for working with the relevant Texas A&M research compliance program to ensure and document that all Texas A&M compliance obligations are met before the study begins.

I, Tony Yang, certify that all research compliance requirements related to this Undergraduate Research Scholars thesis have been addressed with my Research Faculty Advisor prior to the collection of any data used in this final thesis submission.

This project required approval from the Texas A&M University Research Compliance & Biosafety office.

TAMU IRB #: 2019-0793 Approval Date: 09/02/2020 Expiration Date: 07/04/2022

TABLE OF CONTENTS

Page

ABSTRACT

Eating and Exercise Detection with Continuous Glucose Monitors

Tony Yang
Department of Computer Science & Engineering
Texas A&M University

Research Faculty Advisor: Dr. Bobak Mortazavi
Department of Computer Science & Engineering
Texas A&M University

Eating and exercise detection using continuous glucose monitor (CGM) signals is key to provide recommendations for a healthy lifestyle. However, this can be challenging given imbalanced data and other contexts. Previous works have used accelerometers, gyroscopes, glucose monitors, and other sensors but not necessarily all three plus others combined. Therefore, I aim to build a model by testing various techniques and testing glucose along with different statistical body measurements, such as electrodermal activity, heart rate, blood volume, accelerometer, gyroscope, etc. A sliding window is used to extract statistical measures from each body measurement, such as standard deviation, mean, and range to look for patterns correlated to eating and exercise. I select an extreme gradient boosted decision tree algorithm with Synthetic Minority Oversampling Technique. I compare the performance of just solely using glucose and then adding more sensory data and discovered that there is not consistent change in performance. I also adjusted the window and overlap to compare eating detection performance and found that there is not a concrete impact on the performance. Furthermore, I performed exercise detection and compare with and without CGM. There appears to be no significant performance difference

with or without glucose. In addition to eating detection, I also examine for correlation between glucose variation and exercise moments. I finally conclude that it is not feasibly possible to detect eating with my current methods. However, for exercise detection, I can produce better detection results compared to eating, but my current method for detecting correlations between glucose levels and exercise moments can be later improved.

ACKNOWLEDGEMENTS

Contributors

I would like to thank my faculty advisor, Dr. Bobak Mortazavi and my Graduate Mentors, Dr. Sorush Omidvar and Sicong Huang, for their guidance and support throughout the course of this research.

Thanks also go to my friends and colleagues and the department faculty and staff for making my time at Texas A&M University a great experience.

The data analyzed/used for my undergraduate thesis were provided by the Systems and Technology for Medicine and IoT Lab. The analyses depicted in my undergraduate thesis were conducted in part by Systems and Technology for Medicine and IoT Lab and were published in and these data are unpublished.

All other work conducted for the thesis was completed by the student independently.

Funding Sources

No funding was provided.

1. INTRODUCTION

Eating and exercising are some of human's essential activities that have a direct impact on glucose levels. In addition, eating and exercising properly (i.e., consuming the right number of calories, having sufficient nutrients, eating the correct foods, exercising the right amount of time, etc.) is also a key component to healthy glucose levels. Not having proper diet and exercise could lead to major health problems, such as improper weight (overweight or underweight), heart diseases, diabetes, and/or cancer [6] [21]. Certain levels of glucose may be considered too low or too high for exercise as a result of diabetes or hyperglycemia [3]. Therefore, it is essential to be able to detect when someone is eating and exercising so that the proper diet and exercise recommendations can be made based on the glucose levels.

Detecting when someone is eating can be difficult because there are other activities (e.g., bathing, cooking, getting dressed, smoking, socializing, etc.) that could mimic eating when observing one or two features measured from various human body statistics [2] [7] [13]. These features include glucose levels, motion (accelerometer and gyroscope) sensors, galvanic skin response, body temperature, and among other various statistics. Furthermore, each person's body composition is different and could lead to different readings based on certain health problems. For example, smoking will increase blood sugar levels since nicotine causes an increase in blood sugar levels just like with eating [18]. Therefore, it might not be sufficient to just look at one feature alone.

Another major issue is the fact that people are not eating or exercising most of the time. Consequently, I have an imbalanced dataset with most times being classified as not eating or exercise [19]. So, it would be naive to say that a model does a good job purely based on accuracy alone. A 90%+ accuracy score can easily be achieved with a model always predicting “not eating/exercising” along with other models that don’t predict the “eating/exercising” output as accurately. I, therefore, need to look closely at how well a model can predict the “eating/exercising” moments using other metrics (e.g., F1 Score, ROC AUC Score, Balanced Accuracy, PR AUC Score, etc.) that examine how well a model performs in detecting true positives (e.g., eating/exercise) in addition to accuracy.

1.1 Literature Review

When I examine how glucose generally changes with exercise, glucose levels do lower with steady state cardio exercises due to the muscles using them as energy but certain intense exercises may result in an increase in glucose levels due to stress hormones being released [9] [15] [23]. Eating carbohydrates increases in blood glucose levels [3]. However, the amount of variation, if any, can vary from person to person even with the same foods consumed [12].

Exercise detection in previous works have used accelerometers or cameras as a means of detecting exercise. In one particular experiment, a camera was used to detect a full body sense of motion [11]. This study was extremely successful in that it had managed to differentiate between exercise from other activities 84.6 percent of the time with the use of neural networks to train a model for exercise detection [11]. However, the major downside to this particular study was that this was conducted exclusively at a gym rather than in a day-to-day life setting. As a result, the specific detection patterns that are used in this particular study are not necessarily a true representation of what could be exercise patterns on the street. Another experiment that presents a similar problem is one that all participants performed the same exercises in a controlled setting using an accelerometer in order to compare exercise vs. non-exercise. They achieved a very high metric of over 95 percent in both recall and precision [13]. This experiment had used a principal component analysis (PCA) as a means of being able to train a model to differentiate exercise from non-exercise [13]. A similar experiment had also used an accelerometer sensor in which participants engaged in repetitions of the same exercise but with Long Short-Term Memory (LSTM) Neural Networks used to train and detect specific exercise drills. Results varied from depending on the exercise performed and had F1 scores were in the range of 0.595 through 0.989 [8].

While there are some research projects centered around exercise detection, there are many more around eating detection. Many research projects aimed at detecting eating in the past have used data from the accelerometer as the most common feature followed by data from the gyroscope not including any glucose sensor or energy expenditure sensor [2] [5] [20]. Other methods include a microphone (using recurrent neural networks and LSTM to detect chewing swallowing, biting, and other eating motions), piezoelectric sensor, radio-frequency transmitter and receiver, camera, and among other types of sensors [1] [2] [10]. In one study, they used a three-axis accelerometer on a smartwatch along with short questionnaires that examines contextual information about individuals. This produced an F1 score of 87.3% for real-time eating. They used a python library in sklearn random forest classifier offline to detect eating gestures (as well as non-eating motions). Then, they used a threshold-based approach in which if 20 eating gestures were detected in the span of 15 minutes, a questionnaire would pop up for the user to verify if the model predicted correctly [14]. The gyroscope sensor was the second most used sensor aside from glucose monitors. In another experiment, both the accelerometer and gyroscope were used. The accelerometer has a higher recall, and the gyroscope has higher precision. When combined, however, they both produce better recall, precision, and F1 score compared to when they are used individually [19]. In another study, they used an air microphone sensor and photoplethysmogram (blood volume) sensor attached to the ear along with an accelerometer to detect eating moments. They used support vector machines with radial basis function kernel to build the model. This produced a result with a 0.938 accuracy and a recall of 0.807 thus leading to an F1 score of 0.800. This did include more features and achieve a high F1 score. However, this sensor would not be truly feasible in the real world since participants oftentimes did not wear sensors due to the discomfort many of them expressed [16] [22]. Most work has been focusing on using motion sensors. However, there are still some studies that have just used only used glucose monitoring. They have used Kalman filter estimation, simulation-based explanation, backward difference method, and the second derivative of glucose to be able to detect meals [4] [17].

My work attempts to look at glucose data and use a sliding time window to extract statistical features, such as standard deviation, and examine patterns that may correlate with eating and exercise. I want to compare how CGM relates to exercise and eating since glucose fluctuates with those two activities and are directly related to a person's health. However, using CGM monitors alone might not be enough since certain foods and exercises affect glucose levels differently in each person. Therefore, I plan to combine several sensors, including the glucose, gyroscope, accelerometer, heart rate, body temperature, and others, rather than just a few select features in previous works to derive more features. I build different models for each participant that will be used to detect eating and exercise activities and distinguish them from non-eating and non-exercise moments and will compare the performance.

1.2 Problem Formulation

The main idea is to look at the correlation of exercise and eating with CGM signals. I am also examining for patterns in the data that could correspond to when someone is eating, such as an increase in glucose levels, body temperature, heart rate. I am also looking for any motion sensor data measured from the accelerometer that could mimic a hand moving from food sitting on the table to putting the food in a person’s mouth. For the case of exercise, I am looking for, most importantly, a correlation where I see a decrease in glucose levels and then a spike during or shortly after exercise. When intense exercise is occurring, I also expect to see an increase in skin electrodermal activity and body temperature.

However, it is entirely possible that other daily activities (e.g., stress causing a spike in blood sugar, smoking leading to an increase or decrease in glucose) could mimic certain CGM patterns in the features in eating and exercise, resulting in misclassification. Therefore, I look at the other measurements besides CGM to help distinguish other daily activities from eating and exercise but even then, that still might not work. In addition, each person has a different body composition (i.e., weight, height, body fat percentage, etc.) and some might have certain health conditions, which may result in the features producing different results when performing daily activities. For example, those with diabetes tend to see a higher increase in blood sugar levels or those with generalized anxiety disorder might have high heart rates at certain times. So, there isn’t one model that can work on the whole population.

Additionally, the data were collected from participants who are living their day-to-day lives. This results in a highly unbalanced label of non-eating and non-exercise being overly represented since neither activity are dominant in a normal person’s daily life. Therefore, it is just as important to be able to accurately capture the eating and exercise moments just as it is to be able to capture the non-eating and non-exercise moments.

2. METHODS

2.1 Glucose Data Description

I have two sensors (Dexcom and Abbot) that measure glucose, both of which are continuous glucose monitors. The Dexcom monitor collects a reading every five minutes while the Abbot reads every fifteen minutes. With them both being continuous glucose monitors, I cannot use measurements from both monitors to be in the same model since they both read at different rates and are both monitoring the same feature, glucose in this case.

2.2 Algorithm Selection

Next, I need to determine which machine learning method would be best to model the data. I decided against using a neural network since I do not have enough data to train a neural network. In addition, neural networks can also take a long time to train and test. Therefore, neural networks are not time efficient for my purposes. I then decided on using an extreme gradient boosted decision tree classifier (See Figure 2.1) since I had multiple features, limited data, and complex patterns in data that could correlate with eating, exercise, and other activities (e.g., driving, sedentary work). I choose this because it is simple enough to determine eating vs non-eating (or exercise vs. non-exercise) moments using each participant’s features. I decided that I will set a maximum depth of ten as to not train the tree to become too specific for certain features due to unbalanced data. To ensure I get consistent output for every runtime of the code, a random state for each model is set.

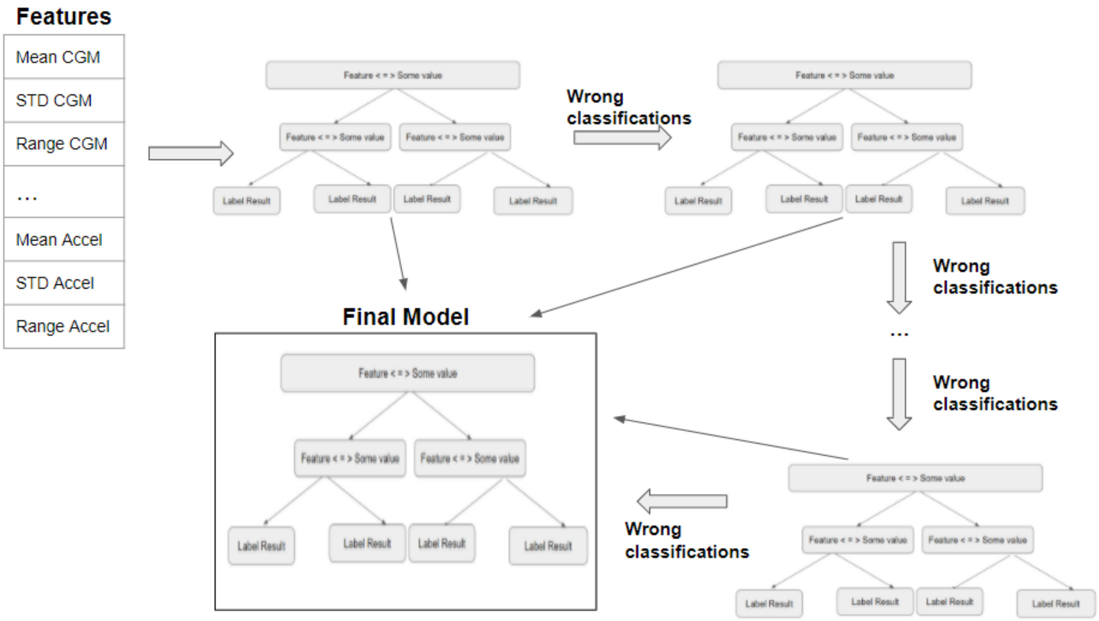


Figure 2.1

The process of how extreme gradient boosted trees train a model.

2.3 Handling Unbalanced Data

Then, I must decide how to handle unbalanced classifications. I am trying to detect exercise and eating. Thus, my binary classifications are eating vs. non-eating and exercise vs. non-exercise. I concluded that using the Synthetic Minority Oversampling Technique (SMOTE) was my best option since the technique can generate similar data to help train my model. This was better than under sampling the majority data since that would discount gathered true data. Over sampling my underrepresented classification of eating and exercise would not be as effective since it might not account for similar patterns in features. Under sampling my overrepresented classification of non-eating and non-exercise would not be effective since it does not mimic the similar scenarios like SMOTE seen in Figure 2.2.

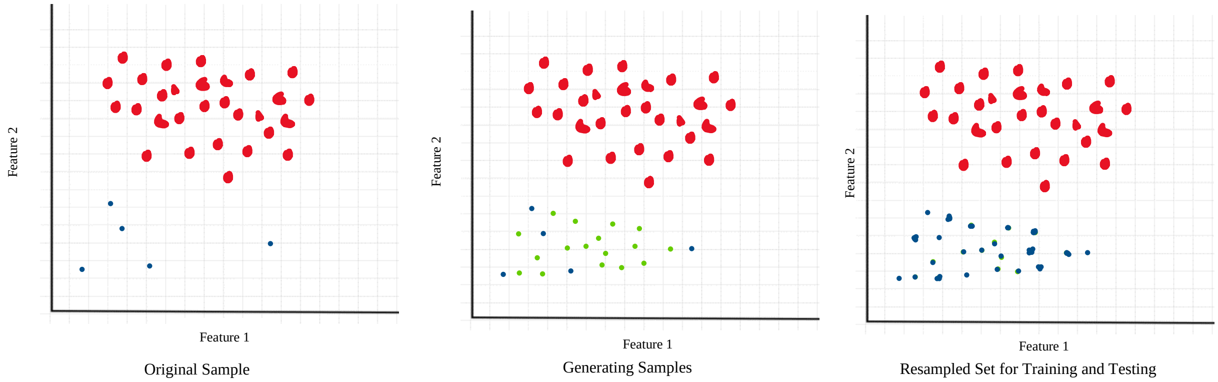


Figure 2.2

A two-feature example of how an unrepresented class is resampled by generating samples with similar features. The red represents the overrepresented data. The blue represents the original unbalanced data and the green represents the generated samples.

2.4 Handling Time Differences in Timestamps of Data Logs

One of the other major challenges that was posed is that the dates and times for the sensors were not always on the same time zone. In addition, the data was also collected during November 6th, which is also the day that Daylight Savings Time ends. Therefore, the timestamp for the sensor in Central Time Zone ended up being overwritten for the time range November 6th, 2:00 AM through November 6th, 2:59 AM when the clock went back one hour. Therefore, I had to discard this portion of the data due to missing data that was overwritten. In addition, there was also some delay in the Apple Watch's accelerometer and gyroscope sensor, and this varied between participants.

Therefore, when I was extracting data, I had to consider the time difference and delay. I analyzed this by looking at the three devices' data and aligning according to the user's reported data of meals and exercise moments. For example, if the user reported that they were sleeping at a certain time, I should be able to see a flatline of zero for all axis of the accelerometer reading since the sensor would not be moving for a significant amount of time or that glucose levels should increase after eating. See Figure 2.3 below. Another indicator would be the increase in glucose levels because of when someone had eaten. Using the user log timestamps in addition to the sensor time stamps, this allows me to align the times properly.

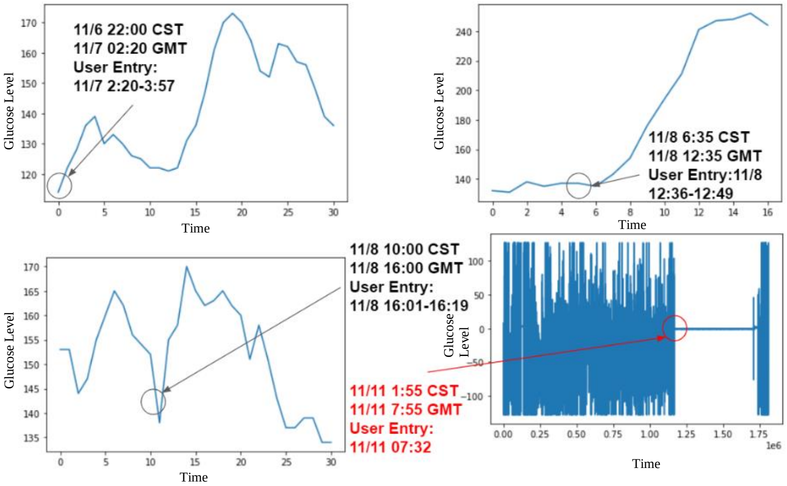


Figure 2.3

The glucose data timestamps were in CST time and the user logs and E4 Accelerometer was in GMT. The graphs in the top row and left column represent the glucose levels after the participant reports eating. The bottom left graph shows a comparison when the user logs their sleep time and the actual sleep time.

Additionally, a bigger challenge is that the user logs are only relative. For example, they may say they were sleeping for eight hours and fifteen minutes when they were sleeping for eight hours and fifty minutes. Thus, makes aligning the timestamps a difficult task and forces me to use my best judgment. Consequently, this may not result in accurate predictions or readings. Therefore, I decided against analyzing participant two’s data since they had not logged any data about their activities, making it essentially impossible to detect their patterns.

2.5 Sliding Window and Measuring Statistics

I then decided that I would use a sliding window technique with possible overlap across the data all the time. In each window used for detection, I measure the standard deviation, mean, and range of each feature (glucose - Dexcom or Abbot, each axis of the accelerometer and gyroscope and the force, as well as body temperature, galvanic skin response, electrodermal activity, and photoplethysmographic data). See Figure 2.4. The standard deviation is used to see how average much of a change there was on average from timepoint to timepoint as eating will result in a change in specific features. The mean is also used to correlate any threshold of values to eating since eating normally results in different levels than non-eating. Finally, I also use range to measure variation but to also look for potential skewness as large ranges typically imply some activity such as eating, or exercise has occurred.

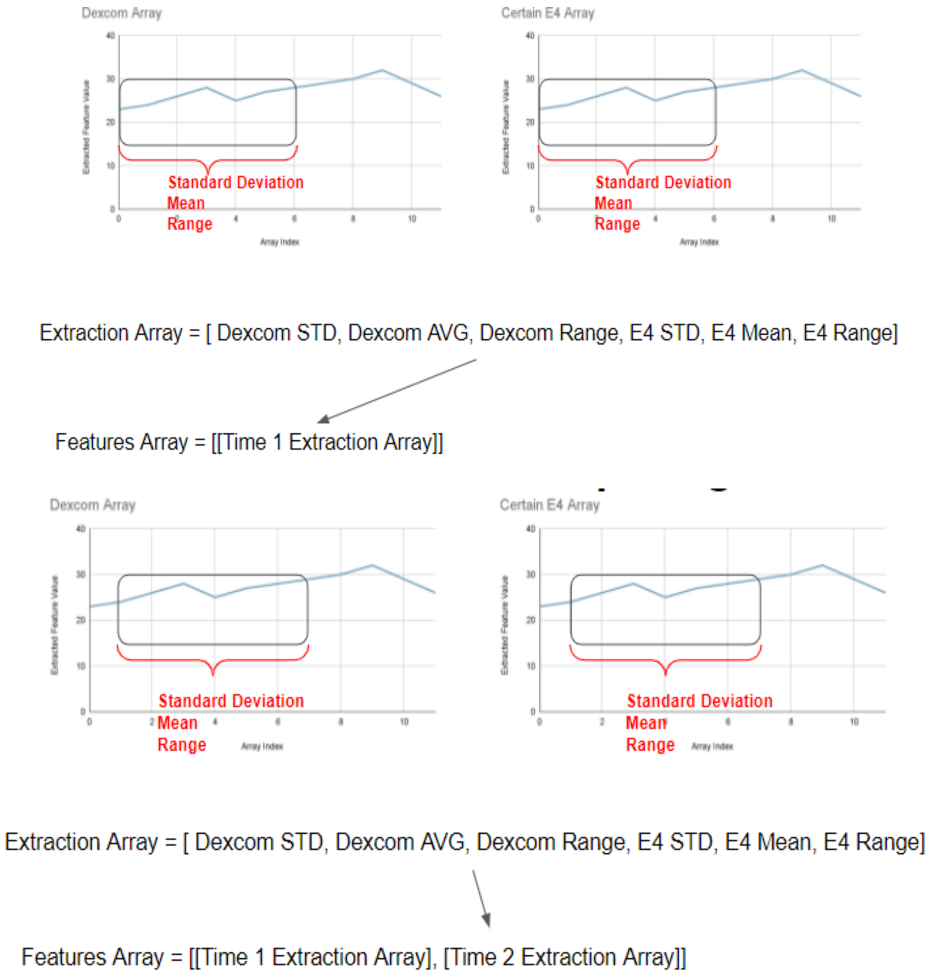
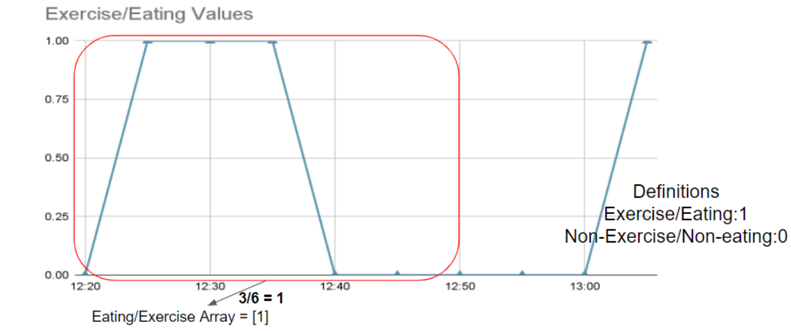

Dexcom Array
Certain E4 Array
Extracted Feature Value
Array Index
Standard Deviation
Mean
Range
Extraction Array = [Dexcom STD, Dexcom AVG, Dexcom Range, E4 STD, E4 Mean, E4 Range]
Features Array = [[Time 1 Extraction Array]]
Dexcom Array
Certain E4 Array
Standard Deviation
Mean
Range
Extraction Array = [Dexcom STD, Dexcom AVG, Dexcom Range, E4 STD, E4 Mean, E4 Range]
Features Array = [[Time 1 Extraction Array], [Time 2 Extraction Array]]

Figure 2.4

The sliding window loops over the data and extracts the features as shown above.

I define a window to be an eating/exercise window if the user is eating/exercising for any moment during the duration of that window, at least 5 minutes. The threshold is low because I need to be able to generate more training data. See Figure 2.5.

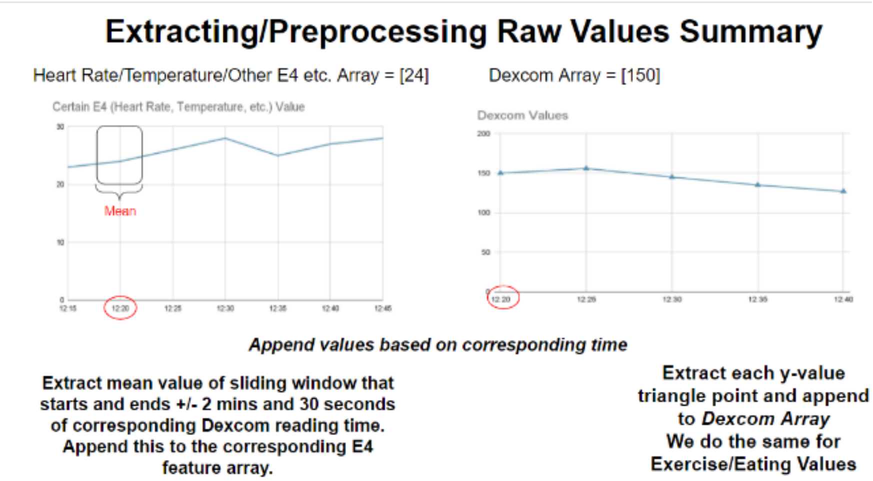


We use the same sliding window parameters (size and overlap) as with the features.
Store result in eating/exercise array.

Figure 2.5

I look at the span of window and see how long an activity occurred and determine if I am going to mark that window as if the activity occurred.

Many other measurements were not taken in five-minute intervals. Thus, they were all grouped into the nearest five-minute mark and averaged. See Figure 2.6. For example, all the motion data from November 6th 13:27:30 through 13:32:29 averaged and the time stamp at 13:30:00 for the motion features was marked as the average of all the data closest to that five-minute mark rather than using the exact motion data point at that timestamp. This is to get a better picture of what could be occurring around that time. Other limitations include that the window size cannot be less than 15 minutes since the Abbot monitor reads every 15 minutes. Thus, any window size less than 15 minutes may result in no data being able to be read.



Extracting/Preprocessing Raw Values Summary

Heart Rate/Temperature/Other E4 etc. Array = [24]

Dexcom Array = [150]

Append values based on corresponding time

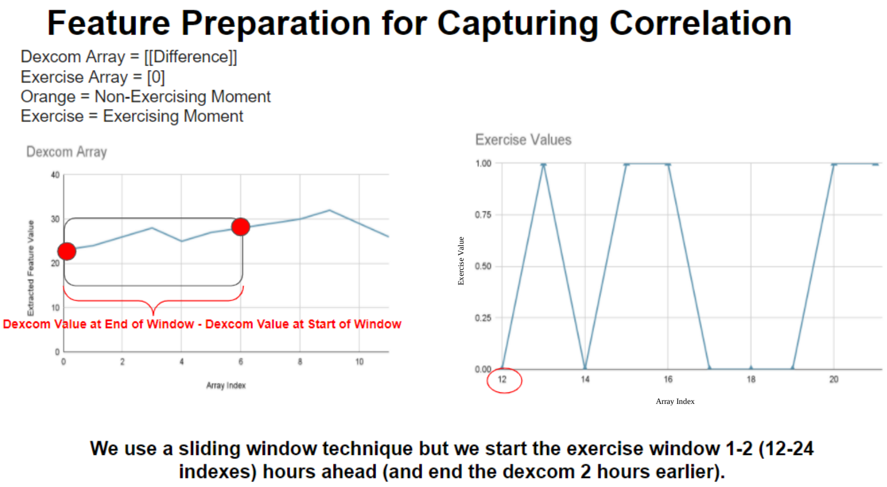
Extract mean value of sliding window that starts and ends +/- 2 mins and 30 seconds of corresponding Dexcom reading time. Append this to the corresponding E4 feature array.

Extract each y-value triangle point and append to *Dexcom Array*
We do the same for Exercise/Eating Values

Figure 2.6

Grouping feature values that read continuously in intervals shorter than five minutes.

For the windows used to detect correlation, I extract the starting value minus ending value of the glucose value for that window and correspond it to the time frame I am looking, whether that be 2 hours after or 1 hour after. See Figure 2.7. This is to be able to detect if there was an increase or decrease in glucose levels. The range would not be reliable since it is always positive and thus is unable to detect increase or decrease.



Feature Preparation for Capturing Correlation

Dexcom Array = [[Difference]]
Exercise Array = [0]
Orange = Non-Exercising Moment
Exercise = Exercising Moment

We use a sliding window technique but we start the exercise window 1-2 (12-24 indexes) hours ahead (and end the dexcom 2 hours earlier).

Figure 2.7

This figure explains how correlation feature and values are preprocessed for correlation between exercise and glucose levels.

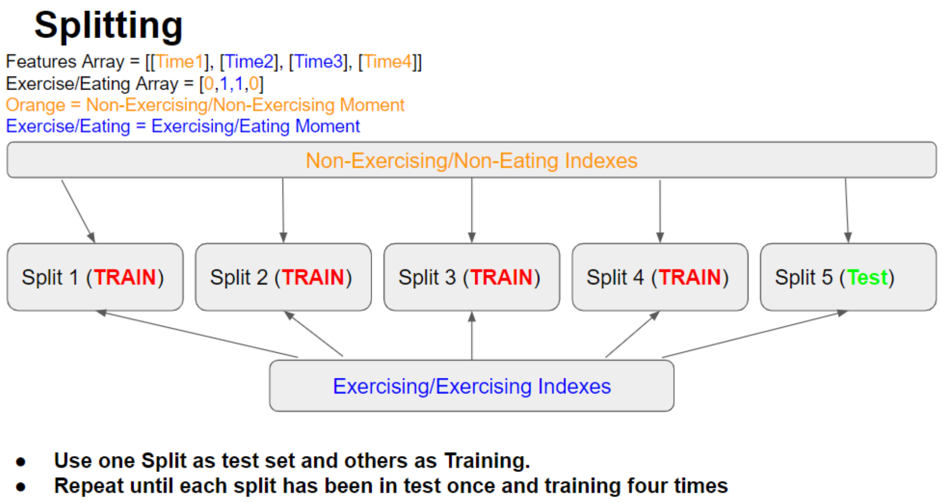
2.6 Data Pruning

One step I also had to take was that I needed to trim or not use some parts of the data. This is mainly because some parts of the data are missing in one dataset. For example, participant 3 had no reading from the Abbot sensor until November 5th and had major gaps until around 8:55 AM November 6th, in which I simply indexed off any readings before that, thus truncating the reading of that data. For the motion data, I manually edited the .csv file and removed any dates that went past the glucose monitor's reading. Participant 1 also had lapses in data. One occurred at the end in which I manually edited the .csv file and deleted rows with missing Abbot readings. The other occurred in the middle of the data and not wanting to delete a large amount of data, I wrote in the code to not consider any windows where data is missing.

As with the motion data, I also removed any data that went past the glucose reading monitors to ensure that the data matches up. When I begin to add other features (i.e., body temperature, heart rate, galvanic skin response, and photoplethysmographic data), I will have to remove even more data since the monitor is not always reading. Instead, it only reads for part of the day and not continuously like the glucose and motion monitors. In this case, removing data will be the only option as to not disturb the data for use in training models without these features. I also remove any data in which the window size may not have read a full window size of data, which typically happens at the end of the file. In other words, the window size may not divide evenly with the number of time points thus resulting in the last window missing a few time points and thus not necessarily creating an equal and balanced reading for standard deviation, mean, and range.

2.7 Training and Testing

To evaluate each model, I do a fivefold cross validation. To create each of the splits for cross validation, I first separate the eating labels and their corresponding features apart from the non-eating labels and their corresponding features. I then evenly split the eating and their corresponding features into the five folds. See Figure 2.8. The same is done again for the non-eating labels and their corresponding features. This is to ensure that when training that the model also has some underrepresented labels of non-eating.


Splitting
Features Array = [[Time1], [Time2], [Time3], [Time4]]
Exercise/Eating Array = [0,1,1,0]
Orange = Non-Exercising/Non-Exercising Moment
Exercise/Eating = Exercising/Eating Moment
Non-Exercising/Non-Eating Indexes
Split 1 (TRAIN)
Split 2 (TRAIN)
Split 3 (TRAIN)
Split 4 (TRAIN)
Split 5 (Test)
Exercising/Exercising Indexes

- **Use one Split as test set and others as Training.**
- **Repeat until each split has been in test once and training four times**

Figure 2.8

This figure illustrates how I evenly separate the labels first and the corresponding features so that each split contains the unrepresented label.

I then check by using the test set and use metrics such as the overall accuracy, balanced accuracy, F1 Score, ROC AUC score, and PR AUC Score. See Figure 2.9. The same procedure is repeated for exercise experiment, except I use the E4 accelerometer instead of any Apple Watch data as well as I only used Dexcom glucose readings.

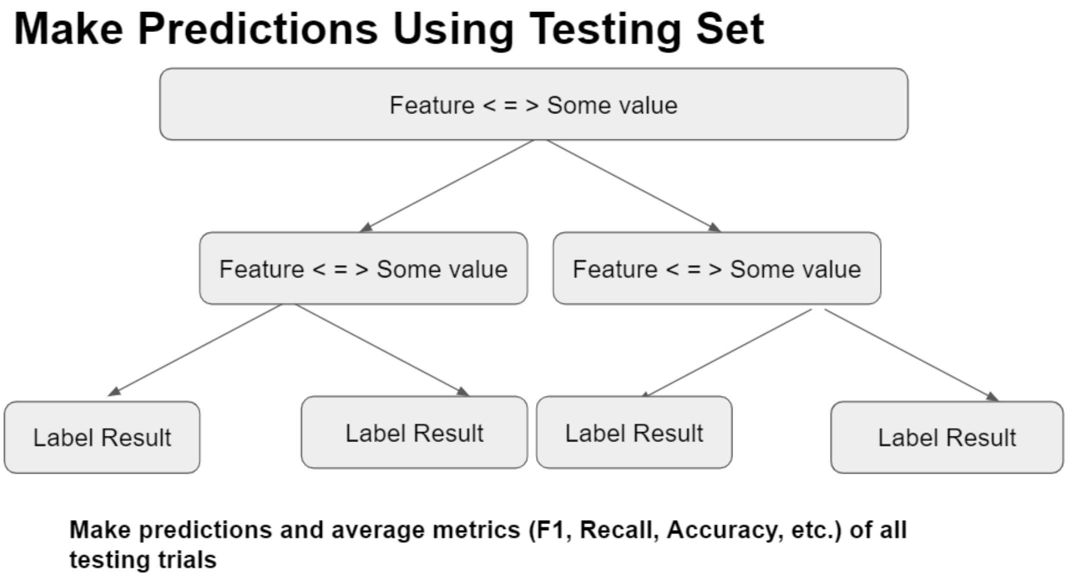

Make Predictions Using Testing Set
Feature < = > Some value
Feature < = > Some value
Feature < = > Some value
Label Result
Label Result
Label Result
Label Result
Make predictions and average metrics (F1, Recall, Accuracy, etc.) of all testing trials

Figure 2.9

This figure depicts how a decision tree is used. If the evaluation is true, it will go down one branch. Otherwise, it goes down the other.

2.8 Data Description

2.8.1 Data Collection

The data being used here is collected by participants recruited by the Systems and Technology for Medicine and IoT (STMI) Lab. For privacy reasons, I do not know which individual the data belongs to. Instead, they are labeled with just a number (i.e., participant 1). I used multiple participants in this dataset to train on, namely participant 1, participant 3, and participant 4. All of the data collected is then represented as a .csv file with the header representing the features. Additionally, all timestamps are indicated on the 24-hour clock.

2.8.2 Glucose Data

Glucose monitor data was collected over the course of a few days, from November 03rd through November 11th for participant 3 and participant 4 and from November 5th through November 9th for participant 1. All times indicated in this dataset are in Central Time. The Abbot only reads every fifteen minutes while Dexcom reads every five minutes. Both monitors were stuck in the arms of participants. Dexcom data typically ranged from 50-150 while Abbot data ranged typically from 80-250.

2.8.3 User Logged Eating and Exercise Moments

There is a self-reported data log that users input the time they started eating as well as the time they ended eating. I always rounded to the nearest 5-minute mark in the combined glucose data mark to determine if a subject was eating or not at that time. A big challenge was that many of the user's inputs didn't make sense. For example, participant one indicated that he or she was eating from November 7th, 2021 02:20 to November 8th, 2021 03:57, which is over 24 hours. I find this impossible and assumed it was a typo and that the user meant the end time was November 7th, 2021 03:57. Another example was in participant 3's activity logs in that the start time occurred ten hours after the finish time. The user logged the start time as November 7th, 2021 03:00 and the finish time as November 6th, 2021 17:00. I adjusted the finish time to one hour after the start time since that would make sense as to when someone would finish dancing and could be the result of a user inputting their logs incorrectly. In addition, I also determined that moments of actively walking during work or dancing is still counted as exercise. After all, actively walking or dancing during a workout vs. at work is going to be no different as the sensors are unable to distinguish the two moments. Eating/exercise is classified as 1 and non-eating/non-exercise is classified as 0. All timestamps in this dataset are in Greenwich Mean Time and was recorded in an apple watch application.

2.8.4 Accelerometer, Gyroscope, Roll, Pitch, and Yaw Data

Other data included accelerometer and gyroscope data in the Apple Watch. This data was collected about every millisecond and was continuous with no breaks except for those where the reading failed due to failure in the sensor. Data here was collected from the apple watch sensors, which have an accelerometer and gyroscope and are typically in the single digits with an absolute value of no more than five. Accelerometer data was also collected in E4 watch during the same time that watch collected other data. Furthermore, timestamps collected in this dataset were complicated as there was sometimes where there were delays in readings.

2.8.5 Body Temperature, Heart Rate, Galvanic Skin Response, and Photoplethysmographic Data

In addition to apple watch sensor data, I also collected data about the subject’s Body Temperature, Heart Rate, Galvanic Skin Response, and Photoplethysmographic Data from an E4 Watch. Each day that data was collected is represented in a separate file. Heart rate was collected once every second while temperature and electrodermal (Galvanic Skin Response) data were collected four times every second. Photoplethysmographic Data was collected 64 times per second. All timestamps for each of the features from the E4 watch are labeled in Greenwich Mean Time.

Given the number of timestamps these features produced, the Photoplethysmographic was too large to be viewed in the Excel application and even Google Spreadsheets. Furthermore, these sensors never ran the entire day. In other words, it did not run 24/7 while this experiment was going. It would usually start sometime in the morning hours and end in the evening or early morning hours of the next day. Photoplethysmographic data typically ranges from a few hundred in the negative values to a few hundred in the positive range. The galvanic skin response data ranges from 0 through about 5 but stays in the single digits usually. The heart rate is typically from the 50s to a couple hundred and the temperature ranges from teens to the 30s.

3. RESULTS

3.1 Eating Detection

My attempt to detect eating produced inconsistent results. In addition, statistical metrics were not above 0.5. For example, my F1 scores were typically never above 0.5. However, I did experiment with combining more features, adjusting the window size, and changing the overlap to look for any significant changes.

With combining more features, one major thing I noticed is that I noticed similar trends when comparing Abbot vs. Dexcom monitors in participant one and participant four of my statistical measurements aimed at detecting eating moments (i.e., F1 Score, Balanced Accuracy, ROC AUC Score, and PR AUC Score) when I add more features. See Figure 3.1 below depicting changes in the F1 Score.

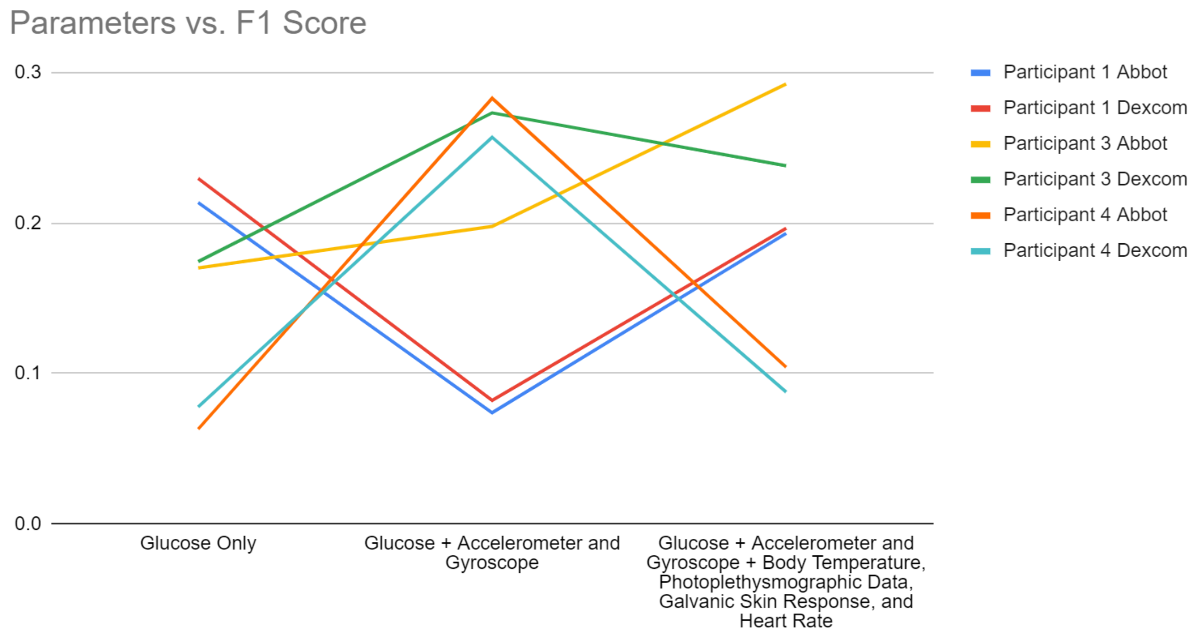


Figure 3.1

This graph depicts the performance as I add more features.

The key reason this might occur is because certain features such as accelerometer and gyroscope, might play a bigger role as a feature in detecting eating in certain participants depending on how accurately the participant records their timestamps. Certain foods might have less of an effect on glucose depending on the participant. Participant one’s log timestamps were not very aligned with the Apple watch’s accelerometer and gyroscope and thus has lowered predictive probability but went up due to E4 data potentially having a stronger correlation. Participant four’s log timestamps for were more aligned with the watch’s data and thus allowed me to have higher predictability. I also noticed that there is a decrease in the metric when using the E4’s data on participant four. This is likely because there is less correlation between E4’s data and eating and thus has less predictive power. Participant three’s trend was not entirely consistent. This could be attributed to many reasons such as missing data or misaligned data.

Another major factor that I examined was the window size, which varied in performance depending on the participant and whether Dexcom or Abbot was used. See Figure 3.2. Unlike with adding more features, the results here proved to be much more inconsistent. My intent was to try to generate more eating moments and examine a wider range of feature measurements that correlate to just even short moments of eating. For some, this may slightly increase the predictability power and then decrease.

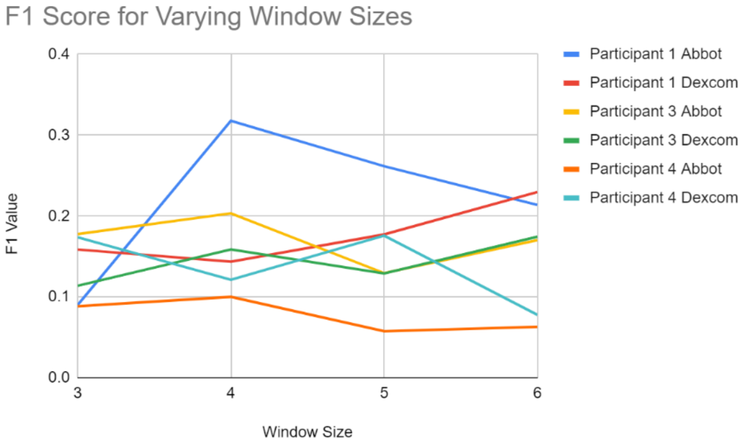


Figure 3.2

This graph depicts the performance of varying window size.

All models above used glucose with an extreme gradient boost classification tree with SMOTE to increase the underrepresented classes. This SMOTE technique attempts to allow the classifier to better fit the model since it now has an equal pool of classes with the underrepresented class being able to be represented more and have its distinct features, that would be different from non-eating and non-exercising moments, more synthetically represented. However, because of features that lack correlation with eating, their performance was still not as good.

Another change that did not make much of a difference is the overlap in comparing CGM with eating detection. There does not appear to be any significant variation resulting from changes in the window size. This is because overlap simply allows data points (except points at the beginning and end) to be represented multiple times, thus, resulting in more similar statistics being produced.

3.2 Exercise Detection

With exercise detection being compared to glucose, I was able to produce better results compared to eating detection. I used the same method of using an Extreme Gradient Boosted

Decision Tree, applying the SMOTE on the training set, and evaluating on the fivefold test split as I had previously utilized in my eating detection experiment. I used a fixed window size of 30 minutes and overlap of 25 minutes. I tend to see higher scores in F1, Area Under the Precision-Recall Curve, and Area Under the Receiver Operating Characteristics (AUC ROC) compared to that of eating detection. The key reason is that there are likely certain features (such as higher heart rate, electrodermal activity, lowered glucose levels, etc.) tend to be more correlated with the exercise allowing the model to look for these patterns and then using them as predictors in determining if exercise occurred.

Below are the graphs of the five trials for each of the participants I used. I compared using E4 sensor data with Dexcom data and without and found that they both produce similar results. This is likely since there are similar variations in a participant's glucose levels throughout the day. See Figure 3.3 A-C below.

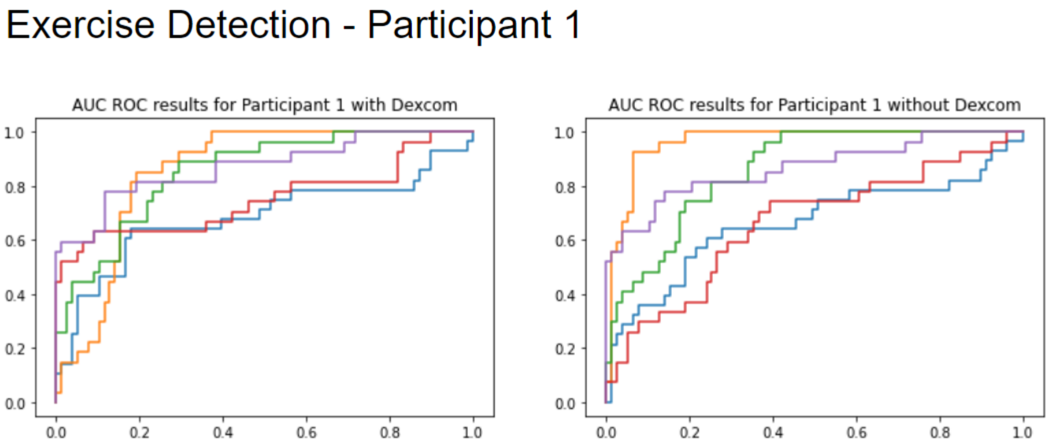


Figure 3.3 A

The above depicts the curve of AUC ROC for participant 1 with vs. without Dexcom, which is the false positive rate vs. the true positive rate.

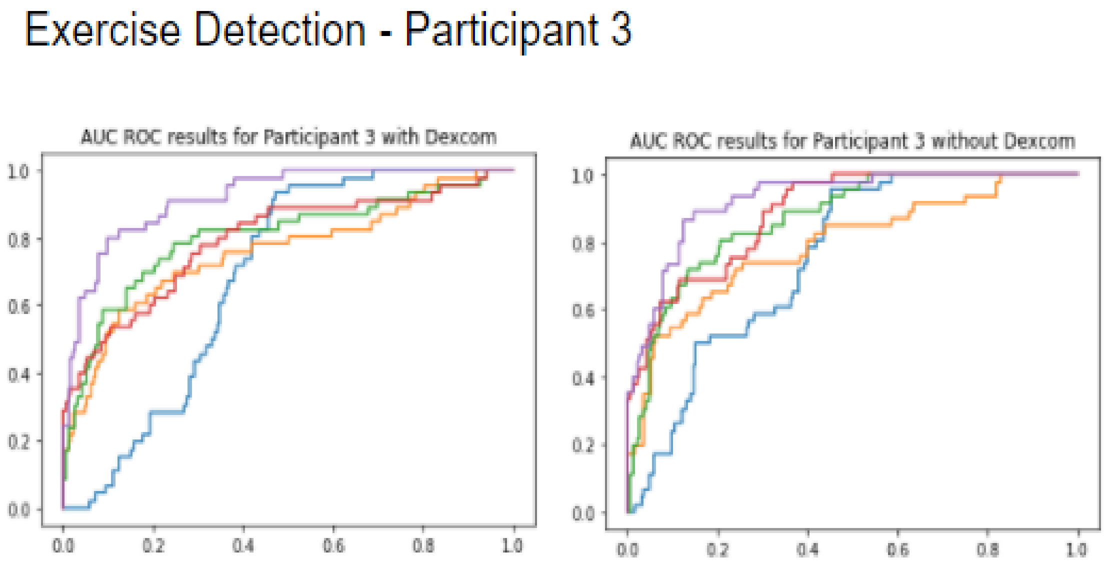


Figure 3.3 B

The above depicts the curve of AUC ROC for participant 3 with vs. without Dexcom, which is the false positive rate vs. the true positive rate.

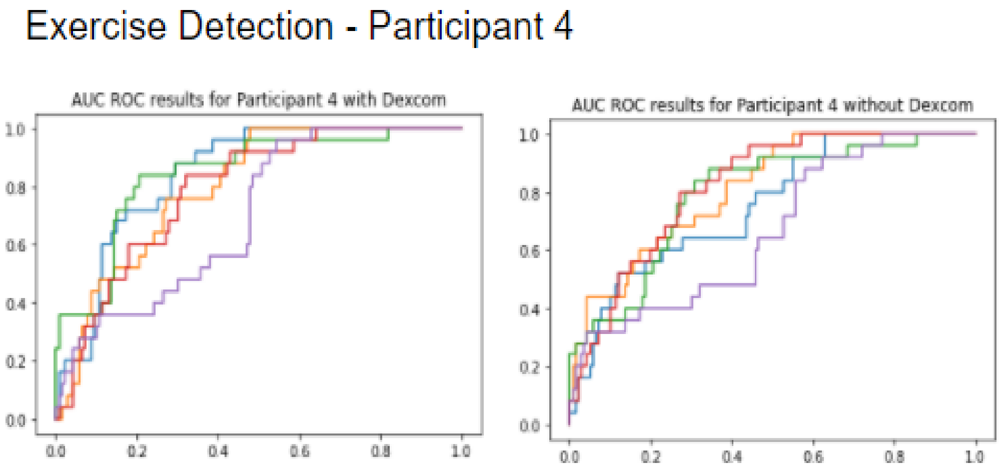


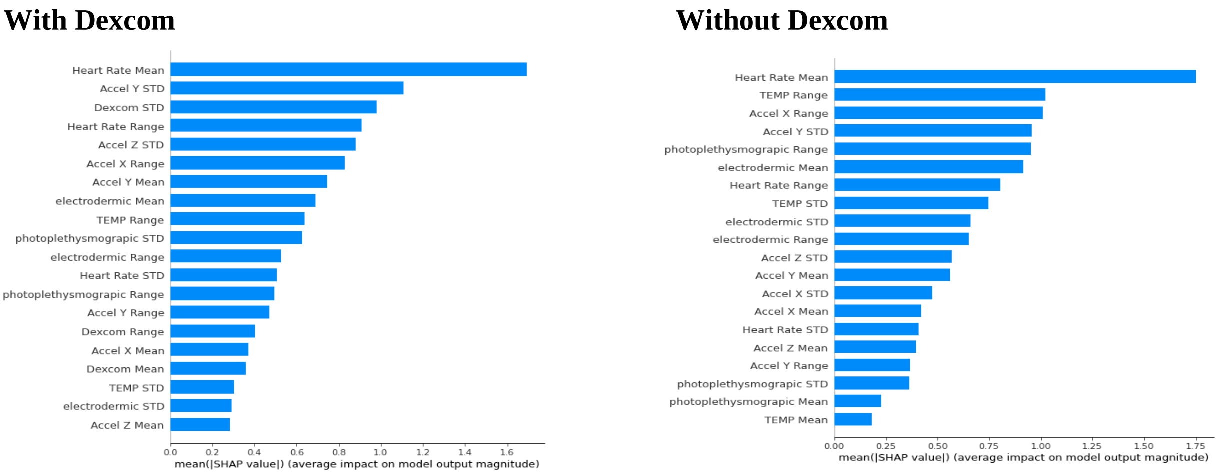
Figure 3.3 C

The above depicts the curve of AUC ROC for participant 4 with vs. without Dexcom, which is the false positive rate vs. the true positive rate.

Ideally, the higher the value curve is leaning towards the top left corner, the more robust the model performs. This means that as the true positive rate increases, the false positive rate should not increase as much. As you can see, the model does have some capability to detect exercise, but it still misclassifies many non-exercise moments as exercise. This is likely since, just like with eating detection, there are features that further distinguish exercise from non-exercise but there are still some that are similar in both exercise and non-exercise moments. For example, heart rate might increase when someone is stressed or anxious.

In terms of feature importance in comparing with glucose, this varies from participant to participant. I observe that glucose is only a minor factor and that only a couple of features will significantly make an impact on detecting exercise and that there is often an outlier in which the feature with the highest impact will often have a much bigger impact than others. This is likely because certain activities that a participant does will result in certain features being more applicable to detecting exercise. However, as a major challenge, I am simply detecting exercise in general rather than what specific type of exercise they were doing (e.g., running, walking, weightlifting, etc.). I use the SHapley Additive exPlanations (SHAP) values to examine any impact on exercise detection. Higher SHAP values means higher impact on the prediction. See Figure 3.4 A-C below.

SHAP Values for Participant 1



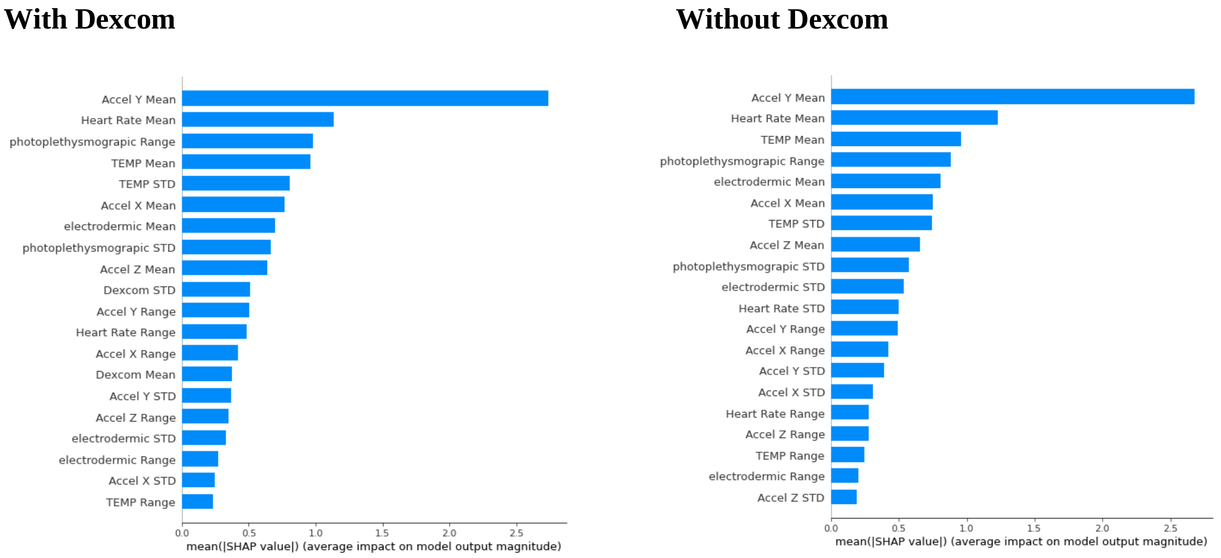
With Dexcom

Without Dexcom

Figure 3.4 A

The above two graphs compare the SHAP values for participant 1 with vs without Dexcom monitor.

SHAP Values for Participant 3



With Dexcom

Without Dexcom

Figure 3.4 B

The above two graphs compare the SHAP values for participant 3 with vs without Dexcom monitor.

SHAP Values for Participant 4

With Dexcom

Without Dexcom

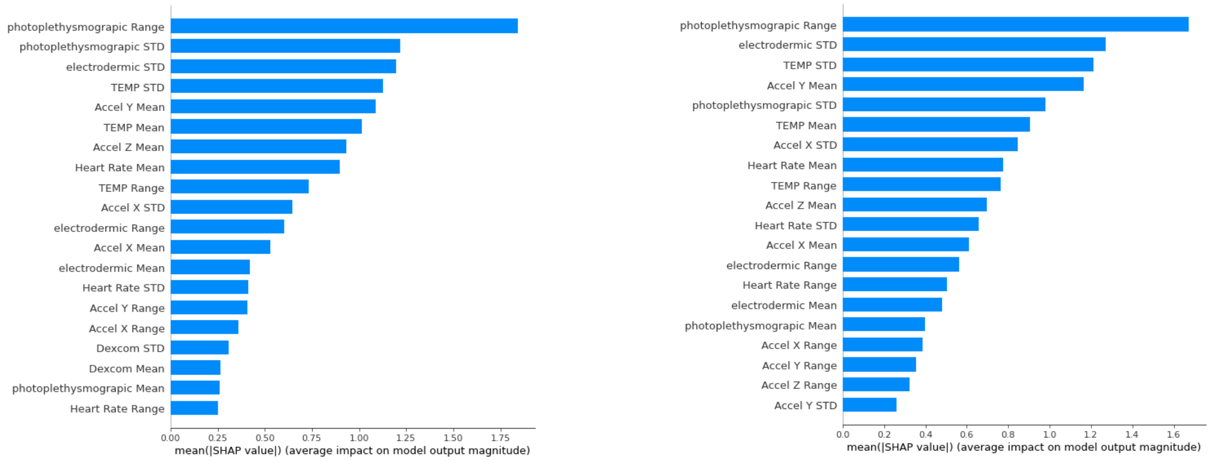


Figure 3.4 C

The above two graphs compare the SHAP values for participant 4 with vs without Dexcom monitor.

3.3 Exercise Correlation with Glucose Levels

Another thing I am trying to examine is for any correlation between the exercise and glucose level after a certain timeframe. I namely looked at glucose variation one hour after exercise and glucose variation two hours after exercise. I found that there is a stronger correlation with glucose variation one hour after exercise as opposed to two hours after exercise. See Figure 3.5.

Glucose Level Correlation with Exercise After a Certain Time

One Hour After Exercise

Participant 1	**Participant 3**	**Participant 4**
-0.12	**-0.028**	**0.087**

Two Hours After Exercise

Participant 1	**Participant 3**	**Participant 4**
-0.09	**12**	**0.03**

Figure 3.5

Here is a depiction of correlation numbers for my three participants for one hour after exercise and two hours after exercise.

It is possible that the decrease in correlation is probably because humans' glucose variations tend to be more similar to that when exercise occurs and when it does not occur two hours after exercise. Another interesting result is that some participant one had positive correlation rather than negative correlation. It is likely that participant one just has a body that functions differently and therefore continues to have a decrease in glucose levels rather than an increase. The same can be said for participant three for one hour after exercise although participant three had a positive correlation two hours. Therefore, it is likely that participant three experienced an increase in the blood glucose levels sometime between one to two hours after exercise.

4. CONCLUSION

These models purely examine eating versus non-eating moments and exercise vs. non-exercise moments to compare if glucose is correlated with those activities and if other body statistics have a bigger impact. These models further assumes that every feature like that of eating or exercise activity is truly eating or exercise. It is difficult to tell that it could possibly be another activity that creates similar results to eating or exercise thus resulting in misclassification. It could be possible that having a model train on more features, such as sound from a microphone sensor, better distinguish eating from smoking and classify the two accordingly. Additionally, future improvements could involve collecting more participants and gathering their health data (i.e., body weight, if they have diabetes, do they smoke, are there any anxiety disorders, etc.). Furthermore, this is also assuming all foods and exercise moments have the same effect on the body though it is possible certain foods can affect glucose levels differently on each different person and each person might perform different workout motions of different intensity.

After adding features and experimenting with various models and techniques, I have observed that more incorrect results are produced in detecting eating motions with CGMs or with any of the other body metrics. I likely believe that this is because eating only has a minor effect on the glucose levels of these participants and minimal effect on the other body metrics that I used. In addition, the effects of eating vs. non-eating on the body might not be correlated as much with the other body metrics, such as heart rate, electrodermal activity thus making those features less reliable.

Like with eating detection, glucose was not the most significant factor in exercise detector. Instead, E4 data typically played a more significant role. Unlike eating detection, I was able to fare better than exercise detection as I typically saw better performance compared to eating detection. This is likely because there are higher variations electrodermal activity, accelerometer, and heart rate and thus is subject to noise when detecting attempting to detect exercise but can be improved upon. It normally misclassifies at least 30% of exercise moments as non-exercise but still means that it is more correct than not. Exercise can come in many different forms and intensity thus resulting in different effects on the body, which in turn affects the sensor readings. Thus, in order to improve upon exercise detection, I would need to be able to classify exercise activities into what kind of exercise it is. For example, detecting walking is much different than detecting weightlifting but both can be classified as exercise.

In addition to trying to detect exercise, I also examined for any correlations for glucose levels and exercise. I found that there was little correlation between glucose and exercise. This is likely due to the fluctuations in glucose levels and my small window size unable to detect if the fluctuation is significantly caused by exercise.

These models do perform better than average but more sensors may need to be combined to better separate eating from non-eating and exercise from non-exercise. In addition, this was just purely based on a small number of subjects rather than the population as well as the fact that the timestamps were misrepresented by the users and that they combined certain activities (e.g., one participant logged that he or she was exercising and driving during a specific period), which made the model harder to distinguish between the two classifications.

REFERENCES

[1] Alshurafa, Nabil, et al. SwallowNet: Recurrent Neural Network Detects and Characterizes Eating Patterns, Mar. 2017.

[2] Bell, Brooke M., et al. Review of Automatic, wearable-based, in-field eating detection approaches for public health research: a scoping review, 2020.

[3] "Diabetes and Exercise: When to Monitor Your Blood Sugar." *Mayo Clinic*, Mayo Foundation for Medical Education and Research, 20 Jan. 2022, https://www.mayoclinic.org/diseases-conditions/diabetes/in-depth/diabetes-and-exercise/art-20045697.

[4] Doyle, Francis J. "Detection of a Meal Using Continuous Glucose Monitoring." American Diabetes Association, 2008, https://care.diabetesjournals.org/content/31/2/295.long.

[5] Farooq, Muhammad, and Edward Sazonov. "Accelerometer-Based Detection of Food Intake in Free-Living Individuals." IEEE Xplore,2018, https://ieeexplore.ieee.org/document/8309402.

[6] "Healthy Eating for a Healthy Weight."Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 19 Apr. 2021, https://www.cdc.gov/healthyweight/healthy eating/index.html.

[7] Huo, Zepeng, et al. AISTATS, Uncertainty Quantification for Deep Context-Aware Mobile Activity Recognition and Unknown Context Discovery 3 Mar. 2020.

[8] Hussain, Afzaal, et al. "Sensor-Based Gym Physical Exercise Recognition: Data Acquisition and Experiments." *Sensors*, vol. 22, no. 7, 24 Mar. 2022, p. 2489., https://doi.org/10.3390/s22072489.

[9] Johnson, Scott. "Your Diabetes and High Blood Sugar after Exercise." *MySugr*, 26 June 2015, https://www.mysugr.com/en-us/blog/high-blood-sugar-after-exercise/.

[10] Kalantarian, Haik, et al. "ResearchGate." Monitoring Eating Habits Using a Piezoelectric Sensor-Based Necklace

https://www.researchgate.net/Publication/270595886 Monitoring Eating Habits using a Piezoelectric Sensor Based Necklace, 2015,

[11] Khurana, Rushil, et al. "Gymcam." *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 4, Dec. 2018, pp. 1–17., https://doi.org/10.1145/3287063.

[12] Means, Casey, et al. "How a CGM Can Help You Find Your Optimal Diet and Lower Blood Sugar." *Levels*, 6 Dec. 2021, https://www.levelshealth.com/blog/optimal-diet.

[13] Morris, Dan, et al. "Recofit." *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2014, https://doi.org/10.1145/2556288.2557116.

[14] Morshed, Mehrab Bin, et al. vol. 8, ser. 12, JMIR Publications, A RealTime Eating Detection System for Capturing Eating Moments and Triggering Ecological Momentary Assessments to Obtain Further Context: System Development and Validation Study.

[15] Oerum, Christel. "Why Some Types of Exercise Can Make Your Blood Sugar Increase." *Diabetes Strong*, 19 Mar. 2020, https://diabetesstrong.com/why-some-types-of-exercise-can-make-your-blood-sugar-increase/.

[16] Papapanagiotou, Vasileios, et al. "A Novel Chewing Detection System Based on PPG, Audio, and Accelerometry." IEEE Xplore, 4 Nov. 2016, https://ieeexplore.ieee.org/document/7736096.

[17] Samadi, Sediqeh, et al. "Meal Detection and Carbohydrate Estimation Using Continuous Glucose Sensor Data." IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 3, 2017, pp. 619–627., https://doi.org/10.1109/jbhi.2017.2677953.

[18] Centers for Disease Control and Prevention, Centers for Disease Control and Prevention, 10 Aug. 2021, https://www.cdc.gov/diabetes/library/features/smoking-anddiabetes.html#:˜:text=Nicotine %20increases%20your%20blood%20sugar,close%20to%20their%20target%20levels.

[19] Stankoski, Simon, et al. "Smartwatch-Based Eating Detection: Data Selection for Machine Learning from Imbalanced Data with Imperfect Labels." Sensors, vol. 21, no. 5, 2021, p. 1902., https://doi.org/10.3390/s21051902.

[20] Thomaz, Edison, et al. National Center for Biotechnology Information, Atlanta, Georgia, 2015, pp. 1–32, A Practical Approach for Recognizing Eating Moments with Wrist-Mounted Inertial Sensing.

[21] “Updated Guidelines Urge Americans to Make Every Bite Count.” Mayo Clinic, Mayo Foundation for Medical Education and Research, 6 Feb. 2021, https://www.mayoclinic.org/healthy-lifestyle/nutrition-andhealthy-eating/in-depth/dietary-guidelines/art-20045584.

[22] Van den Boer, Janet, et al. “The Splendid Eating Detection Sensor: Development and Feasibility Study.” JMIR MHealth and UHealth, vol. 6, no. 9, 2018, https://doi.org/10.2196/mhealth.9781.

[23] “Why Does Exercise Sometimes Raise Blood Glucose (Blood Sugar)?” *Exercise Can Raise Blood Glucose (Blood Sugar) | ADA*, Accessed 2022, https://www.diabetes.org/healthy-living/fitness/why-does-exercise-sometimes-raise-bloodsugar#:~:text=Adrenaline%20Can%20Raise%20Blood%20Glucose,usually%20come%20down%20during%20exercise.