AMD: ANALYSIS OF MOOD DYSREGULATION

A Machine Learning Approach

A Thesis

Presented to

The Faculty of the Graduate School

At the University of Missouri-Columbia

In Partial Fulfillment

Of the Requirements for the Degree

Master of Science

Implemented and Defended By:

NICKOLAS M. WERGELES

Professor Yi Shang, Thesis Advisor

December 2016

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

AMD: ANALYSIS OF MOOD DYSREGULATION –

A MACHINE LEARNING APPROACH

Presented by Nickolas M. Wergeles,

A candidate for the degree of Master of Science in Computer Science,

And hereby certify that, in their opinion, it is worthy of acceptance.

Professor Yi Shang

Professor Dong Xu

Professor Timothy Trull

ACKNOWLEDGEMENTS

My most sincere appreciation goes to Professor Shang Yi for everything he has done for me. Without Professor Shang, I would not be able to get my master's degree. He has supplied me with all the equipment, supplies, advice, guidance, encouragement, friendship, and support anyone could ask for. Whenever I needed something, Professor Shang would be there to help me even when others were nowhere to be found. It brings me great pleasure to work with Professor Shang and for him to be my advisor. I would strongly recommend anyone to choose him as an advisor in the future and I hope to continue our friendship.

Next comes the members of the lab. As I am writing this I start to smile when thinking of all the good times we had and the help I received from the members of the lab. Special thanks to Zhang Chen, Sun Peng, Peng Zeshan, Yao Xinjian, and Shi Ruiqi. These great individuals were always there for me and I take great pleasure in sharing knowledge between these brilliant people. All of these colleagues come highly recommended by me and I am very lucky to get to know these individuals as well as all the other members of the lab.

Last but not least, I would like to express my deepest gratitude towards my family, Michael and Barbara Wergeles, Melissa Wergeles, Dixie Semelka, Vladimir Wergeles, Tebby Wergeles, Alex Wergeles, and Frank Semelka. If it were not for my family, I would not even be in computer science, studying my master's degree, or be the person I am today. My Uncle Alex influenced me to study computers, I received my engineering abilities and people skills from my father's side, and my mathematical abilities and book smarts come from my mother's side. My family's advice has always been the best guidance in my life and with my entire lifespan, I could not repay them for all the things they have done for me. Family is the most important thing in this world. Friends will come and go but family will be forever, and without family we have nothing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

There is a popular saying, “Stress kills.” This statement can be true with repeated exposures to psychological mood dysregulation, which can lead to or worsen stress related conditions such as heart disease and cancer. Therefore, mobile ambulatory assessment systems are actively being developed for various psychological studies. However, to the best of our knowledge, there are very few being used for the detection of mood dysregulation with a continuous measurement collected in the natural environment. This research presents a new automatic machine learning pipeline, called Analysis of Mood Dysregulation (AMD), which is used to assess mood or emotional dysregulation caused by underlying psychological disorders, environmental factors, and daily activities. The data is collected by unobtrusive wearable sensors, without pre-calibration, worn by subjects in their natural environments.

In this research, we propose, build, train, and test multiple machine learning models for continuous prediction on rapidly varying and sporadic mood data. As a result, each model will predict whether changes in the physiological data represent mood dysregulation for each subject in the study. All models were trained using two weeks of physiological and self-initiated mood data collected from 22 subjects during their everyday lives. We found that creating time categories for each day improves the accuracy of AMD by more than 10%. Additionally, mood dysregulation during the afternoon and evening categories happens 60.54% more than the morning and night categories. We then analyze the relationship between time, physiological measurements, and mood dysregulation to develop a model that can predict mood dysregulation episodes with 93.31% accuracy. Moreover, to train and test the model yields an average total execution time of 35.66 seconds consisting of approximately 386,000 records for each user.

1. INTRODUCTION

There is a popular saying, “Stress kills.” This statement can be true with repeated exposures to stress and mood dysregulation, which can lead to or worsen conditions such as heart disease and cancer [1]. However, stress can also be a positive force in everyday life, but only in moderation [2]. Stress can enhance certain actions, improve performance, and increase excitement when people are in danger, during an exam, etc. [2] [3] [4].

Excessive, chronic, and repeated exposures to stress can lead to significant negative health consequences [2] [5] [1]. For short-term side effects, excessive stress can cause headaches, trouble sleeping, and fatigue [2] [6] [7] [8]. For long term side effects, excessive stress can be associated with risk for several chronic diseases including cardiovascular diseases and cancer [2] [9] [10]. Animal and human studies have shown that stress can also play a role in psychological or behavioral problems, such as depression, addiction, rage, anxiety, and other mood dysregulation issues [2] [11] [12] [13] [14].

So what is mood dysregulation? Carpenter in 2013 describes emotion and mood dysregulation as the inability to flexibly respond to and manage emotions [15]. He continues to say, although this definition may appear straightforward, there is considerable variation in the phenomena studied under the heading of emotion dysregulation in borderline personality disorder (BPD). According to Linehan’s biosocial theory, individuals with BPD are emotionally sensitive from birth [15].

This sensitivity leads to a propensity to experience negative affect across contexts and situations, which then makes it difficult to learn appropriate emotion regulation strategies [15]. This deficit in appropriate regulation strategies likely contributes to a tendency to engage in dysregulated behaviors in order to manage and reduce negative affect. This four component

process results in negative consequences, which, in turn, reinforce emotion sensitivity. The result is a recursive pattern of emotion dysregulation as show in Figure 1.

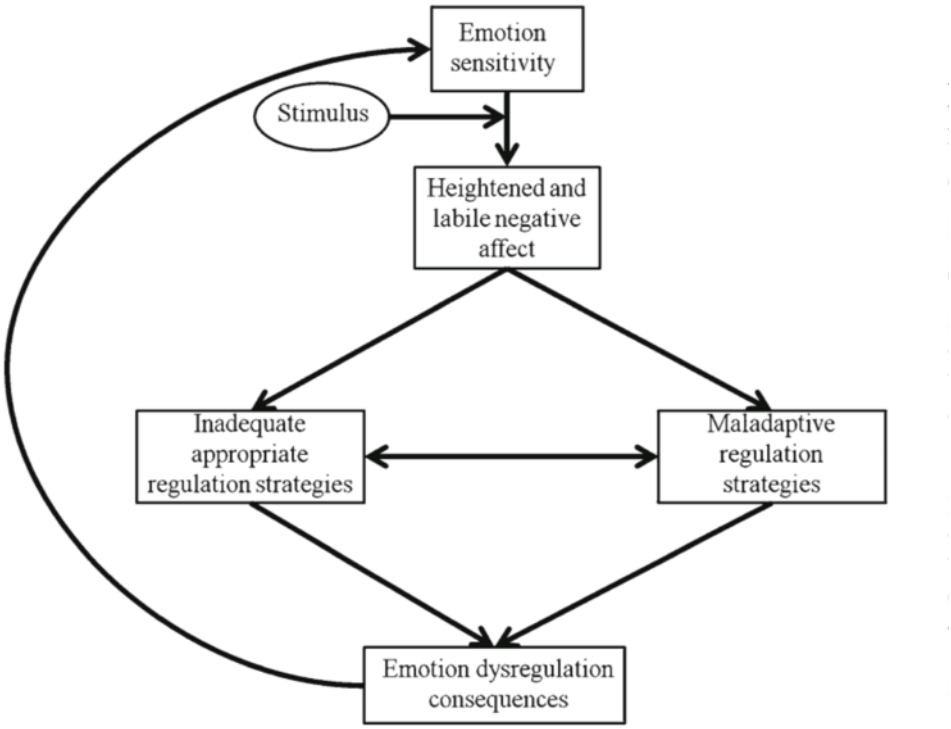

Figure 1. Multi-component model of emotion dysregulation in borderline personality disorder (BPD) [15]. Individuals with BPD are theorized to be sensitive to emotional stimuli from birth. Interpreting a stimulus in a negative way in the environment leads to increases in negative affect and instability. Heightened and unstable negative affect both makes it difficult to learn and to employ appropriate emotion regulation strategies. This leads to an increase in maladaptive and impulsive regulation strategies. Emotion dysregulation consequences occur as a result, which, in turn, reinforce emotion sensitivity [15], this in turn is a cycle.

1.1 Mobile Development Problem

Currently, most methods in clinical psychological research primarily rely on questionnaires and interviews with examiners in the lab setting [16]. Figure 2 depicts this process of relying on questionnaires and interviews with examiners in the lab setting, including a real-life participant in our research project. In behavioral science, periodic self-reports are commonly used to measure perceived stress in natural environments [2]. Self-reports allow the collection of measurements

of perceived stress from the view of the subject, often administered multiple times per day to reach a desired sampling of stress. Nevertheless, periodic self-reports collect only subjective aspects, which often miss true mood dysregulation episodes, and can impose a significant burden on subjects depending on how many self-reports they answer each day. The main challenge in addressing the epidemic of mood dysregulation and stress is the lack of robust methods to measure a person's exposure to stress-producing activities in the natural environment [2].

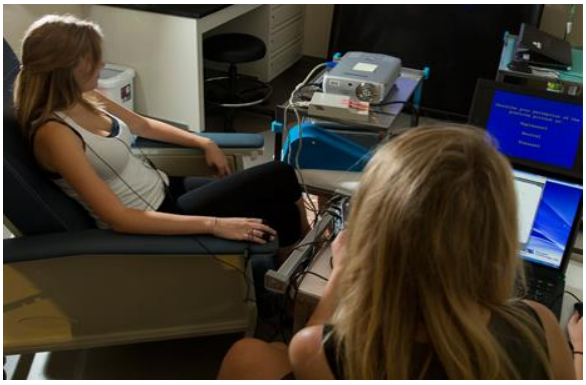

Figure 2. Real-life participants in the lab setting completing questionnaires and interviews. The right photo is an example in our research project. This was the primary method of obtaining periodic self-reports before the release of our mobile ambulatory assessment system (mAAS).

1.2 Mobile Development Solution

Therefore, to ensure capture of accurate, real-time data and mood dysregulation episodes, a continuous measure of mood dysregulation is needed. With the rapid development of mobile technologies, a promising new solution is a mobile ambulatory assessment system with real-time data monitoring and collection of real-life subject behavioral, psychological, and physiological data [16]. Figure 3 shows an example of using a mobile ambulatory assessment system on a smartphone and is the system currently being used today in our research with approximately 40 real-life subjects.

Ambulatory assessment comprises the use of field methods to evaluate the ongoing behavior, physiology, experience and environmental aspects of subjects in naturalistic or unconstrained settings [17]. Our mobile ambulatory assessment system collects information about the external environment as well as the participants' physiological and mental states. This information is collected through random system-generated and user-initiated self-report surveys and combined together. Then machine learning models can be developed to identify changes in mood, stress, as well as other psychological problems by using the combined data.

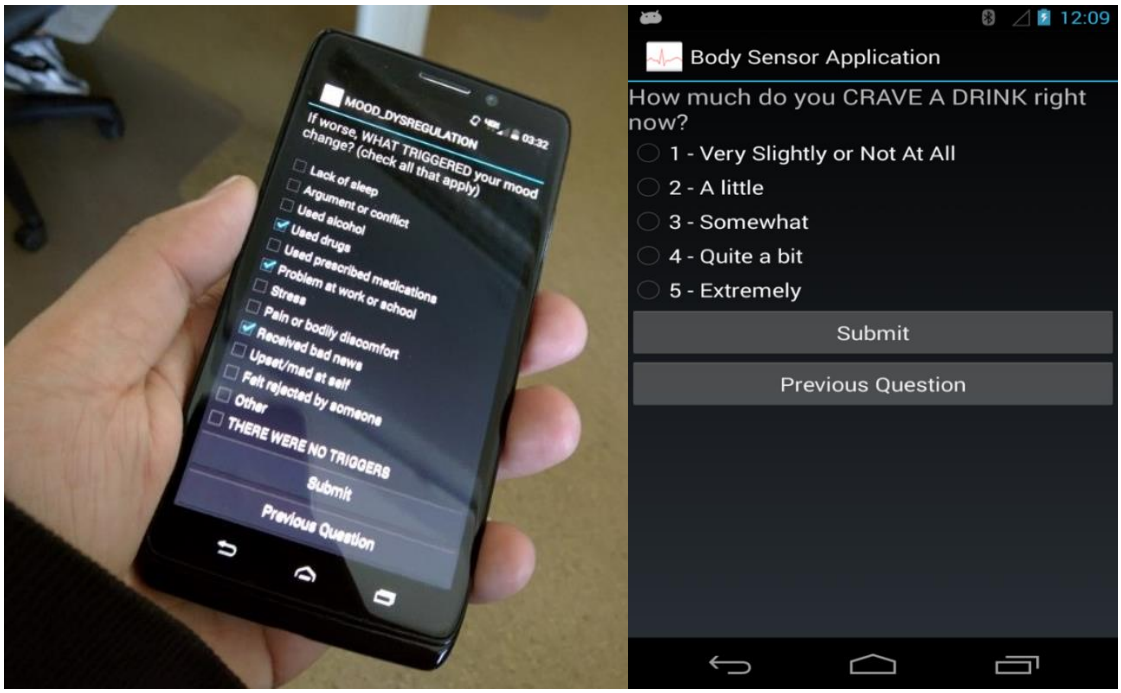

Figure 3. Example of using a mobile ambulatory assessment system on a smartphone and is the system currently being used today in our research with approximately 40 real-life subjects. Instead of subjects going into the lab several days after the mood dysregulation event to complete a self-report, the user can now administer and collect data in real-time, allowing for a more efficient and accurate way to collect data.

This same information can also be applied to context-aware applications. In context-aware computing, the context can be classified as any information that can be used to describe the state of something that is relevant to a user's interaction with an application [18] [19]. Combining methodology from psychophysiological field research with wireless body area sensor networks

and mobile devices can improve context-aware computing. To provide self-reports, a person must be willing to have their daily life interrupted, sometimes as many as 20 or 30 times per day. A high subject burden may lead to compliance issues and may affect the quality of self-reports and measures collected. Therefore, a passive approach requiring minimal attention from the subject would be a significant advancement for measuring mood dysregulation [2].

In Chapter III we will review our mobile ambulatory assessment system, mAAS [20]. mAAS was designed as a general sensor and survey data collection system during a subject's normal life and daily activities for clinical studies of mood dysregulation. Similar systems based on smartphones and wireless body sensors have been developed in a variety of literature domains such as activity classification and monitoring [21] [22], personal health monitoring [23], social networks [24] [25], safety and environmental monitoring [26], and transportation [27].

1.3 Machine Learning Problems

Measuring physiological measurements such as heart rate, breathing rate, activity, and other body metrics could be the first step for a passive approach and a continuous method to correctly classify mood dysregulation, stress, and other psychological problems. However, it is naïve to assume this will be the solution to all the problems because physiological sensor measurements present many other challenges not everyone would suspect. First, everyone is different, especially when considering their physiology. Some people's heart rate will be increasing during times of mood dysregulation, others have a decrease in heart rate. Therefore, wide between-person differences in physiological sensor measurements to mood dysregulation make it difficult to build a machine learning classifier which works on all subjects or even on a large number of subjects. Second, the sensors for measuring physiology must not be obtrusive. They must also be wearable, wireless, and battery powered which is capable of lasting all day. The data collected from these

sensors must be accurate by providing scientifically valid measurements while subjects are in their natural environments.

Third, in order to build and develop a machine learning classifier for physiological measurements, this requires collecting ground truth in a subject's natural environment. However, in the literature the most viable solution for collecting ground truth in a subject's natural environment are periodic self-reports [2]. Periodic self-reports limit the quality and quantity of ground truth with their discrete characteristics and subjectivity that can be collected in a subject's natural environment. Fourth, with all this being said, readers must be aware that everyday habits or events that occur naturally in a person's daily life, such as physical activity, caffeine intake, smoking cigarettes, drug usage, even simple things such as eating, drinking, and conversation are confounders in the physiological measurements when compared with mood dysregulation [2]. This means these events affect physiology dramatically and can even mask the physiological response to mood dysregulation.

To the best of our knowledge, the literature does not yet address or solve all of these issues. It also does not provide a machine learning classifier capable of a passive, scientifically valid, and continuous prediction of mood dysregulation from physiological measurements collected in natural environments. Several attempts of measuring emotion or stress by using physiological measurements exist in past literature [2] [28] [29] [30] [31] [32] [33] [34] [35] [36]. Even though there is much literature listed here, the research performed and the measurement tools are not suitable and cannot be applied to subjects in their natural environments using a machine learning approach. Some more recent work has attempted to solve the problem with between-person differences in controlled environments. One attempt was using a personalized classifier for emotion in physiology [3] [37]. However, calibration stages are not scalable in controlled

environments, and thus is not a practical solution for people in natural environments. Therefore, more research is needed to provide a sufficient solution and a deeper understanding to classify mood dysregulation continuously collected in the natural environment using a machine learning approach.

1.4 Machine Learning Contributions

This research will present a new automatic machine-learning pipeline, also known as the Analysis of Mood Dysregulation (AMD). AMD is used to analyze mood or emotional dysregulation, caused by various underlining psychological disorders, in real-life subjects. AMD uses data collected by our smartphone-based mobile ambulatory assessment system (mAAS) for psychology research. AMD is also used to discover relationships between physiological and environmental sensor readings and mood dysregulation events. Working together, AMD and mAAS support the whole process of mobile ambulatory data collection, analysis and, optionally, intervention as show in Figure 4. In Figure 4, mAAS is the environmental data and AMD is the machine learning models. The system has been used to collect data from over 40 patients in their natural environment and machine learning analysis results are presented in this research.

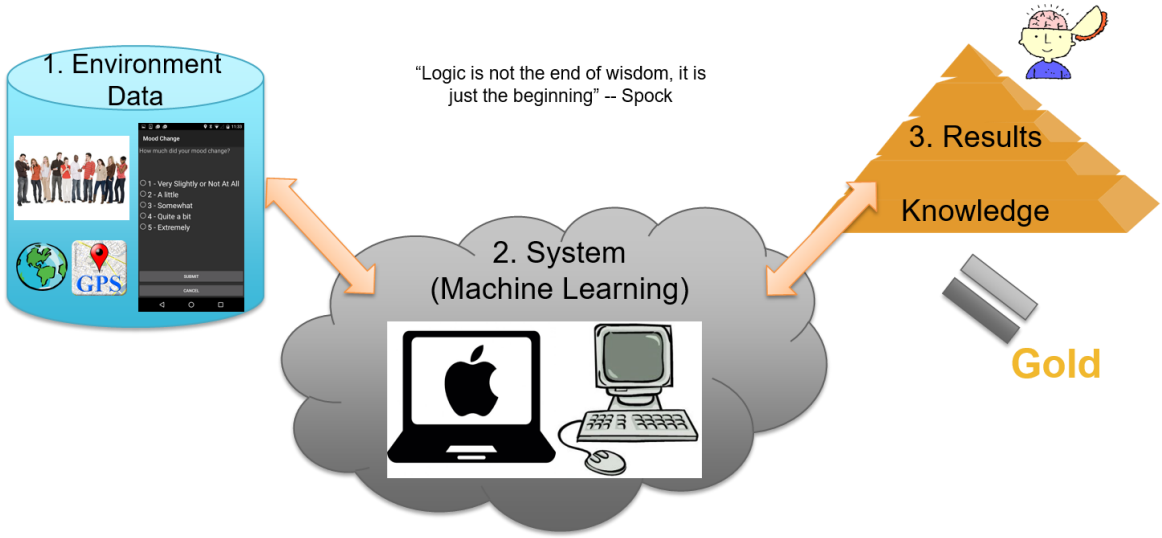

Figure 4. Analysis of Mood Dysregulation (AMD) workflow. The environmental data is collected by our mobile ambulatory assessment system (mAAS). The machine learning models are implemented in AMD's pipeline. Knowledge and results are analyzed which can be worth a lot if we can help save people's lives.

AMD uses several models to classify mood dysregulation using physiological sensors, each model predicts whether or not a one second measurement corresponds to a physiological response to mood dysregulation. Each model is useful as a direct measure of health outcomes that result from everyday wear and tear, such as heart or lung diseases. In addition, converting self-reported stress to a binary mood dysregulation state is not an obvious task due to subjectivity and wide between-person differences. Therefore, this research develops and trains a machine learning classifier to detect the mood dysregulation state from self-reports. To do this, first, the self-report data are combined with the physiological sensor data. Then, a machine learning classifier is performed on the combined data.

Our research differs from all other existing research because here, data is collected in the natural environment. In [2], they collect data in the natural environment for their field study but all three of their models are trained and tested using data from a 21-person lab study. Participants were *carefully* exposed to three diverse and validated stressors (public speaking, mental

arithmetic, and cold pressor challenges). Physiological data and self-reports were collected. Then the subjects were allowed in their natural environments while they tested their models.

However, in this research the models are trained and tested with data collected in the natural environment. The physiological data was captured using a newly developed, wearable, non-obtrusive wireless body metrics shirt called Hexoskin [38]. The Hexoskin provides various body metrics such as electrocardiography (ECG) using 3 cardiac dry and textile electrodes, respiration using 2 respiration loops (thoracic and abdominal) and a respiratory inductive plethysmograph (RIP), and activity using a 3-axis accelerometer [38].

AMD is modeled using a variety of ECG features, all of which have been proven to respond to mood dysregulation, such as heart rate, heart rate variability, and RR interval [2]. These metrics are complemented with additional features from respiration such as breathing rate, tidal volume, and minute ventilation (inductance plethysmography). After removing outliers and unreliable data from the feature set, I normalize the feature values to account for the baseline of each individual. Overall, there are approximately 12 features used for the models.

1.5 Mood Study Procedure

Participants were carefully selected based on various regulations. Then selected participants were fitted with the appropriate size Hexoskin and given an Android Nexus 5 smartphone. Participants wore the Hexoskin for approximately 15 days as they went about their normal daily life. Throughout each day, approximately 8 self-reports were administered, collected, and securely stored which include a morning report, 6 random reports administered at a semi-random times throughout the day, and a bedtime report. I applied several models to this field data after appropriate cleaning and screening of the data. Figure 5 depicts how the mood study procedure is constructed and applied.

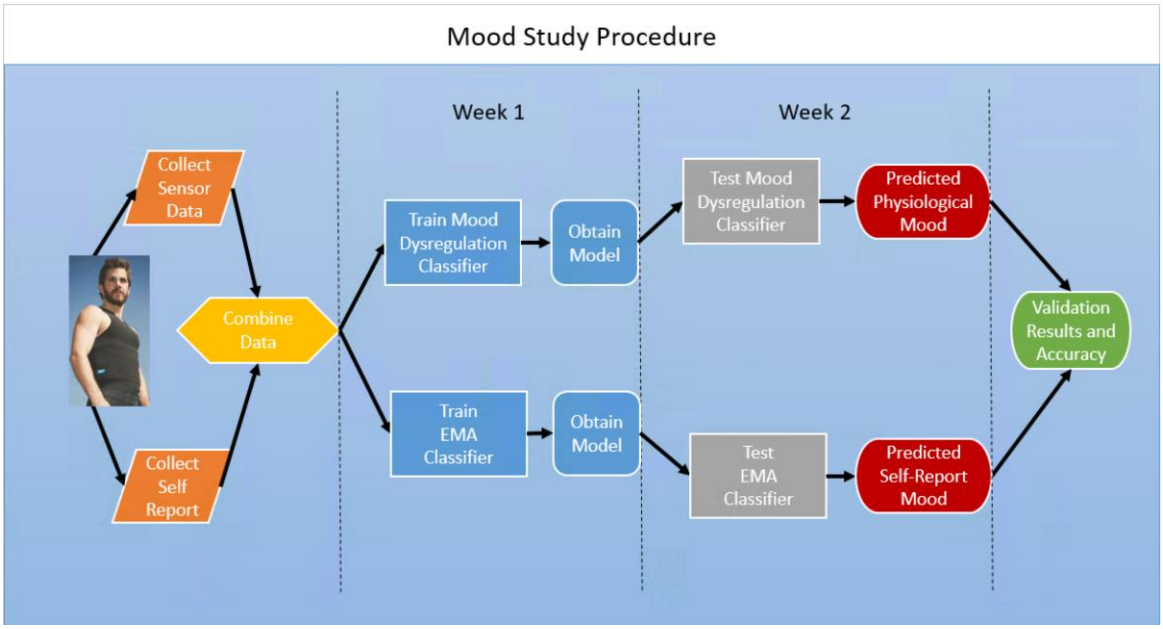

Figure 5. Flow chart of the mood study procedure. There are approximately two weeks of physiological and self-reported survey data obtained. During the first week, we trained and built the machine learning models. During the second week we tested the models and predicted mood dysregulation. Afterwards we analyzed the models and looked for knowledge from the data.

1.6 Discoveries and Knowledge Gained

This research discovers many different important factors for analyzing mood dysregulation from physiological measurements collected in the natural environment. One, after smoothing the data with statistical methods, the accuracy obtained was lower when compared with using the pre-processed data from Hexoskin. Two, each person is different and there are very wide between-person differences so to obtain the best results, each model has to be self-trained to each subject. Third, when using time as a key feature, the accuracy of the models are greater than 90%. Surprisingly, just using time and respiration features alone were key features for obtaining high accuracy. Fourth, when using ECG or RIP independently in the event that one sensor is not obtaining an accurate reading, AMD obtains an accuracy of greater than 80%. Such times include when a high activity level occurs. During these times, the ECG reading will be of poor quality.

1.7 Thesis Organization

This paper is organized as follows: Chapter II describes related work and supported literature. Chapter III goes over the mood study overview, procedure, and how each session was administered. Chapter III will briefly go over the background of our mobile Ambulatory Assessment System (mAAS) and the data collected from the system. Then at the end of Chapter III, a statistical analysis of the subjects is given. Chapter IV, describes the system improvements to mAAS implemented by me during the time of my research. Chapter V will present a new automatic machine learning pipeline called AMD, which selects the proper data, cleans and smooths the data, creates features, and runs several machine learning models for classifying mood dysregulation. Chapter VI will present the machine learning results and show different experimental results leading up to our best result. Chapter VII concludes this research and discusses the knowledge gained. Chapter VIII discusses possible future work and some ideas I have for this research in the future. Chapter IX presents the references used to create this thesis.

2. RELATED WORK

Machine learning and wireless body-area sensor networks have been a popular hot topic in recent research. Wireless body-area sensor networks have been used for a variety of applications in machine learning, mobile health, physiological monitoring, and context aware computing. Mobile systems have been developed to continuously collect biosensor and self-report data to assess or predict psychological states in [20] [23]. In [23], the iHeal research uses a biosensor that measures electro-dermal activity, motion, temperature, and heart rate to attempt to identify substance cravings. When the system detects a change in sympathetic nervous system activity, it collects information from the biosensor and self-reported information. The subject supplies information about stress, cravings, activities, environmental factors, and persons around them [23].

While self-reported information is currently in discussion, it is important to note that capturing self-assessment of emotion, usually through surveys and other self-reports, provides important yet oftentimes inaccurate information [39]. Results from lab experiments in [2], users correctly self-assessed their own stress only 84% of the time. Therefore, subjects may incorrectly self-assess for different reasons. First, humans do not necessarily experience emotions in a binary way. For example, a person can experience different degrees of stress at different times. One person may not feel stress during public speaking, while another may feel so stressed they are unable to express themselves correctly or express anything at all. Second, subjects may incorrectly self-assess psychological information in a natural environment because there is little control over the participants' physical and social environments, unlike in a laboratory setting, which makes the ability to identify the participants' contexts critical for the subjects and the sensors [40].

Research to determine drug usage in a subject's daily life is activity being pursued as well [41]. Experiments were done in [41] and a separate analysis was applied in field studies and in lab settings. Models were built on a mathematical principal to predict if the subject had used cocaine. The models in [41] were able to achieve a 100% true positive rate while keeping the false positive rate to 0.87 per day over 9+ hours per day of lab data and 1.13 per day over 11+ hours per day of field data. One main difference between detecting cocaine use and mood dysregulation is that the effect of cocaine is much greater and sharper on human physiological factors.

Research in [2] tested two models for continuous prediction of stress from physiological measurements captured by wearable sensors. Both models in [2] were trained using data collected from 21 subjects in a lab study. They were exposed to cognitive, physical, and social stressors representative of that experienced in the natural environment. Their physiological classifier achieved 90% accuracy and their perceived stress model achieved a median correlation of 0.72 with self-reported rating. However, from a total of 422 hours of data collected in the field, 37% of the data had to be removed due to confounding from physical activity [2]. An additional 29.45% of data were removed due to poor quality or losses in the wireless transmission. The stress models were applied to the remaining 33.55% of data (i.e., 142 hours) of valid data. In addition, out of 21 subjects, 4 subjects were eliminated from the analysis because of missing sensor data (ECG or RIP), excessive noise, and missing self-reports, leaving 17 subjects for the field evaluation [2].

With this being said, determining psychological states of subjects based on the physiology measurements is no straightforward task but has been examined from the 1800's. As early as 1890, William James brought interesting questions forward about the relationship between physiology and psychology [42]. More recently, in 1990, John Cacioppo and Louis Tassinary from

Ohio State University revitalized the interest for inferring psychological significance or states from physiological signals. Over the past couple decades, several attempts to identify predictors of stress have been tried. Various metrics have been proven to be activated by stress such as heart rate, heart rate variability, respiratory sinus arrhythmia (RSA), respiratory patterns, electro-dermal response and blood pressure [30] [31] [32]. While it has been proven these features do respond to stress in physiological data, they may be initiated by other factors on the body such as physical activity, changes in posture, speaking, etc. Hence, using these features to predict stress has been exceedingly difficult [2].

There are three main challenges our research has found when determining mood dysregulation from physiological measurements, all of which are supported in the literature. The first challenge is to overcome the confounding factors that may mask the changes in physiology caused by the change in mood dysregulation. In 1996, Michael Myrtek and Georg Brugner from Freiburg, Germany attempted to predict changes in human emotion from physiological measurements. However, they did not find significant correlations between those exhibited by physiology and those collected by self-reports [34]. The main hypothesis for the lack of a good correlation from their research was the presence of confounders in the physiological data. Even in more recent attempts in detecting emotion in the natural environment by Jennifer Healey, Lama Nachman, et al. in 2010, only the measurements collected near the markings provided in the self-reports by the subjects were used. This was due to the lack of ground truth available for the rest of the available data [29].

The second challenge found in predicting mood dysregulation from physiological measurements is dealing with and accounting for the wide between-person differences. In 2008, Jonghwa Kim and Elisabeth Andre from University of Augsburg observed that a personalized

model produces better accuracy than a population-level model [31]. In 2010, Yuan Shi and others from Carnegie Mellon University obtained a similar result also proving that personalized stress models are better for detecting stress from physiological measurements [37]. However, personalized models are not as practical since they require collecting training data on each subject in order to produce the personal classifier [2]. Therefore, there needs to be an approach in the middle where the model is not personalized at the individual level nor is it generic to the population level, but personalized to groups or categories of subjects.

The third, more trivial, challenge is having a wearable sensor that people are willing to wear in order to collect accurate, consistent, and valuable data. As well as a mobile application the user is willing to put time and effort to answer survey prompts throughout the day. The sensor must collect measurements from multiple modalities and process them on the body [2]. Leveraging many recent developments in wearable sensing and smartphones, many thanks goes out to Hexoskin for developing their wearable body metrics shirt which is one of the most advanced biometric shirts on the market today. The Hexoskin collects ECG which is used to calculate heart rate and RR intervals, respiration which is used to calculate minute ventilation and tidal volume, activity which is used to calculate cadence and calories burned, as well as a long list of other metrics. Hexoskin connects to the smartphone through Bluetooth and then wirelessly transmits all this data to the smartphone. The smartphone will then upload all of the physiology data to their servers where users can view the data through a dashboard as well as download the data individually by each record [43].

3. MOOD STUDY OVERVIEW

This research conducted a two-phase user study to collect training and testing data for the machine learning models of mood dysregulation. Each subject was carefully selected and invited into the lab for Hexoskin fitting, Android smartphone setup, training for cell phone application usage, and then let out into their normal daily activities. In the first session, physiological, random assessments, and self-report measures were collected from over 40 subjects while they were in their natural environments. The first phase data was collected for one week, which provided the training data needed to develop the machine learning models for mood dysregulation. The subjects were invited back to the lab to backup all of their data, fix any problems associated with the Hexoskin or smartphone, etc.

In the second session, physiological, random assessments, and self-report measures were collected again from the same participants in their natural environments. The second phase data was also collected for one week but now the machine learning algorithms were predicting mood dysregulation for each subject during their second week or second session. Figure 6 shows a visual representation of how machine learning was incorporated into the mood study. This section describes the study in more detail, including the population from which the participants were selected, the measures collected from them, the Hexoskin wearable body metrics, the lab procedure as well as the field procedure, and a statistical analysis about the data collected.

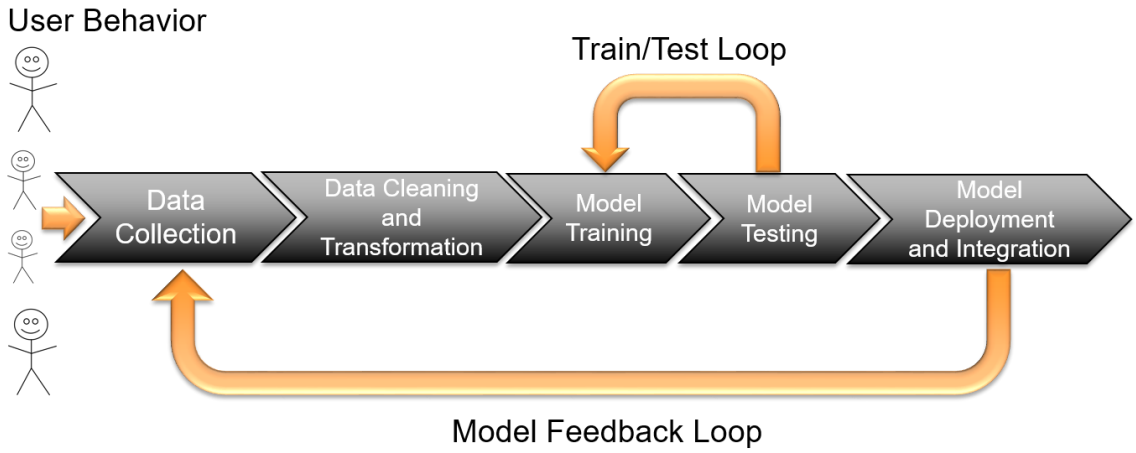

Figure 6. Visual representation of how machine learning was incorporated into the Mood Study. First we collect user behavior, then we perform data clean and transform the data by creating some features, next comes model training and testing which can be in a loop executed several times, then we deploy the model and try to predict Mood Dysregulation from real-life subjects in the study. This process is then repeated several times through the model feedback loop to achieve the best model.

3.1 Participants

This study was done at the University of Missouri campus in Columbia, Missouri (Mizzou). Participants were 40 females in treatment for a disorder of emotional distress. Participants were recruited through an outpatient clinic in Columbia, Missouri as well as announcements in a weekly news bulletin published through Mizzou. Participants met the general eligibility criteria if they:

1) Were between the ages of 18 and 45,
2) Were not pregnant or planning to become pregnant,
3) Had no history of head trauma that has resulted in sustained impairment in mood, attention, or concentration,
4) Did not have a medical diagnosis of cystic fibrosis or diabetes, as these impact sympathetic nervous system activity, and
5) Are not contraindicated for magnetic resonance imaging.

Additionally, participants were required to meet DSM-5 diagnostic criteria for a mood disorder (e.g. current MDD, bipolar I, or bipolar II), an anxiety disorder (e.g. GAD, PTSD, or Social Anxiety Disorder), or Borderline Personality Disorder. Eligibility was determined through a diagnostic interview that included the MINI and SIDP-IV structures.

3.2 Measures

3.2.1 Sensory Measurements

The Hexoskin Wearable Body Metric Shirts, shown in Figure 7, were used to monitor various metrics known to respond to mood dysregulation, stress, and other psychologically and physically demanding conditions. These metrics include categories such as cardiovascular, respiratory, activity, and thermoregulatory systems [2]. Four sensor were used in this study, the first three of which are available from the Hexoskin:

1) An electrocardiograph (ECG) attached to the body with 3 cardiac dry and textile electrodes to measure electrical output of the heart,
2) Two respiratory inductive plethysmograph (RIP) bands to measure relative lung volume one of them is thoracic located at the rib cage and the other is abdominal located around the user's stomach,
3) A three-axis accelerometer used to determine the activity of the user such as cadence, calories burned, distance traveled, number of steps, etc.,
4) A global positioning system (GPS) which is on the smartphone used to obtain the latitude and longitude values for the user's location.

Figure 7. The Hexoskin Wearable Body Metrics Shirt, which includes measures of ECG, respiratory inductive plethysmograph (RIP), and three-axis accelerometer. 1) The photograph on the left is an example of someone wearing the Hexoskin. 2) The photograph in the center is a picture of the Hexoskin with the processing mote and charger, located below the shirt. 3) The photograph on the right shows where the sensors are located in the shirt, the three blue squares are for the cardiac sensors, the two white bands are for the respiratory sensors, and the activity sensors are located inside the processing mote.

The Hexoskin uses a secure Bluetooth wireless communication using an 802.15.4-to-Bluetooth bridge that sends the data received from the sensors to a mobile phone via Bluetooth. The smartphone then sends the data to the server via cellular or WIFI connection with full real-time data transmission. Hexoskin comes with lithium-ion batteries with a battery life greater than 14 hours in recording mode, 400 hours in sleep mode, and capable of USB fast charging in 90 minutes [44]. The frequency for the ECG channel is 256Hz, the two respiratory channels at 128Hz, and the three-axis acceleration channel at 64Hz with 0.004g resolution [44]. The Hexoskin has a 1GB memory capacity which is equivalent to approximately 157 hours of full raw data capacity [44]. The Hexoskin can be used with the two main smartphone operating systems available, iOS and Android. Other features include automatic garment connection detection, automatic start recording on garment connection, automatic stop recording on garment disconnection, event

marking with button during recording, and 3 LEDs for battery status, recording status, and Bluetooth status located on the mote module [44].

3.2.2 Survey Reports

There are three main surveys the participants would answer on a daily basis: the A) morning report parcel, B) random assessment parcel, and C) mood dysregulation parcel. Each main survey may contain one or more of the following subset categories of questions and there are five total subset categories of questions: 1) mood and impulsivity items, 2) situation and setting items, 3) mood change items, 4) life events/experiences items, and 5) behavioral dysregulation items. Each of the three main surveys are described in further detail below and an example of each of the subset categories of questions are given.

A. Morning Report Parcel

During the participant's mood study, each participant was required to complete a morning report parcel every morning when they woke up each day. Each night when the user goes to bed, they will set a morning report prompt alarm. If no alarm for the morning report has been set, the alarm will default to noon the next day. Once the morning report has been completed, the random assessment parcel schedule will be created for that day. The morning report contains two main categories of questions 1) mood and impulsivity subset items as well as 2) situation and settings subset items.

The mood subset included questions answering how much the subject felt the following in the past 15 minutes such as afraid, nervous, scared, upset, frightened, angry, alone, alert, proud, strong, excited, guilty, and many other emotions. The mood subset questions also included subjects answering "How well does this describe the subject over the last 15 minutes" which included questions like "I felt and acted on a strong impulse," "I gave up easily," "I did something

for the thrill of it," and "I did something without really thinking it through". These questions were administered randomly, so each time the subject answered the mood subset questions they would not be in the same order. Each of these items were answered on a five-point scale: 1 – Very Slightly or Not At All, 2 – A little, 3 – Somewhat, 4 – Quite A Bit, 5 – Extremely. Examples of the morning report parcel are show in Figure 8.

Each morning report parcel contained a subsection of situation and setting questions. This subset of questions included a multiple selection type for each answer. The situation and setting included questions answering in the past 15 minutes who have you been with: (Check all that apply) and in the past 15 minutes where is your location: (check all that apply). The available answers to the situation and setting question included no one, partner/spouse, other family member, friend/acquaintance, and other. The available answers to the location question included home, work, bar/restaurant, outside, other public place, and other.

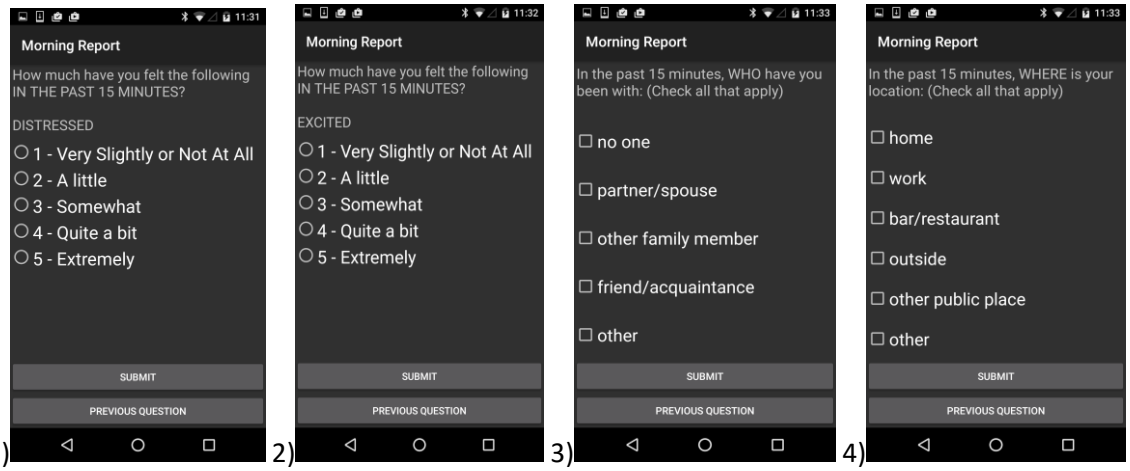

Figure 8. Four examples of the morning report parcel. The morning report parcel is administered every morning at the alarm time when the user set the morning report prompt the night before. 1) Asking the user if they felt distressed in the past 15 minutes. 2) Asking the user if they felt excited in the past 15 minutes. 3) Asking the user who they have been with in the past 15 minutes. 4) Asking the user their location in the past 15 minutes. Examples 1 and 2 are from the mood subset questions and examples 3 and 4 are from the situation and setting subset questions, both of these combined create the morning report parcel.

B. Mood Dysregulation Parcel

The mood dysregulation parcel contains four categories of questions: 1) Mood subset questions discussed in the morning report section, 2) situation and setting subset questions discussed in the morning report section, 3) Mood change subset questions, 5) and Behavioral Dysregulation Items which will be discussed next in the random assessment parcel. The mood dysregulation parcel is a self-report survey where the user will trigger the survey's administration. This is important to note when reading about the machine learning implementation because these mood dysregulation surveys are the true positive sample for the predication.

The mood dysregulation parcel includes questions such as: "How much did your mood change," "Are you in a better or worse mod than before," "If better/worse," and "What triggered your mood change." Whenever the user experiences mood dysregulation they are trained to fill out a survey. Once the machine learning algorithms in AMD are trained and calibrated to the user, AMD will prompt the user automatically whenever the physiological sensors show signs of mood dysregulation. This allows the user to focus more on their everyday tasks and the researchers to get more real-time accurate data. Examples of the mood dysregulation parcel are show below in Figure 9.

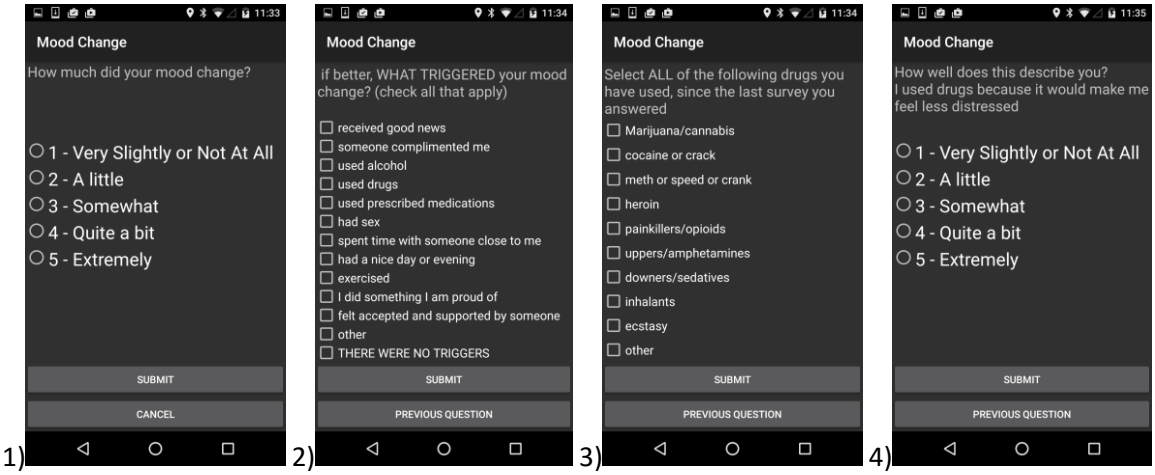

Figure 9. Four examples of the mood dysregulation parcel. This survey is self-administered by the subject whenever they feel mood dysregulation by clicking the survey button. 1) Asking the user how much their mood has changed on a five point scale. 2) Determining why their mood has changed. 3) Distinguishing the drugs the user has used since last survey they answered. 4) Asking the user if they used drugs because it made them less distressed and answered on a five-point scale. Examples 1 and 2 are from the mood change subset questions and examples 3 and 4 are from the behavioral dysregulation items.

C. Random Assessment Parcel

Throughout the day, the random assessment parcel is triggered six times at random time periods during the ambulatory assessment of the subject. A scheduling scheme has been employed to make sure that the random surveys are triggered during the time periods where subjects are most active during their ambulatory assessment and to increase the randomness of the survey. A service running on the background collects the time periods and calculates the minimum delay the surveys have to maintain between times, before another survey is triggered. This will increase the randomness of the survey and ensure that the random survey prompts are noticed by the subject.

If we were to prompt a user at 4am, the user will probably not notice the prompt, therefore we must calculate when the subject is most active while using the mobile application. However, for another user who works nights, this user might be more active at 4am, therefore the application will calibrate to each subject in order to increase subject compliance. The random

assessment parcel includes four categories: 1) the mood subset questions listed above, 2) the situation and setting questions listed above, 4) life events/experiences, 5) and behavioral dysregulation items.

The life events/experiences includes questions such as since the last prompt have you had a disagreement (if yes, with whom), have you received bad/good news, have you slept, have you used caffeine, have you used over the counter-medications, and have you taken your medications as prescribed. Examples of the random assessment parcel are shown below in Figure 10 visualized in four screenshots from the mobile application.

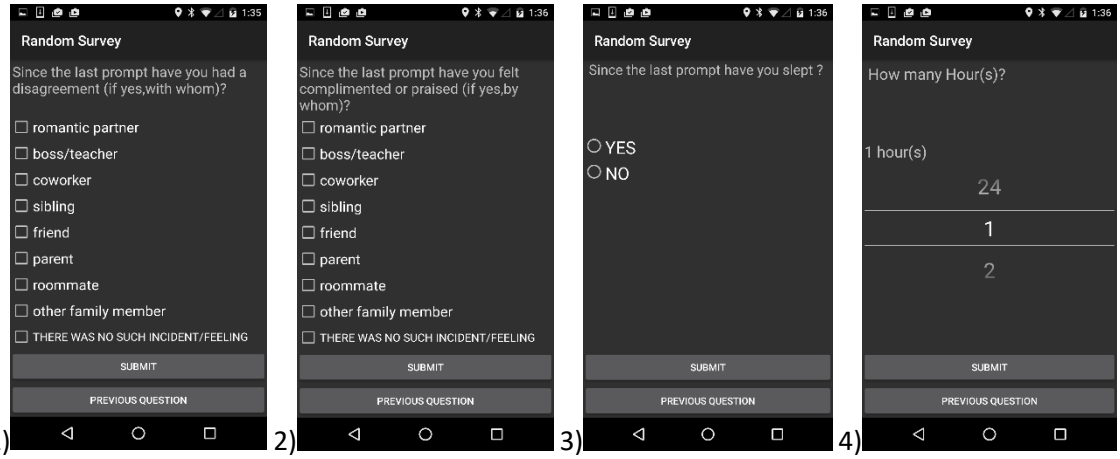

Figure 10. Four examples of the random assessment parcel. This survey is administered six times randomly throughout the day when the user is most actively using the mobile application. Example 1) is asking the user if they had a disagreement and with who, 2) determining if the user has been complemented or praised and by whom, 3) whether the user has slept, 4) for how many hours, all examples are asking since the last prompt or survey. Examples 1-4 are from the life events/experiences subset questions.

The behavioral dysregulation items include questions such as in the last 15 minutes how much were you focusing on your feelings, since the last survey you answered have you consumed alcohol, I drank alcohol because it would make me feel less distressed, select all of the following drugs you have used since the last survey you answered, I used drugs because it would make me feel less distressed, etc. Examples of the random assessment parcel are shown in Figure 10. Other

questions include since the last survey you answered did you ask someone you feel close to how they truly feel about you or whether they really care about you, have a period of uncontrollable eating, have you spent more money than you meant to, and other various questions.

3.3 Lab Procedure

3.3.1 Phone Screening

Participants emailed the administrator if they saw one of the ads posted (see section 3.1 Participants) and were interested in learning more about the study. The participants were called and they completed a 10-20 minute phone screening. The phone screening consisted of questions to determine the presence of emotional distress and Magnetic Resonance Imaging (MRI) eligibility. Participants who seemed eligible at this point were schedule for Session 1.

3.3.2 Session 1

Participants signed the consent form in the lab and were given a copy to take home. Then the Mini-International Neuropsychiatric Interview (MINI) and Structured Interview for DSM-IV Personality (SIDP) were performed to determine full eligibility. MINI is a short structured diagnostic interview, developed jointly by psychiatrists and clinicians in the United States and Europe, for DSM-IV and ICD-10 psychiatric disorders [45]. With an administration time of approximately 15 minutes, it was designed to meet the need for a short but accurate structured psychiatric interview for multicenter clinical trials and epidemiology studies and to be used as a first step in outcome tracking in research clinical settings [45]. SIDP is a semi-structured interview that uses non-pejorative questions to examine behavior and personality traits from the patient's perspective [46]. The SIDP-IV is organized by topic sections rather than disorder to allow for a more natural conversational flow, a method that produces a more accurate diagnosis [46].

Both of these test combined took approximately two hours to complete for each of the subjects. Participants also completed the following questionnaires. As a measure of mood and anxiety symptoms, participants completed the DASS-21. For overall emotional regulation, participants completed the Difficulties in Emotional Regulation Scale (DERS; Gratz & Roemer, 2004). Participants completed the Ruminative Responses Scale as a measure of rumination (Treynor, Gonzalex, & Nolen-Joeksema, 2003), the Acceptance and Action Questionnaire (AAQ-II) as a measure of experiential avoidance (Bond et al., 2006). Participants also completed the UPPS-P as a measure of impulsivity. Finally, participants provided frequency and quantity of drug and alcohol use. At the end of Session 1, Session 2 was scheduled.

3.3.3 Session 2

Session 2 contained Magnetic Resonance Imaging (MRI) for each subject which took place at the Brain Imaging Center. The MRI took approximately 2.5 hours. For the first 30 minutes, participants signed paperwork related to scanning procedures, prepped for the scan by removing all metal from clothing or various accessories or other items, and the administrator oriented them to the scanning procedures as well as the two tasks to be completed in the scanner. Participants then spent 1.75 hours in the scanner, completed the scans, and tasks in the following order: structural, task 1, task 2, resting state, Diffusion Tensor Imaging (DTI). Participants were allowed to take as many breaks as they would like and terminate the scan at any time. All but one participant completed the full scanning procedure. After scanning was completed, participants were trained on the Ecological Momentary Assessment (EMA) portion of the study. They were then assigned a study phone, Hexoskin, and Q-Sensor. Participants started the EMA portion on the same day as the scan unless they requested otherwise. Next, the field procedure will be discussed in further detail in section 3.4 Field Procedure.

3.4 Field Procedure

Each participant was outfitted with the Hexoskin sensors and given an Android Nexus 5 smartphone to carry with them for a total time of two weeks as they went about their normal daily life. The smartphone is the central point of the mAAS system, managing multiple wireless connections with external sensors, collecting sensor data, and transmitting the data to cloud web servers when it is connect to an internet connection. In addition, the system has a survey module implemented on the smartphone, using multiple preset and randomly triggered surveys to determine the emotional state of the subject along with environmental factors that triggered mood dysregulation the subject is experiencing. The first week of data collected is used for training the machine learning models. After the training data is collected the user will come back into the lab to backup all of their physiological and survey data as well as fix any problems associated with the mobile application, Hexoskin, or other various problems the user might be experiencing.

Once everything is in the proper order and everything fits and works correctly, the user is released back into their natural environment for the second week of data collection. Therefore, we used the second week of natural environment data to test the machine learning models of mood dysregulation derived from the first week of natural environment data. While the user was in their second week, the machine learning algorithms will be constantly looking at their physiological data, since the smart phone uploads the data to the server in semi-real time. When the model predicts mood dysregulation, the server will be triggered to initiate a mood dysregulation parcel prompt on the smartphone. The user will then respond to the survey and answer the corresponding questions. Since the machine learning models predict mood dysregulation, the user can focus on their daily life activities and the researcher will obtain more real-time accurate data. Results from lab experiments in [2], users correctly self-assessed their

own stress only 84% of the time. Therefore, the data collected will be more accurate and have a stronger correlation with the physiological data.

During the two weeks of Ecological Momentary Assessment (EMA) data collection, participants' compliance was monitored and they were contacted if they appeared to have substantial missing data. Session 3 was a check-in that took place approximately one week into the EMA, discussed earlier in this section. Participants were paid based on their compliance with the phone surveys and data downloaded from their Hexoskin and Q Sensor devices. Most of the participants completed both weeks of the EMA data collection. At session 4, participants were paid for the final week of the study, based on Week 2 compliance, and returned all study equipment. Figure 11 below is a visual representation of the Mood Study Procedure including the two weeks and how the machine learning was incorporated.

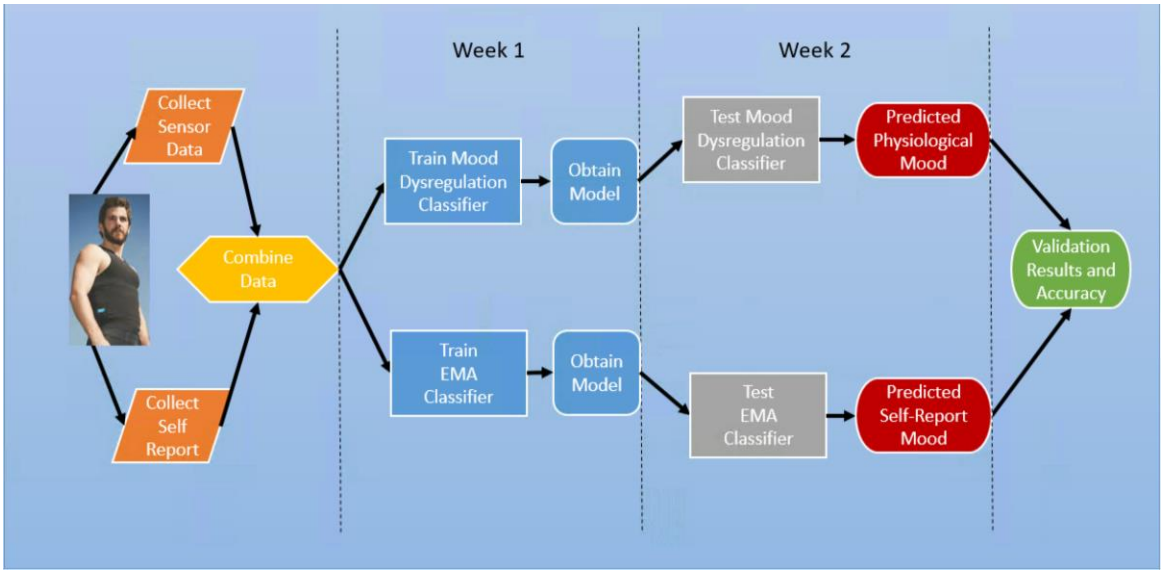

Figure 11. A visual representation of the Mood Study Procedure including the two weeks of Ecological Momentary Assessment (EMA) data collection and how machine learning was incorporated in the study.

3.5 Statistical Analysis

When we started the machine learning approach on the Mood Study data, there were 22 out of a total of 40 participants that completed the study. Currently there are more participants that completed the study and AMD's pipeline can be applied to the new data. From the 22 subjects that were included in AMD's research, there were 318 days of survey records for all the subjects. Figure 12 shows graphs representing total days for surveys for all the subjects and total days for each survey for all subjects.

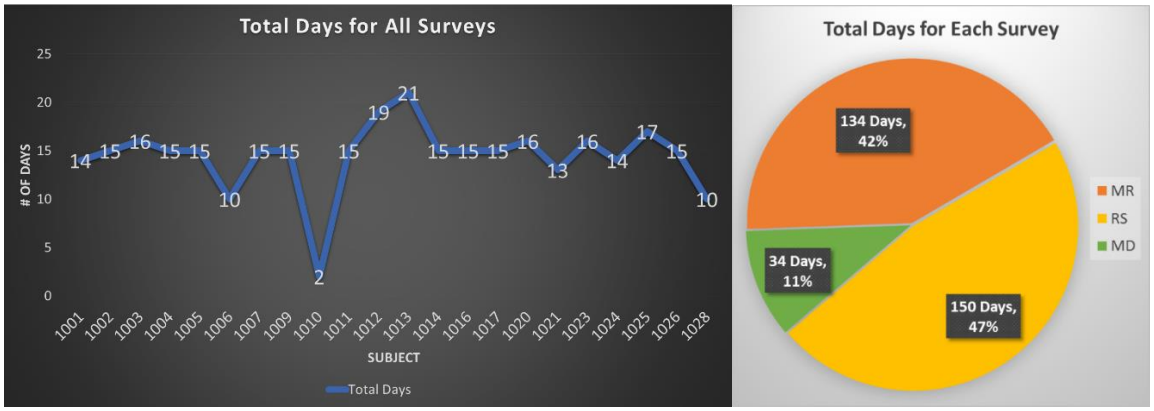

Figure 12. Graphs representing data in the Mood Study. The graph on the left shows total number of days for all surveys for each subject. The graph on the right show the total number of days for each survey for all subjects.

For the 22 subjects, there are 258 days of sensor records from the Hexoskin sensor consisting of 9 physiological attributes. The survey data was combined with the sensor data. To do this, we took the start and end times of the mood dysregulation surveys and cross-referenced them with the times of the sensor data records. Each sensor data record corresponds to one second of each hour. Therefore, each second in-between the start and end times of the surveys that matched with the sensor data was marked as mood dysregulation. After combining the survey and sensor data, there were 197 days of sensor data, approximately a 23.64% decrease of data. Figure 13 below shows the total days for sensor data before combining and total days for

sensor data after combining with survey data. Figure 14 below shows the total days for surveys before they are merged with sensor data and after.

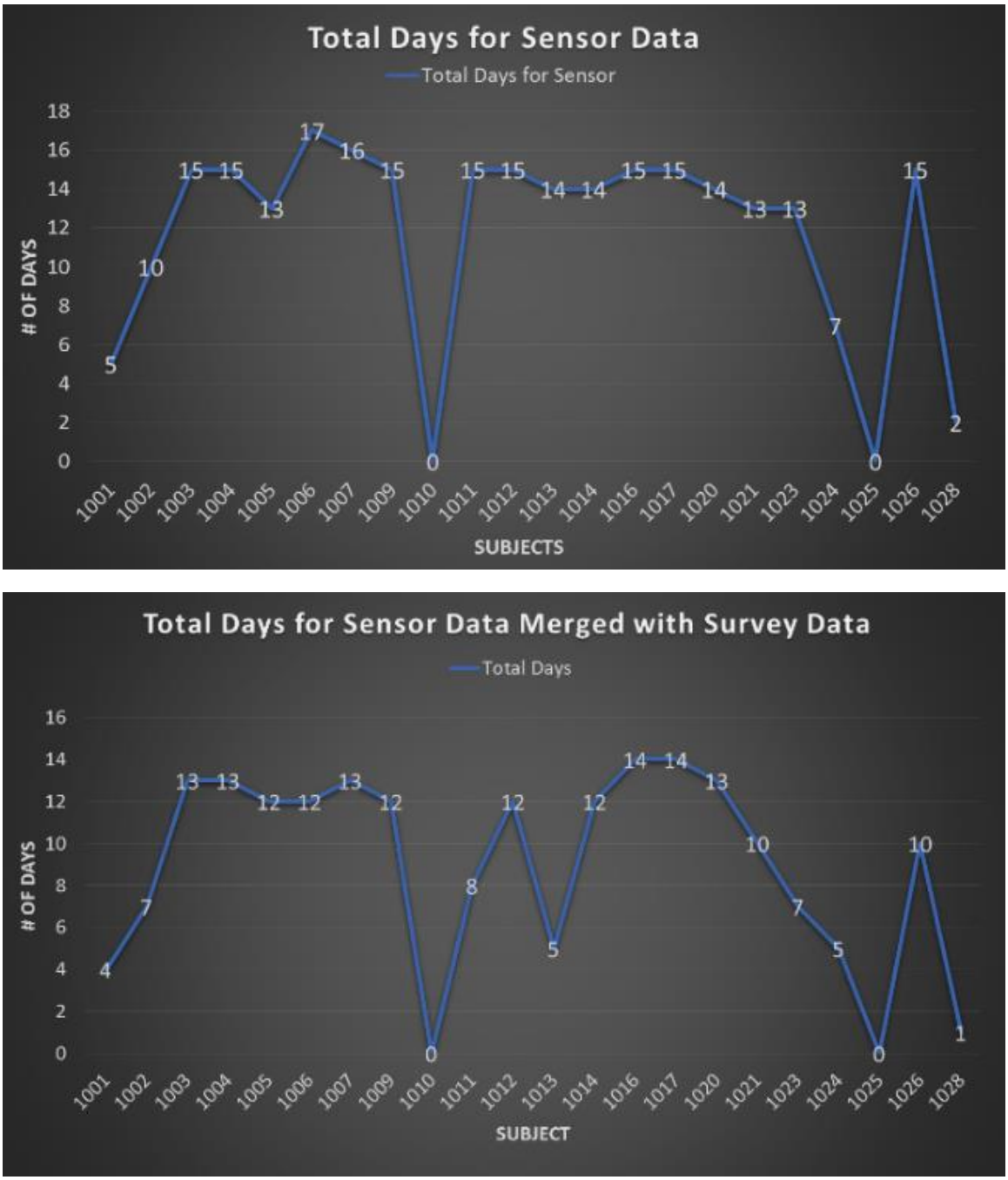

Figure 13. Graphs representing the total days of sensor data available from 22 subjects. The top graph shows the total amount of days for each subject. The bottom graph shows the change in the amount of days after combining the sensor data with the survey data.

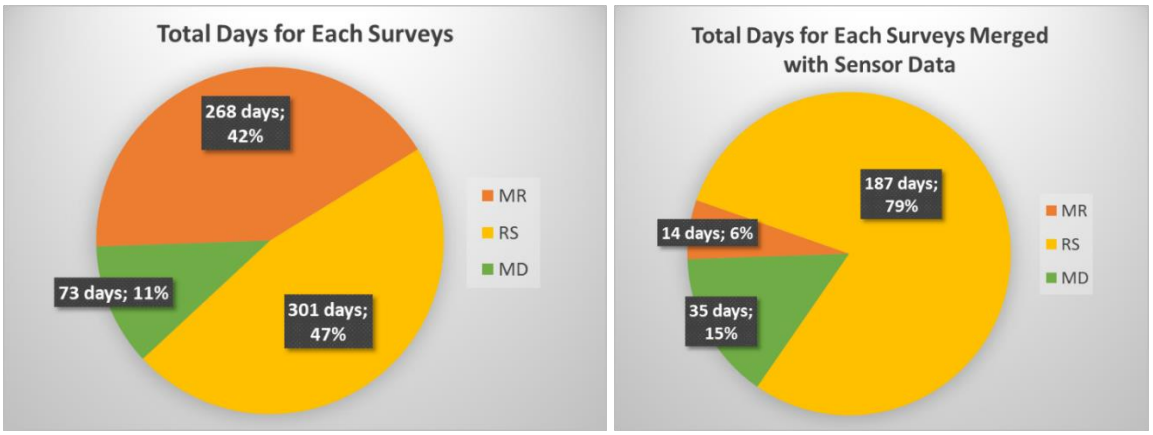

Figure 14. Graphs representing the total days for each surveys available from 22 subjects. The left graph shows the total days for each survey before merged with the sensor data. The right graph shows the total days for each survey after it is merged with the sensor data. For mood dysregulation, there was a 52.05% decrease in the amount of days of data.

Looking at records instead of days, there are 8,154,183 sensor records from 22 subjects collected in the natural environment. For surveys, there are 1,625 survey records. There were three categories of surveys: Morning Reports, Random Surveys, and Mood Dysregulation. For Morning Reports there were 275 surveys out of 1,625 total surveys, or 17%. For Random Surveys, there were 1,246 surveys out of 1,625 total surveys, or 77%. For Mood Dysregulation surveys, the main survey for this study, there were 104 surveys out of 1,625 total surveys, or 6%. When combining the sensor data with the survey data, there were 1,362 sensor records for morning reports, there were 62,353 sensor records corresponding for random surveys, and there were 5,601 sensor records for Mood Dysregulation. If you add all those numbers up, 1,362 + 62,353 + 5,601 = 69,316 which is nowhere near the total amount of sensor records, over 8 million. Figure 15 below shows the ratio of sensor records when combined with survey records and then the difference when compared with all the sensor records.

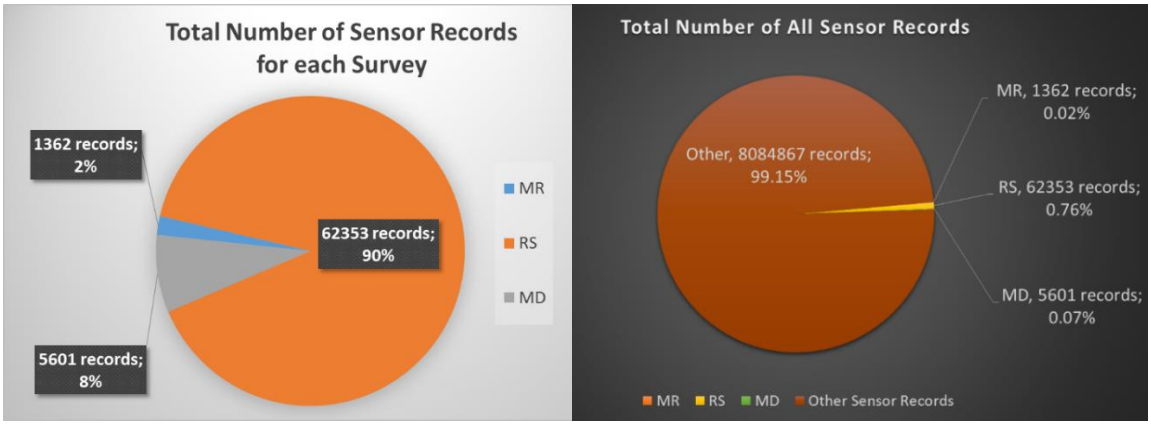

Figure 15. Graphs to visualize the result after Sensor Records are combined with Survey Records. The graph on the left shows the ratio between each category of surveys. The graph on the right shows the same ratio when considering all the sensor records. Mood dysregulation surveys are analyzed in this research which consists of less than 1% of all data.

There were originally over 8 million sensor records and after combining the sensor data with the survey data there were 69,316 associated sensor records. This is a decrease of 99.15% of the original data. This means that there is less than 1% of sensor data that corresponds with the survey data. This caused some problems when analyzing all the subjects. Before combining the data the subjects might have had plenty of sensor data and survey data. However, some subjects had to be dropped because after combining the data there might not have been any corresponding records. Figure 16 below shows the total days for all surveys, the total days for sensor data, and then the result after merging the survey and sensor data together. It is shown that two subjects do not have any corresponding data after combining.

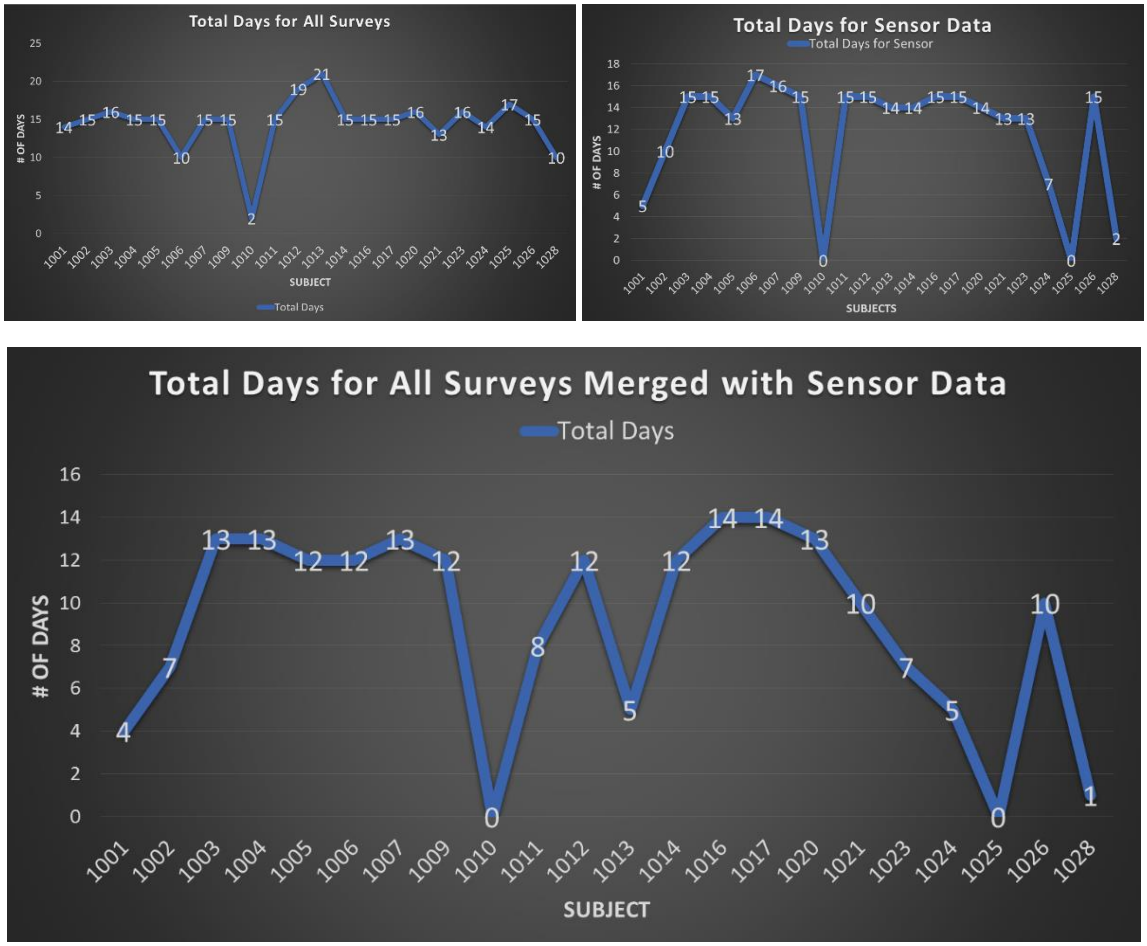

Figure 16. Graphs visualizing the results after combining the two data sets, survey and sensor, respectively. The top left graph represents the total days for all the surveys for each user. The top right graph represents the total days for all sensor data for each user. The bottom graph shows the consequences of combining the two data sets. It can be seen two subjects do not have any corresponding sensor and survey data after combining, thus these two subjects have to be removed.

Taking a closer look at all the surveys for all the subjects we can generate a few more graphs. Since this study focuses only on Mood Dysregulation surveys, we can take a look at these surveys individually. Figure 17 shows the total Mood Dysregulation surveys with the total amount of sensor records for each user.

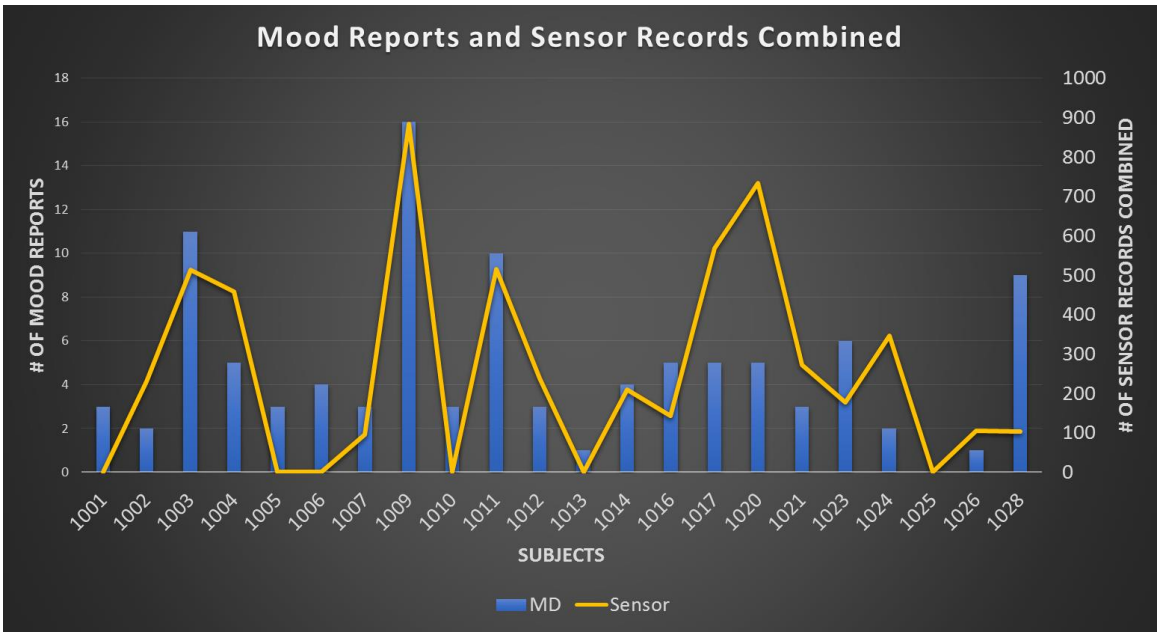

Figure 17. Graph to show Mood Dysregulation surveys and sensor data combined. In the graph, we can see subject 1009 has the most Mood surveys and corresponding sensor records. However, there are 6 users without any Mood surveys corresponding with sensor records. Later these subjects were dropped from the machine learning analysis.

With over 8 million total sensor records, there are only 5,601 corresponding records for mood dysregulation. Out of 318 days of survey data, there were only 73 days for mood dysregulation. After combining the survey data with the sensor data, 6 users did not have any corresponding mood dysregulation survey data with the sensor data. This causes a problem with the machine learning models. As a result these users had to be removed. This will be discussed in further detail in 4.2.1 Data Selection and 4.2.2 Data Combining. Figure 18 below shows the total days for each survey for each subject and then shows the results after the survey data is combined with the sensor data.

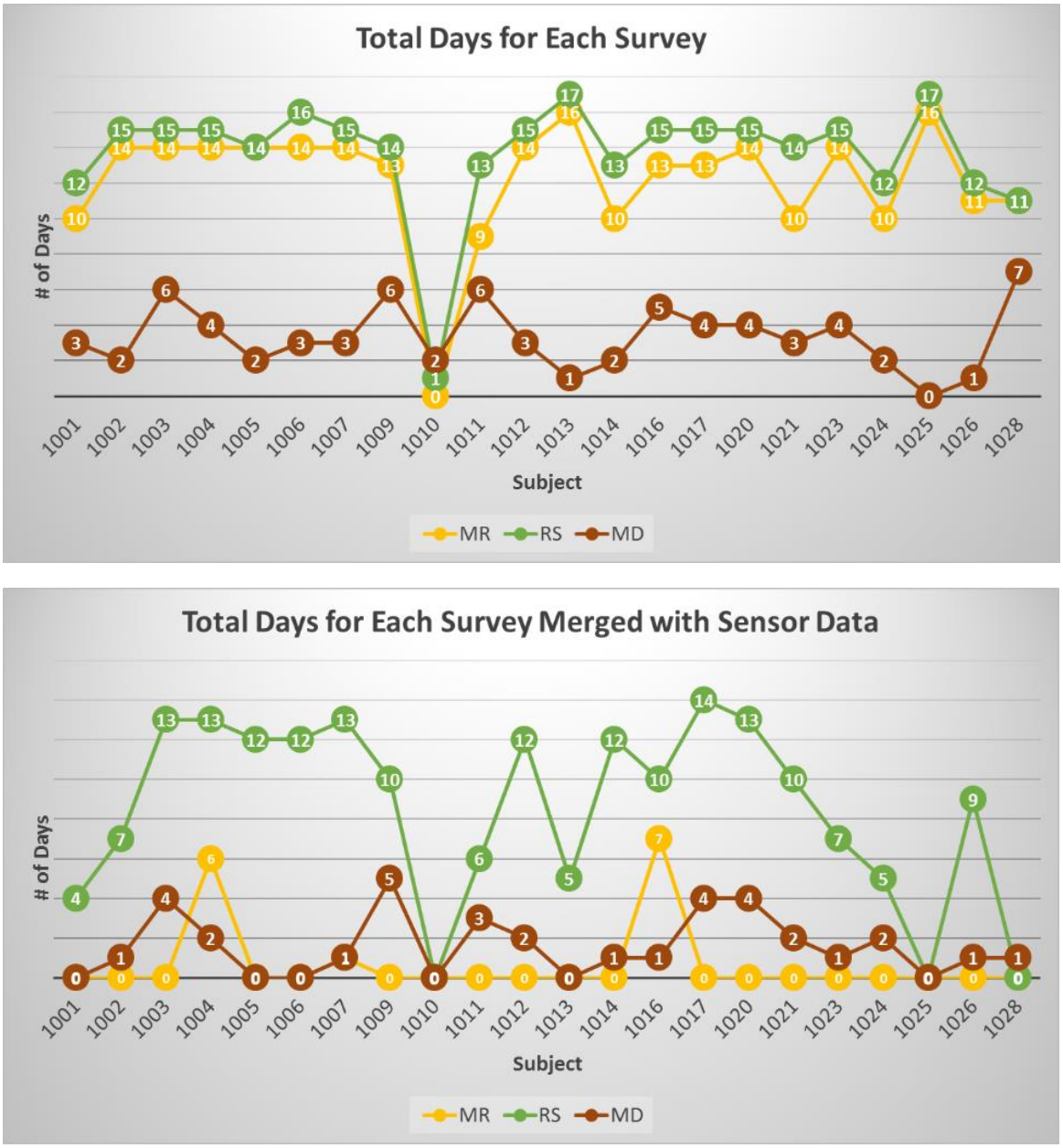

Figure 18. Visual representation showing the results after combining the two data sets. The top graph shows the total days for each survey for each user. The bottom graph shows the results of combining the survey and sensor data and how this effects Mood Dysregulation surveys which is the main focus point of this study. There were 6 subjects who did not have any corresponding Mood Dysregulation reports with sensor data, as a result these subject are removed.

4. MAAS IMPROVEMENTS

4.1 mAAS Overview and Introduction

Our Mobile Ambulatory Assessment System (mAAS) is used to monitor and collect real-time factors for subjects in the natural environment without pre-calibration. The system contains three main components: wireless external sensors, mobile devices, and cloud web servers. The smart phone is the center point of the system, managing multiple wireless connections with external sensors, collecting sensor data, and transmitting the data to the cloud when it is connected to the Internet. In addition, the system has a survey module implemented on the mobile device. It uses multiple preset and randomly triggered surveys to determine the emotional state of the subject. Along with the environmental factors that prompted the subject to have mood dysregulation.

The external sensors connect to the phone and upload physiological data to the server. There is a server program to provide interaction with the mobile application and to display or monitor the data in real-time. The entire mAAS system will collect, store, and display physiological and environmental sensor data from subjects in the study. The smart phone is developed as an Android application. The wireless sensors connect to the smart phone via Bluetooth. The server program is implemented in PHP, HTML, JavaScript, and CSS. There is an interface to display the data once it reaches the server and there is a MySQL database to store the data. The data has a backup storage in the flat file system. Figure 19 below is a flow chart of the mAAS system and the three main components, along with the surveys, to determine the emotional state of the subject.

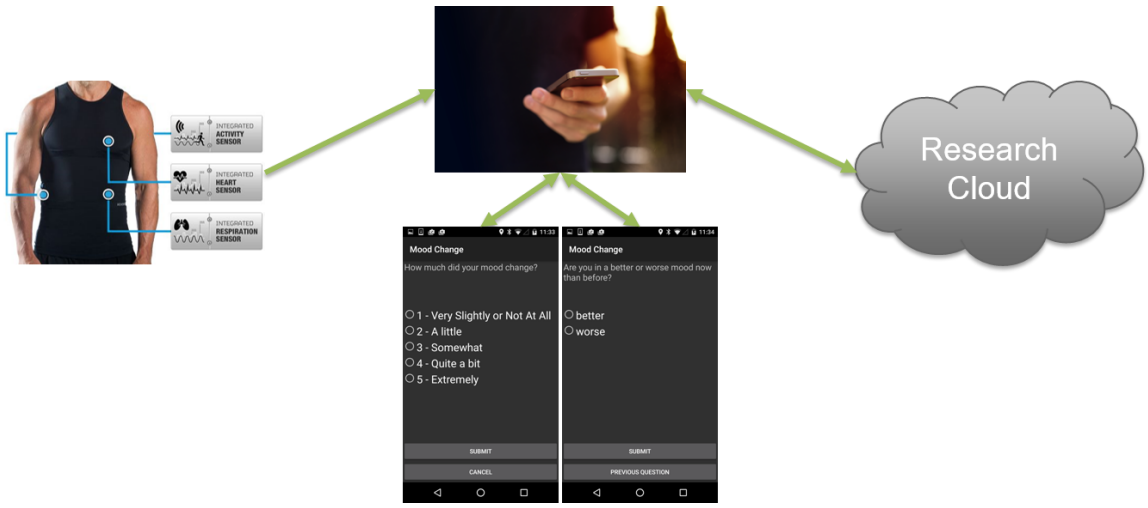

Figure 19. A flow chart of the mobile ambulatory assessment system (mAAS) and the three main components: External sensors, mobile devices, and cloud web servers. The surveys managed by the smartphone are used to determine the emotional state of the subject used in the Analysis of Mood Dysregulation (AMD's) pipeline.

4.2 System Improvement: mAAS Hardware Information Collection

The mAAS system had some problems the users reported. We did not know if the system was causing the problems or if it was user error. Therefore, I developed a monitoring system for the hardware on the mobile device used by the application. We would reference mAAS hardware information as "Hardware info," which will be used proceeding in this discussion. Hardware info can determine if the user turns the phone on/off, starts/stops the application, whether the mobile device has Internet connection, and other valuable information. Below in Figure 20, we display a visual of the android lifecycle and various modules that we received information from. For example, the onCreate() function was called by the Android operating system when the user started our application, therefore at this stage we would report the user started the application. The onDestroy() function would be called by the Android operating system when the user killed our application, which would be reported and a log would be generated.

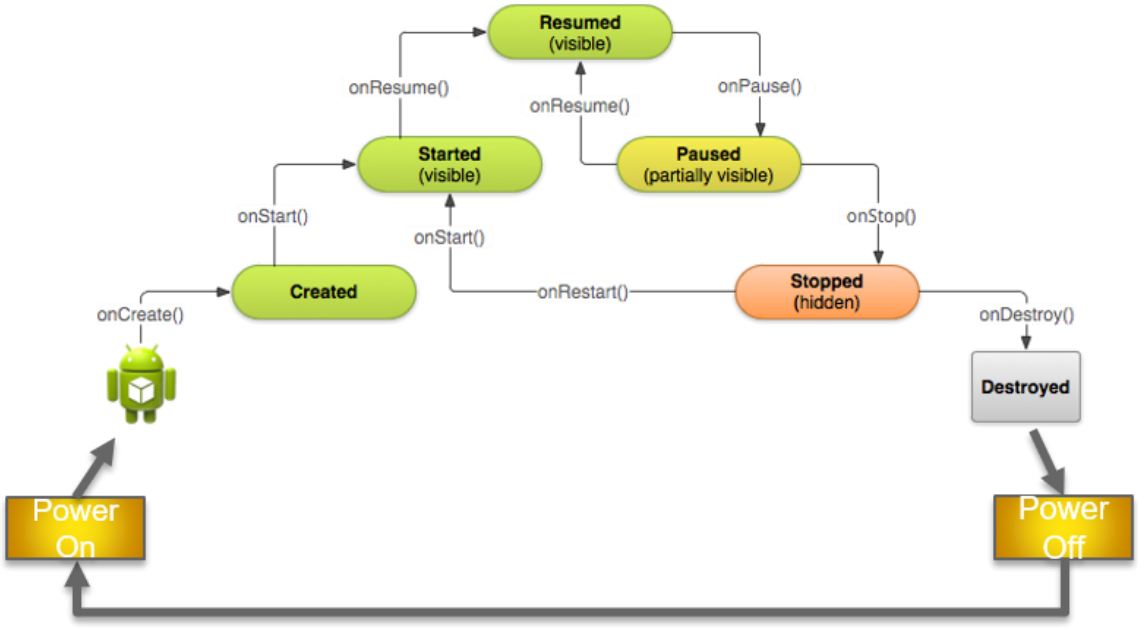

Figure 20. A flow chart to show the Android Lifecycle. At each stage in the lifecycle, we obtain hardware information and develop a log. For example, the onDestroy() function is called when the user terminates our application, which is recorded in our log and is valuable information when evaluation errors in our application.

Some users would claim our mobile application was not working and we did not know how to analyze what was happening. After the hardware info module was implement and deployed, we found out the users were causing a lot of the errors. For example, a user would claim the application had a survey prompt problem and the application was not administering the surveys. This resulted in a low compliance rate for this particular user. After investigating this user's hardware info, we found out this user was turning the device off over half of the day.

Another user said they had completed the surveys but we were not seeing them on the website, thus this user had a low compliance rate. After investigating this user's hardware info, we found that this user did not have an internet connection because they worked in a big building with a basement and their mobile device did not have a cellular connection. This resulted in the surveys not being sent to the server. The list goes on for why hardware info was a very useful

module when monitoring and fixing our application but is too long to discuss every scenario. The following example in Figure 21, is an example of the output from hardware info sampled at every five minutes.

```
                Tue Sep 29 04:50:39 CDT 2015

Is Phone Charging? -> true
  Charging By: AC Outlet
Battery Level: 1.0
Network Connection Status: Connected
  MOBILE: CONNECTED
  WIFI: UNKNOWN
  BLUETOOTH for Network: UNKNOWN
  Is Phone Connected To Active Network? -> true
GPS Mode in Phone Settings: LOCATION_MODE_SENSORS_ONLY
  Is There an Active GPS Signal? -> true
  Longitude and Latitude: 38.88794724, -92.34900453
  GPS Provider: gps
  GPS Accuracy: 17.0 meters
  Is the GPS Accuracy Good? -> true
  Will this GPS location be recorded? -> true
Is BLUETOOTH for Device Supported? -> true
  Is BLUETOOTH for Device On? -> false
Airplane Mode Is On: false
---------------------------------------
```

Figure 21. An example of our mobile ambulatory assessment system's (mAAS) Hardware Information output sampled every five minutes. Some of the important attributes sampled were battery level, network connection, GPS settings, Bluetooth support/status, as well as airplane mode status.

Attributes sampled every five minutes included: 1) whether the phone was charging, 2) how it was charging, 3) the battery level, 4) network connection status, 5) how it was connected to the network, 6) is the network active (as determined by pinging Google), 7) the GPS settings, 8) active GPS signal, 9) GPS accuracy, 10) good or bad GPS accuracy, 11) Bluetooth support and status, as well as 12) Airplane Mode status. These attributes are important to determine if the user is changing some of the settings for the phone. If they did change some of the settings, this

would explain why the data was not reaching the server or why they did not have a network connection. The five-minute sample would be recorded on the device and the recording would also be sent to the server. That way if the mobile device did not have an Internet connection then we would still have record of the log stored on the mobile device. Once the user brought the equipment back to the lab, we would pull all the data off the phone to make sure it matched the data on the server.

There were various attributes recorded asynchronously. Figure 22 below shows what was recorded asynchronously and how these things are useful to debug our application. Asynchronously collected attributes included: 1) whether the user turned the mobile device on/off, 2) Bluetooth on/off, 3) the user started/closed the app, 4) the Bluetooth pairing state and which device the mobile device is connecting to, and 5) if the user changed the system clock including the previous time and the current time. To determine the previous time of the clock after the user changed the clock was not an easy task. In some instances, the user would be changing the system clock in order to skip over surveys during certain times of the day. Some users would say their physiological data is not being collected because the device would not connect via Bluetooth, when in reality they had Bluetooth turned off. The user would say the app is not working when they powered the device off or they closed our application. Therefore, the hardware info was the most helpful source of knowledge when trying to determine what is going on with the mobile application.

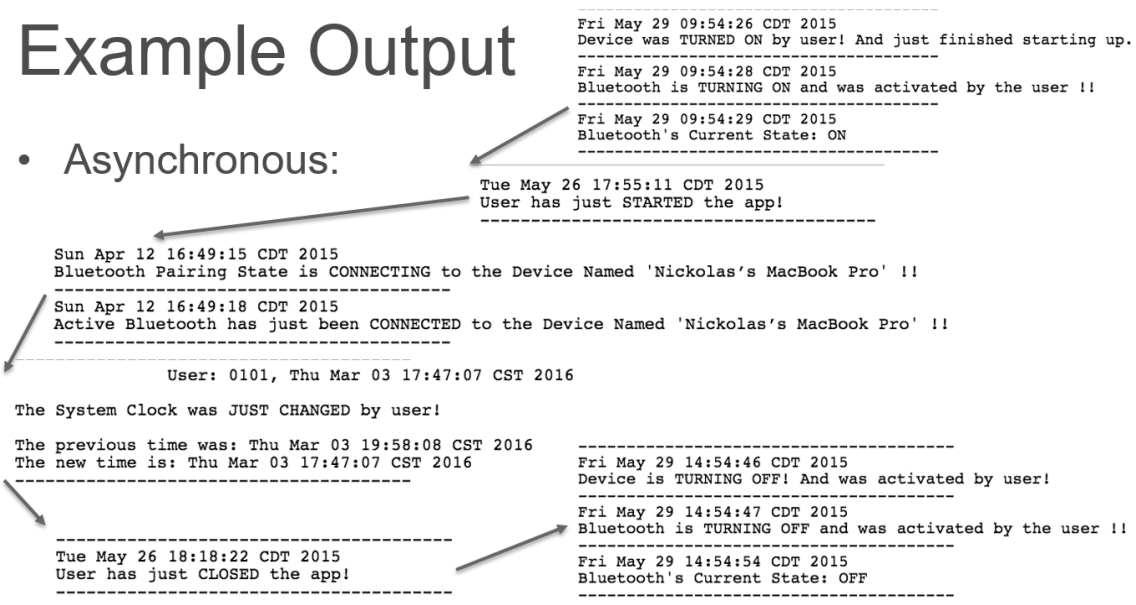

Figure 22. Example output of the asynchronous attributes collected by our mobile ambulatory assessment system's (mAAS) Hardware Information. These attributes are valuable data to debug our application and to determine if users are causing some errors.

4.3 System Improvement: Uploading Missing Survey Data

When the device did not have an Internet connection, a problem occurred where the data stored on the phone would be different than the data on the server. This was because the phone did not try to re-upload the data on an HTTP post error. Once the user was done with the study, they would return the equipment including the mobile device that was given to them. Once we received the mobile device back, we would plug it in to a computer and compare the data on it with the server. This was a timely task considering the data on the mobile device was encrypted. Therefore, to solve this problem, I implemented the Uploading Missing Data Module. Below in Figure 23, we demonstrate what would happened and the solution from the Uploading Missing Data Module.

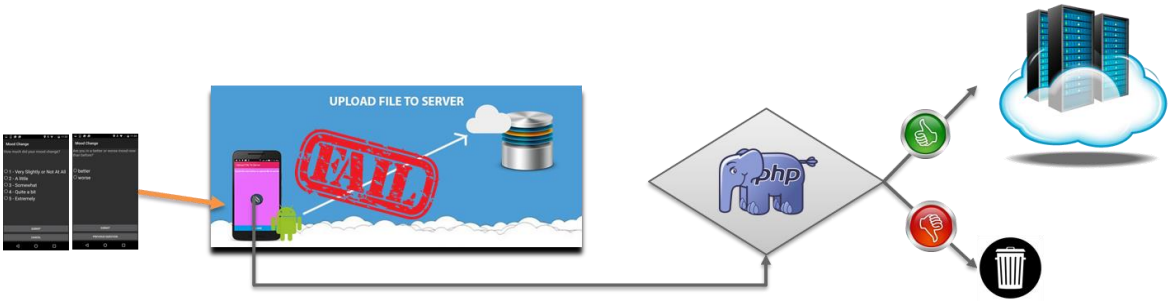

Figure 23. A flow chart of the Uploading Missing Data Module. When the phone failed to upload data to the server, the uploading missing data module was implemented to solve this problem. After the user returned the phone, there was a button under the administrator settings that could be pressed to upload missing data. Then a server program sorted through the data to determine which data was missing and where it needed to be inserted in the correct place.

The Uploading Missing Data (UMD) module was implemented as follows. There was a button under the administrator settings, only accessible by username and password, where the administrator could activate. Once activated, the UMD module would take all the data stored on the mobile device, only the data created by the application, and send it to the cloud. On the cloud server, a program would sort through all the data and determine which data had already been uploaded. The data that had not been uploaded previously would be taken, decrypted, and stored in the proper place. The server program was written in PHP and the data was stored in the file system and in a database. After the UMD module was implemented, we did not need to check the data manually, which saved us a lot of time.

4.4 System Improvement: Obtaining Hexoskin Physiological Data

When we implemented the mAAS system with other sensors, we were able to obtain the data from the sensors on the phone and then take the data from the phone and upload it to our cloud server. Our study went with a new sensor called Hexoskin because it was known for having high accuracy and the sensors were comfortable to wear. With Hexoskin, this functionality of sending data directly from the phone to our server was not available and the data from the sensor had to be uploaded with the Hexoskin application independently. Once the data was uploaded by the

Hexoskin application, it went to Hexoskin servers where pre-processing was done on the sensor data. From Hexoskin servers, we had to go to Hexoskin's dashboard and download data records one by one for each user. This took way too much time and thus caused a problem.

Our research team developed a solution to solve this problem. We developed a program that would implement Hexoskin's Application Program Interface (API). Through Hexoskin's API, we were able to get all the records for all the users automatically. The program was written in Java and ran on the server in order to store the physiological data in the same place as the Hardware Information data and the Survey Data. The program would check records already downloaded, search for new records, compare and find the differences, then download the new records from Hexoskin's server and store them in the correct place. Figure 24 below show the flow chart of obtaining the Hexoskin data. The upload from the phone directly to our server is not possible, therefore the data is sent to Hexoskin's server and obtained through the Hexoskin API.

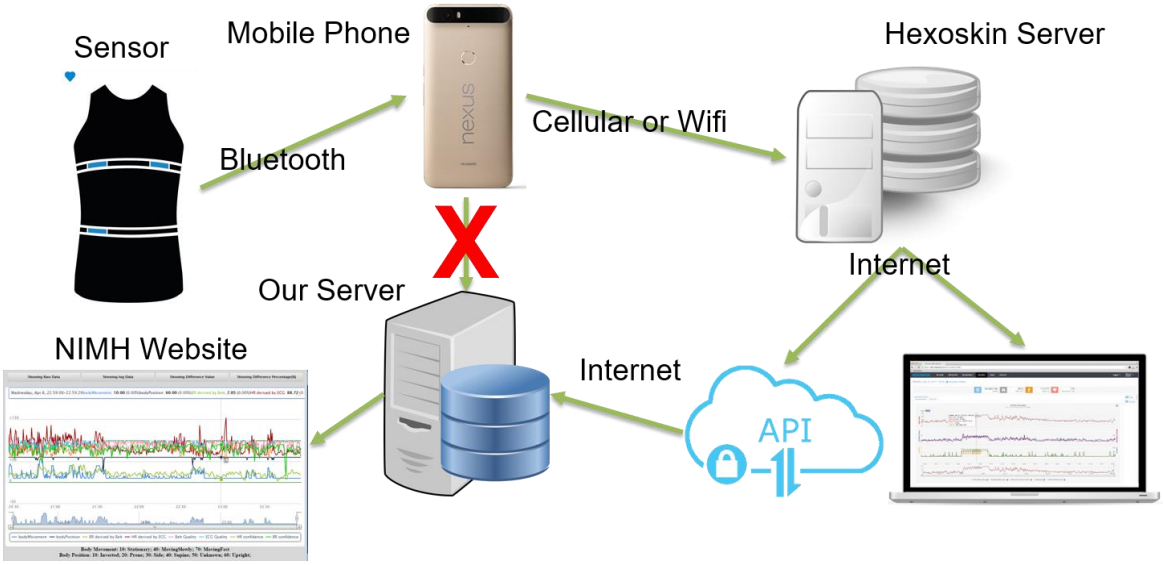

Figure 24. A flow chart of obtaining Hexoskin data. The upload from the phone directly to our server is not possible, therefore the data is sent to Hexoskin's server and obtained through the Hexoskin Application Program Interface (API). Before this module, we had to obtain each record individually and manually through the Hexoskin Dashboard, which was not very efficient since each user may have several records per day.

Instead of running the program several times, we wrote the program to be on a timer. The timer was set to run twice a week to download the new data and was constantly running in the background of the server. When the program was not downloading new data, no resources were used. Each physiological measurement was stored in separate files on the Hexoskin server and could only be retrieve through the API separately. Therefore, an extension of the program was written to download all the physiological measures consisting of multiple files and once the download was finished, the program would take each of the files and combine them based off the timestamp of the data. Since all of the measurements were sampled at 1Hz, the combination was possible without losing valuable data. After the data was combined, a separate Comma-Separated Value (CSV) file was generated with all the data and this file was used in AMD's machine learning pipeline.

5. AMD DESIGN AND IMPLEMENTATION

Machine Learning Analysis of Mood Dysregulation (AMD) can be thought of as a pipeline, consisting of many components, each component performs an operation on the data and then AMD will perform the machine learning and data analysis. There are many components such as selection, combination, preprocessing, transformation, machine learning, and interpretation or evaluation as shown in _. However, to make things more clearly understandable for the reader, AMD will be broken down into many components and discussed in this chapter.

The first main component is the data selection and storage module which selects the valid users and stores them for further processing. Next, the data combination module which does pre-processing and combines the sensor and survey data for further processing. Next the data cleaning module which cleans the data using Loess to find outliers two standard deviations away from the curve. Continuing, the data smoothing module which smooths sensor and survey data using regression imputation. We then created features and performed feature extraction methods in order to scientifically determine which features produce the most weight for determining mood dysregulation. All of these modules are executed automatically.

After these modules, we perform a machine learning prediction module and then analyze the data. The prediction will take the preprocessed data and then perform machine learning techniques on the data, all done automatically. Machine learning models are built, trained, and tested on the data. We build several models and compared the accuracy and efficiency between them. We then analyze the results obtained from the models and try to gain some knowledge from the data. Results will be obtained and data visualization will be provided in order to help researchers to make discoveries and find some meaningful interpretations from all the machine learning results.

The AMD pipeline has been implemented in Matlab and the machine learning techniques are implemented using Waikato Environment for Knowledge Analysis (WEKA). Data visualization can be done using Matlab or WEKA. In the future, machine learning algorithms that have been implemented will be integrated into AMD using Matlab based on the discoveries found and the goals of the project. This research plans to have the results of the machine learning algorithms to automatically give feedback to the user on the mobile device, in order to better understand the current mood dysregulation episode happening to the subject. Once we can predict the mood dysregulation from the physiological data with satisfying results, we can obtain more accurate self-report data based off the prediction from the machine learning algorithms.

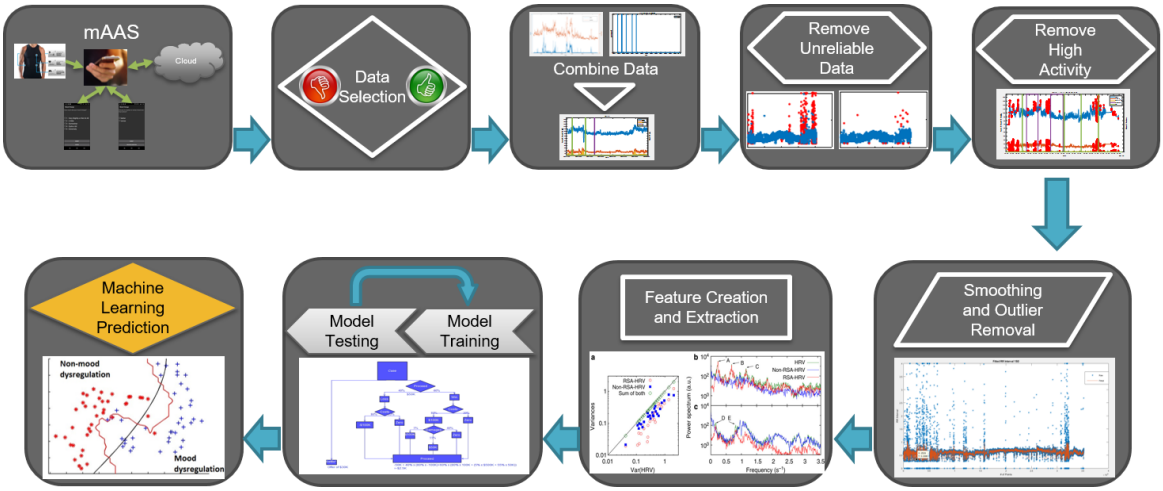

Figure 25. Detailed workflow of Analysis of Mood Dysregulation (AMD's) pipeline. The environmental data is collected through our mobile ambulatory assessment system (mAAS). Data selection and storage module selects the valid users and stores them for further processing. Data combination module pre-processes and combines the sensor and survey data. Next the data cleaning module cleans the data by removing unreliable data and using Loess to find outliers two standard deviations away from the curve. Continuing, the data smoothing module smooths sensor and survey data using regression imputation. We then created features and performed feature extraction methods. After these modules, we perform a machine learning prediction module and then analyze the data. The prediction will take the preprocessed data and then perform machine learning techniques, all of this is done automatically.

Since subjects are not able to distinguish mood dysregulation clearly by themselves, the machine learning algorithms will predict mood dysregulation from the physiological data for the participants, and then prompt the user with a real-time report automatically to accurately get

information from the user of the current mood dysregulation episode. Having this done, we can collect the mood dysregulation episode in real-time and psychologist can understand each subject more clearly. Next, the feature selection methods will be discussed in detail in the next section 5.1. AMD's main components will be discussed in this chapters subsections 5.2 and 5.3.

5.1 Feature Selection and Computation

This research selected and considered features that were usually reported in the literature as the best features for distinguishing mood dysregulation. In addition, we worked with psychologist and physiologists for identifying the best and new features which are not as extensively investigated in the literature. Together, we selected the features that were determined to be the most distinguishing either from previous literature, visually on graphs and plots, or when used in the machine learning classification. There are three categories of the most common datatypes available from the Hexoskin: 1Hz Data, Raw Data, and Asynchronous Data. The available feature from the 1Hz Data are heart rate, breathing rate, minute ventilation, and activity. Raw data has the available features of electrocardiogram (ECG) at 256Hz, thoracic respiration, abdominal respiration both at 128Hz, acceleration X, acceleration Y, acceleration Z all three at 64Hz. The available features from the Asynchronous data are steps, RR interval, inspiration detections, and expiration detections. Next, the details of the selected features will be discussed in section 5.1.1 and 5.1.2. Note: Although all the above features were collected, not all of the features available were used in the machine learning analysis, such as the raw data from ECG. These features can be analyzed in future work to determine the effects of adding these features and what the response of the physiological data will be from the additional features.

5.1.1 Heart Related Features

For heart related features, electrocardiogram (ECG) is the key for all resources here. From the ECG values, we detect the QRS complex, the peak that represents a heartbeat. QRS is a name for the combination of three of the graphical deflections seen on a typical ECG [47]. The QRS complex is usually the central and most visually obvious part of the tracing and corresponds to the depolarization of the right and left ventricles of the human heart [47]. The Q wave is any downward deflection after the P wave [47]. An R wave follows as an upward deflection and the S wave is any downward deflection after the R wave, hence the name QRS complex [47]. Figure 26 below shows a standard human heart beat and depicts what a QRS complex is.

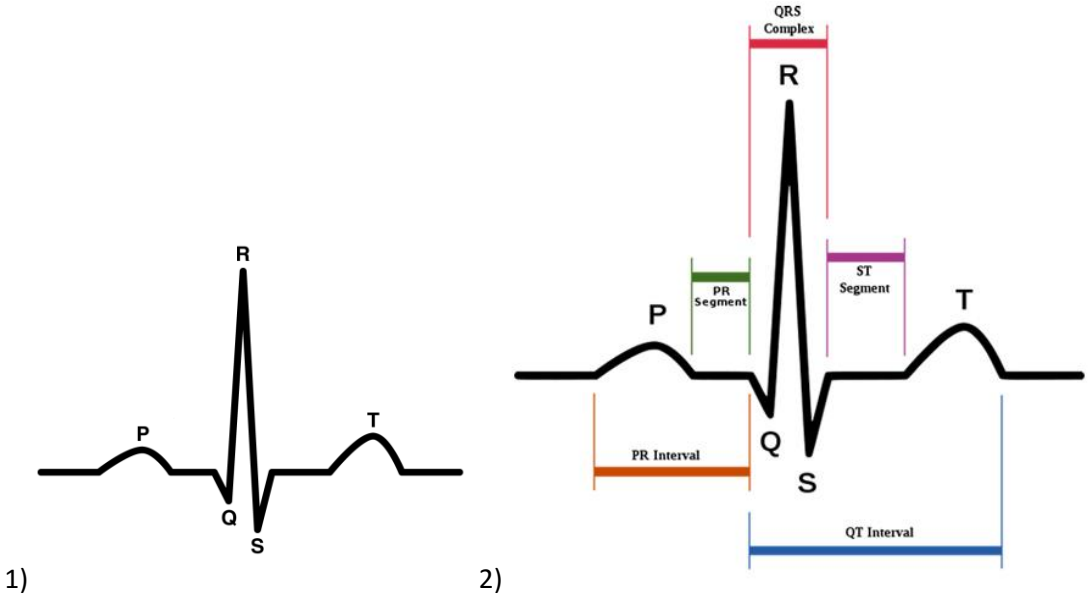

Figure 26. Two examples of human heart beat. 1) Standard electrocardiogram (ECG) wave. 2) Same ECG wave with QRS complex, PR interval, and QT interval schematic representation of normal ECG.

The time interval between these QRS complexes is calculated by measuring the RR interval. RR interval is the time duration between successive R peaks in an ECG signal or sequential QRS complexes. RR intervals correspond to the duration between heart beats, thus can be used to calculate several statistical features describing the behaviors of the heart, such as heart rate.

An example of RR interval is shown in figure 24 below. Several preprocessing steps are taken to prepare the data for training which will be further discussed in the next section 5.2. Hexoskin provides a special channel for heart rate called heart rate status, which contains information on the perceived quality of the data. The values is returned as an integer, however the bits is what needs to be interpreted, as each bit represents a potential flag [48]. Hexoskin uses these flags to know if the QRS detected are reliable or if they should be ignored. As such, there might be discrepancies between each detected QRS and the returned heart rate, if some of the flags have been set [48].

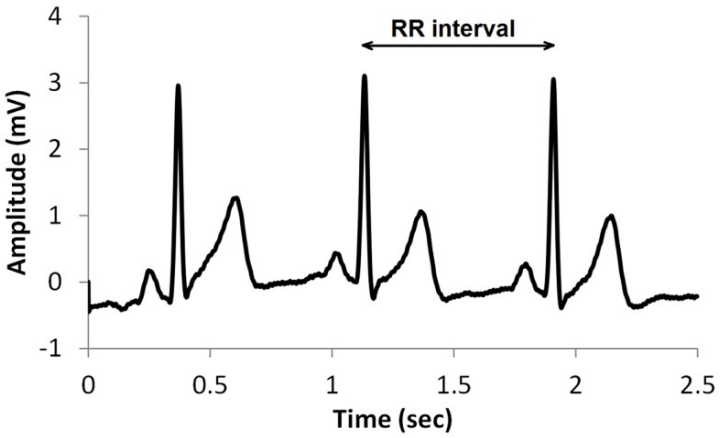

Figure 27. Example of RR intervals. An RR interval is the duration between two successive R peaks in the ECG signal or sequential QRS complexes.

There are several reasons why the heart rate status bits may be set including: A good solid reliable reading in top condition and ready for interpretation, 50_60Hz, saturated, artifacts, unreliable RR, or disconnected. Figure 28 shows an example when the ECG reading is in top condition and ready for interpretation.

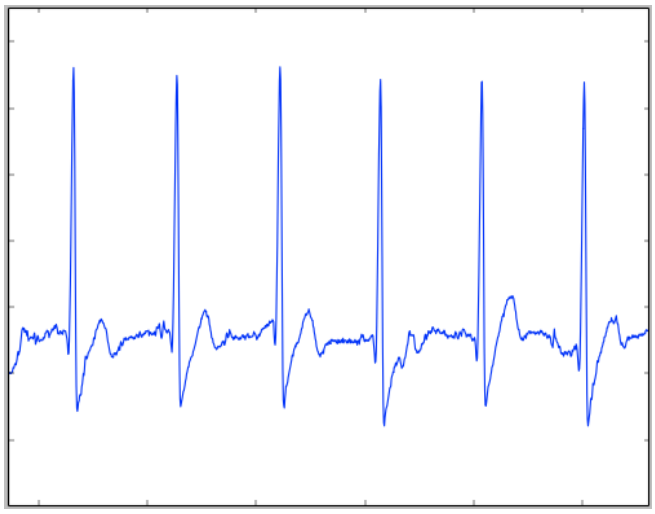

Figure 28. Normal Hexoskin ECG reading. This reading is in top condition and ready to be interpreted. The X axis represents time and the Y axis represents the amplitude of the ECG reading.

Below will describe some of the situations and give examples of what the ECG looks like when each kind of flag is set. If the disconnected flag is set, the person might not be in the shirt but the sensors are on. Maybe the processing mote is turned on but the shirt is not connected for this situation [48]. It is also possible the mote does not have a good connection with the sensors as show in Figure 29 below.

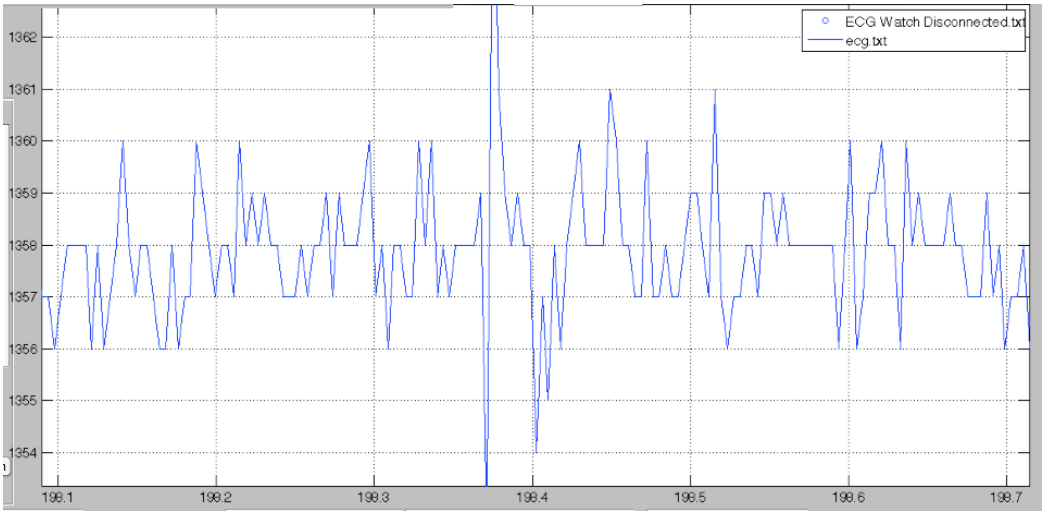

Figure 29. ECG disconnections looks like the following. Note the scale, it can be seen the variations are extremely small when compared to the normal ECG reading.

If the 50_60Hz flag is set, there may be a presence of a significant amount of 50 or 60Hz noise in the signal [48]. Figure 30 shows when the 50_60Hz component is manageable and the reading is still useable by using the QRS-detection algorithms. This algorithm can filter out this component and the QRS will be detected correctly.

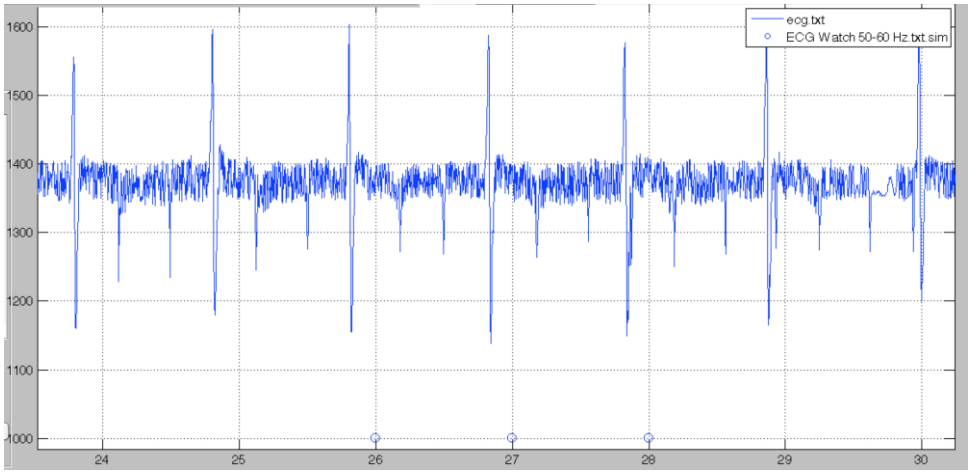

Figure 30. Hexoskin ECG reading with manageable 50-60Hz noise component. Data collected here is stored but may not be used in the machine learning prediction.

However, in the following case, the 50-60Hz component is too strong and important, therefore even heavy filtering cannot get rid of the noise as show in Figure 31.

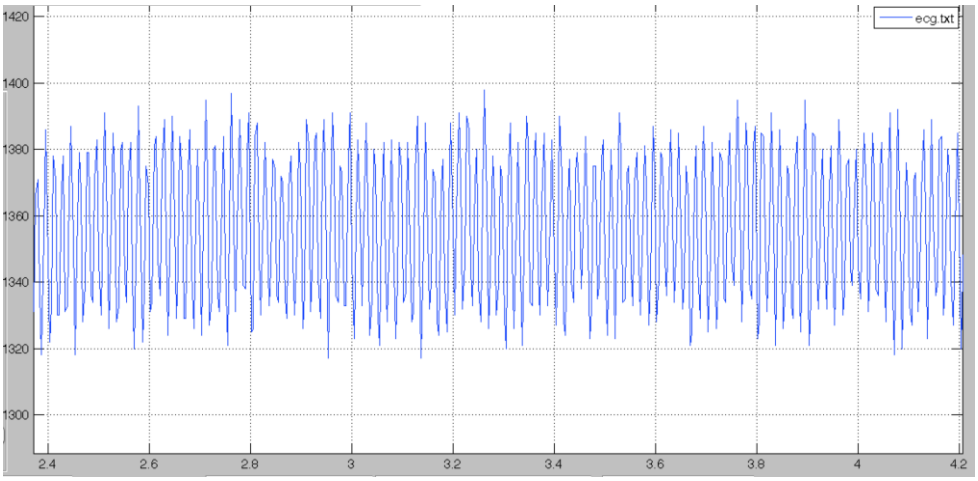

Figure 31. Hexoskin ECG reading where the 50-60Hz noise component is too strong, even heavy filtering cannot get rid of this noise. This data is not used in the machine learning prediction.

If the saturated flag is set, then maybe the signal intensity goes beyond the dynamic range [48]. Spontaneous saturation looks like the following, depicted in Figure 32. This can happen if one or more of the shirt's electrodes stop making contact with the skin for a few moments. The next QRS might be missed, however the QRS detection algorithms will recover quickly [48].

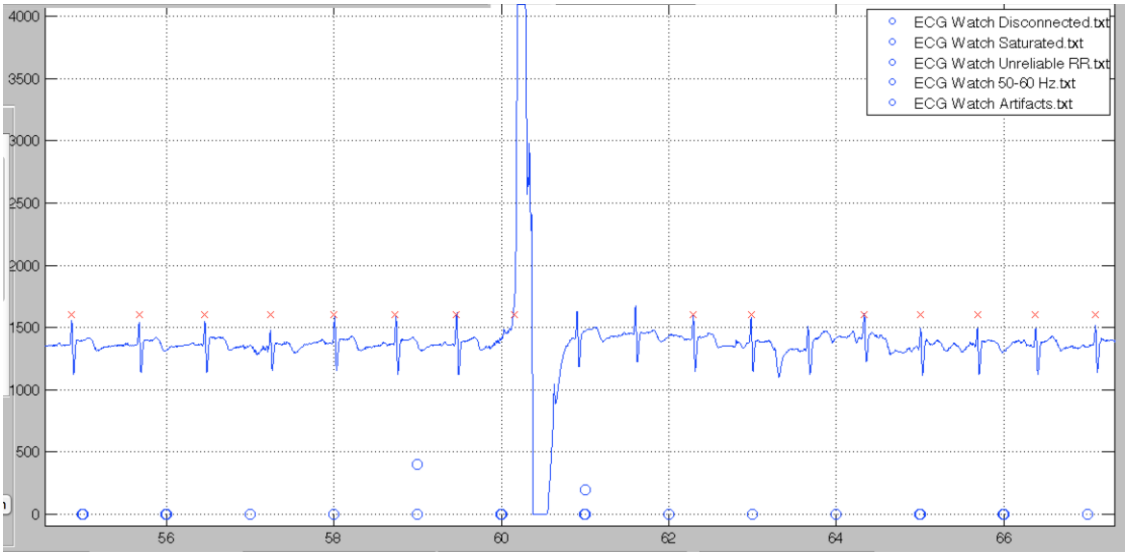

Figure 32. Hexoskin ECG reading where spontaneous saturation happens. This can happen if one or more of the shirt's electrodes stop making contact with the skin for a few moments. The next QRS may be missed, however the QRS detection algorithm will recover quickly.

However, in the following case, the saturation is too strong in which the algorithm will not be able to detect anything reliably, as shown in Figure 33.

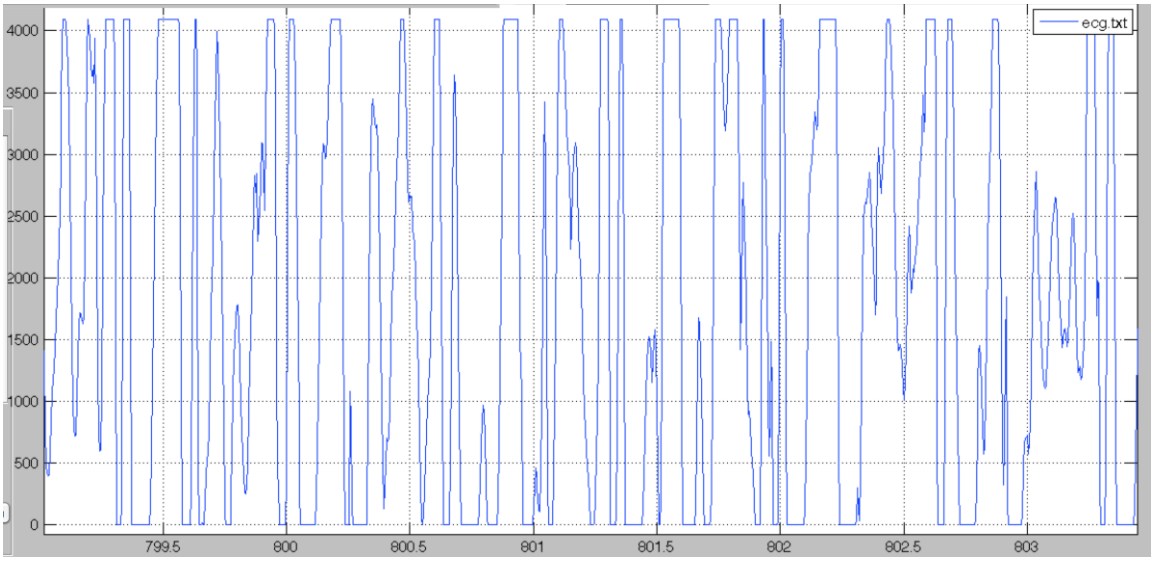

Figure 33. Hexoskin ECG reading where the saturation is too strong for the QRS detection algorithm to detect anything reliably. This data is not used in the machine learning prediction.

If the artifacts flag is set, then any movement artifact may be detected. Movement artifacts look like the following, shown in Figure 34, the red X's represents QRS detections. As you can see in Figure 34, this kind of noise can be interpreted as a QRS, although it will not always be the case, depending on the spectral components of the noise [48]. This is represented in the middle of the figure. It can be seen there is a sharp peek in which the sensor was bumped but the QRS detection algorithm thinks it is a standard QRS. Therefore, if a QRS is detected while the movement artifact flag is set then this should be interpreted with caution [48].

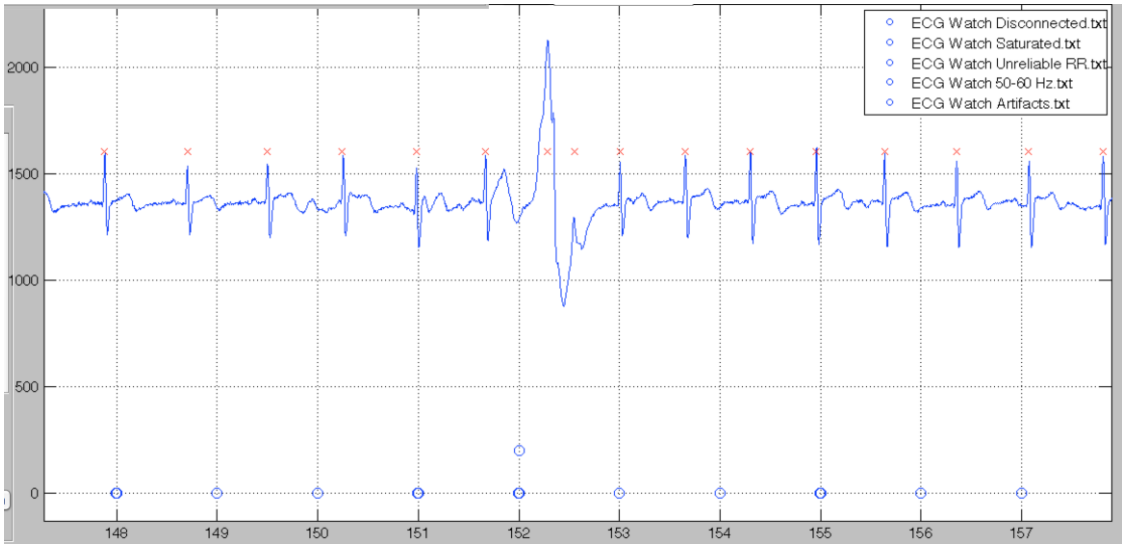

Figure 34. Example of movement artifacts in the Hexoskin ECG reading. The red X's represent QRS detections. As it is shown, sometimes this kind of noise can be interpreted as an actual QRS, although it will not always be the true case, depending on the spectral components of the noise.

If the unreliable RR flag is set, it may be because the RR interval seems suspiciously unreliable which happens either when the signal quality is low and QRS are not detected correctly, or when tachycardia's or brachycardia's are present [48]. The algorithms expect a certain regularity of the QRS intervals, therefore if a detected QRS would result in a suspicious RR interval, like in the 9th and 10th detected QRS, the unreliable RR flag is set [48]. This type of situation is shown below in Figure 35.

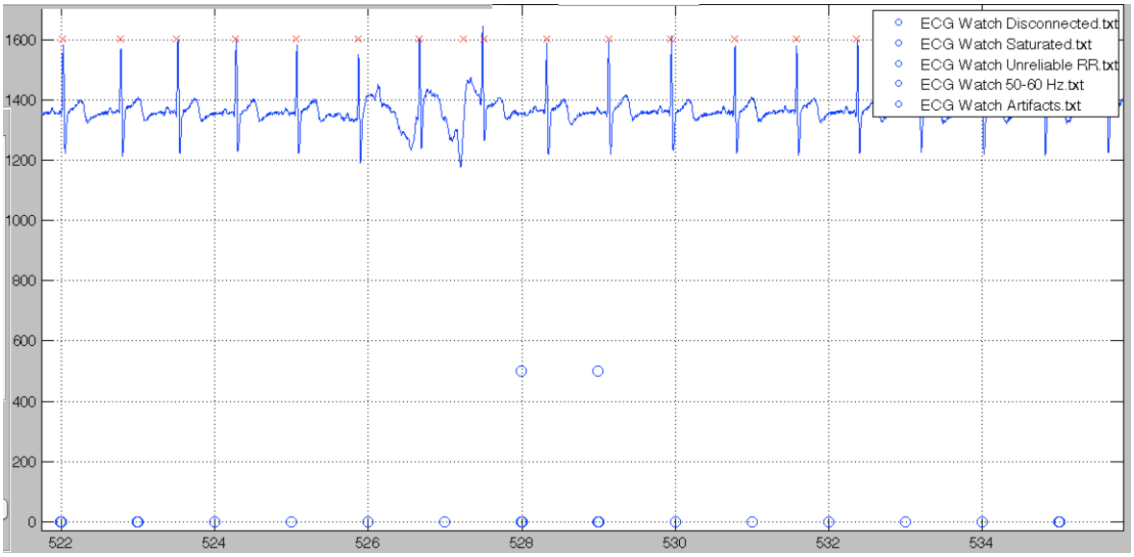

Figure 35. Example of suspicious RR interval in the Hexoskin ECG reading where the red X's represent the QRS detection. Note the 9th and 10th detected QRS shown in the figure, this is caused by some artifact and is not an actual suspicious RR interval.

Sometimes, this can happen due to some artifact link in the previous figure, however it can be caused by actual suspicious RR intervals, as in the sample shown below in Figure 36.

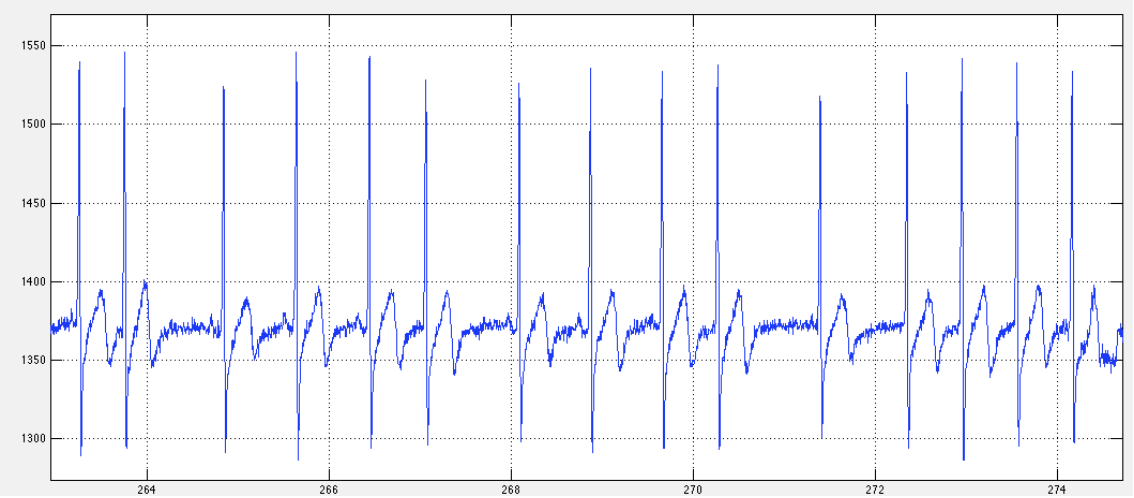

Figure 36. Hexoskin ECG reading where the unreliable flag is set due to actual suspicious RR intervals. Note between the 2nd RR interval, the 6th RR interval, and the 10th RR interval, all examples of actual suspicious RR intervals.

5.1.2 Breathing Related Features

Each respiration cycle is composed of an inhalation and an exhalation period and the computation of the respiration features involves the identification of each cycle. Thus, a respiration cycle starts from a valley which corresponds to the start of an inhalation phase and ends at another valley that identifies the start of the next inhalation [2]. From the raw respiration sensor measurements, the detection of inspirations and expirations is detected, which can then be used to detect breathing rate and other various features. Hexoskin respiration raw data can be seen in Figure 37, the x axis corresponds to time and the y axis corresponds to the ADC value.

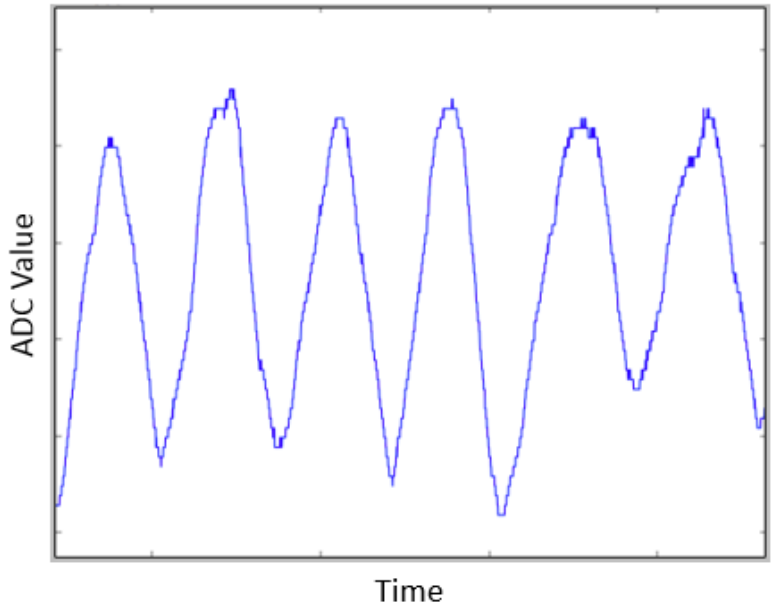

Figure 37. Example of raw respiration data from the Hexoskin sensors. The first valley corresponds to the start of the inhalation phase, this reaches a peak and then starts to decline which is the exhalation phase. This decline ends at another valley, which marks the start of the next inhalation. A respiration cycle starts from a valley and ends at another valley.

In total there were 6 different features computed from the respiration signal including inspiration, expiration, breathing rate, tidal volume, minute ventilation, and breathing rate status. 1) Inspiration duration corresponds to the time elapsed from a valley of a signal to the next peak

denoting the maximum expansion of the chest in the respiration cycle [2]. 2) Expiration duration corresponds to the time duration between the peak and the next valley. Respiration duration is the sum of the inhalation and exhalation duration, i.e. the duration of a breath [2]. 3) Breathing rate is simply the number of breath cycles per minutes [2]. 4) Tidal volume is the lung volume representing the normal volume of air displaced between normal inhalation and exhalation when extra effort is not applied [48]. In a healthy, young human adult, tidal volume is approximately 500mL per inspiration or 7mL/kg of body mass [48].

Tidal volume is derived from the difference between the inspiration and expiration values. 5) Minute ventilation is the volume of air inhaled (inhaled minute volume) or exhaled (exhaled minute volume) from a person's lungs in one minute [2]. Minute ventilation is calculated by taking the average tidal volume and multiplying that value by the breathing rate. 6) Breathing rate status is provided by Hexoskin, which is a special channel for the respiration signal containing information on the perceived quality of the data. The value is returned as a hexadecimal value, however the bits is what needs to be interpreted, as each bit represents a potential flag. Hexoskin uses these flags to know if the respiration cycles detected are reliable or if they should be ignored.

5.2 AMD Machine Learning Pipeline

The data preprocessing module consists of many sub-components including selecting, combining, cleaning, and smoothing sensor and survey data automatically. The survey data was collecting using out mobile ambulatory assessment system (mAAS) in [20]. The mAAS system is a smartphone-based mobile ambulatory assessment system for psychology research. mAAS provides real-time data monitoring and collecting for real-life subject behavioral and psychology data, as well as physiological data. The physiological data was obtained using Hexoskin Wearable Body Metrics Smart Shirts which is a commonly used consumer wearable sensor. The Hexoskin

connects to the smartphone via Bluetooth and the smartphone will upload the data to the cloud. From the cloud, the user can view and download the data using Hexoskin's dashboard. Hexoskin has an API where we can make server programs to obtain the data automatically. Once we obtain the data from Hexoskin's servers, we can then perform AMD's Machine Learning Pipeline. Next, the data selection and storage module will be discuss in further detail.

5.2.1 Data Selection and Storage

For all of the participants that have completed the mood study, consisting of approximately two weeks of sequentially collecting physiological data and self-report mood data, were considered for the machine learning pipeline. Some participants had low compliance rates, meaning the mobile application prompted them to answer surveys but they did not respond or answer them. Another possibility is the participant did not wear the physiological wearable sensor as they were assigned to. Therefore, these participants' data were not inserted into the machine learning model.

Other participants did not complete all days for the mood study, did not obtain valuable physiological data, or other various reasons, all of these subjects were not considered. For non-valuable physiological data from the Hexoskin wearable body metrics shirt, sometimes the sensor would obtain static electricity which would cause the ECG not to respond correctly, causing poor quality heart and breathe metric values. This is a known issue among the wearable physiological sensors community. To fix this problem, you simply wet the sensor slightly in order to remove the static electricity. If you do not remove this static electricity, all of the data can have low scientific value. This happens to people with overly dry skin. To determine if the samples have low value, the Hexoskin provides two attributes to calculate the value of the electrocardiogram (ECG)

reading and another to calculate the value of the respiratory inductance plethysmography (RIP) reading.

The data was saved for further processing for participants meeting all the follow requirements: must have 9 out of 14 days of self-report data, must have at least one mood dysregulation sample, must have at least one mood example with corresponding sensor data, and must have valuable sensor data determined by ECG status and RIP status calculated by the sensor. During the time of my research, there were a total of 22 participants' data to use. Now there is probably more. Out of the 22, only 16 were used due to one or more of the previous conditions not being met.

In a flat-file system, all of the data being used was approximately 10GB of storage. Therefore, the data was relatively large. The flat-file system architecture was poor because the file names were not good and hard to read by a human. Also, each of the metrics were in separate files. Activity was in its own csv file, heart rate was in a separate csv file, RR interval was in its own csv file, and so on. Java code was written to combine all these files together by the timestamp which had the efficiency of $O(n^2)$ because it was in a nested for loop. This did not take very long to run but it is not good computer science practice to have an n^2 efficient algorithm, therefore I decided to put all of the data into a database. The filenames and separate files for each metric was not an issue once all of the data was in the database.

Getting the data into the database was not trivial. The structure of the directories was very poor, the filenames were difficult to read because they were generated automatically by a computer, the combined sensor files were difficult to locate, and some users were split into two different files. An example can be seen in Figure 38 below. The first number is the user, the second number is the record UID, then the year, month, day, hour, minute, second, and then millisecond

when the record was created. You can see in these two examples, there exist a user with an ID of 1011 and 1011b. Even though the user ID is different, they actually represent the same user. Therefore, special code was written to combine these two user IDs into one.

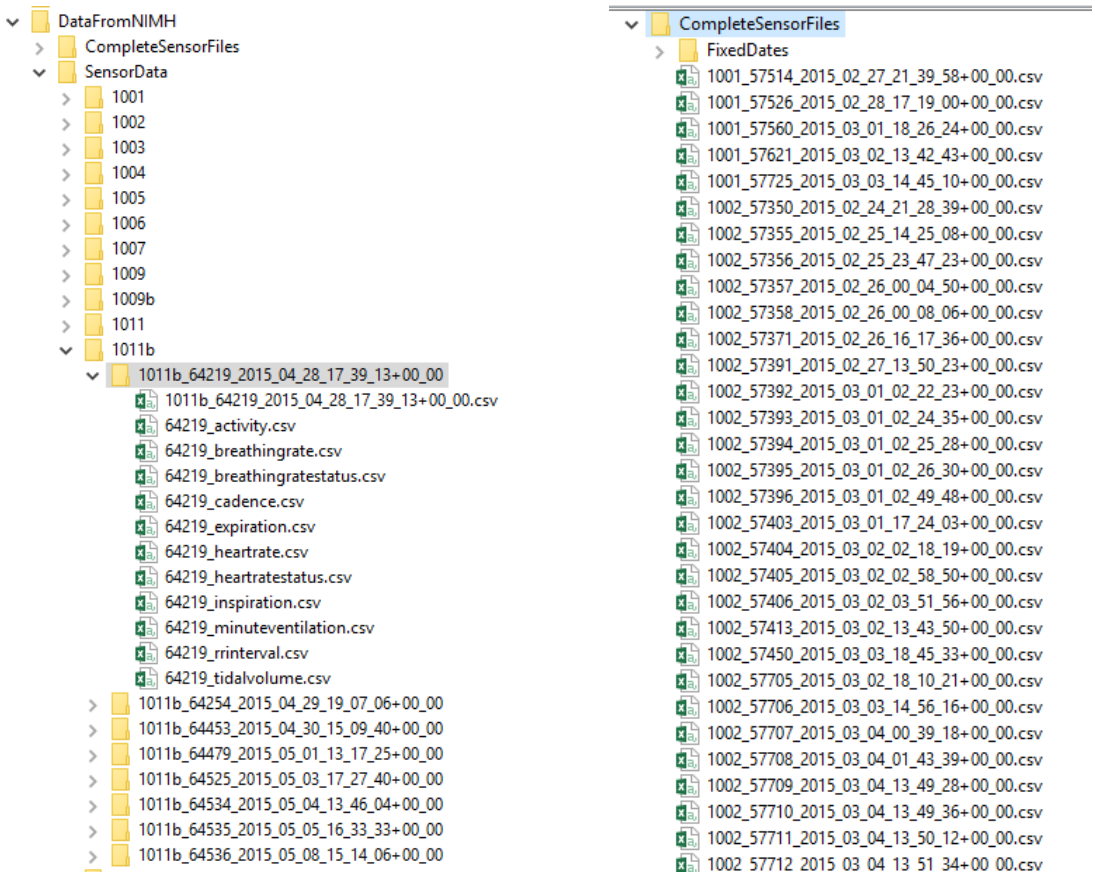

Figure 38. Examples of the file structure in the flat-file system. The directory structure is poor and is hard to read by humans since it was generated by a computer. Therefore, a program was written to take all this data and insert it into a database.

Instead of searching for the file in each folder by hand, I wrote python code to find all the combination sensor files and put them into a new directory. At first I thought this would be difficult. However, after looking into the problem, I noticed the combination files all had a unique character in the filename. The character was a “+” sign, representing the milliseconds the file was created. Therefore, I used a regular expression to search for filenames with a “+” sign in them and move them into a newly created directory.

Next, the default date from Hexoskin was difficult to insert into a database. Here is an example of the date format: “Fri Feb 27 21:40:36 UTC 2015”. Therefore, I wrote another program to convert all the dates. Figure 36 below shows the old date and the new date after the program handles the format problem. I tried to use the data directly from a file in Matlab, however the data was too large to load entirely into memory. Therefore, I must insert into a database and connect that database to Matlab. After all the attributes and columns were correct, I needed to load all the data into the database. To do this, I wrote PHP code which read each combination file in the directory, made in the previous step, and insert the data line by line.

	A	B
1	TimeStamp	Athelete
2	Fri Feb 27 21:39:59 UTC 2015	1001
3	Fri Feb 27 21:40:00 UTC 2015	1001
4	Fri Feb 27 21:40:01 UTC 2015	1001
5	Fri Feb 27 21:40:02 UTC 2015	1001
6	Fri Feb 27 21:40:03 UTC 2015	1001
7	Fri Feb 27 21:40:04 UTC 2015	1001
8	Fri Feb 27 21:40:05 UTC 2015	1001
9	Fri Feb 27 21:40:06 UTC 2015	1001
10	Fri Feb 27 21:40:07 UTC 2015	1001
11	Fri Feb 27 21:40:08 UTC 2015	1001
12	Fri Feb 27 21:40:09 UTC 2015	1001
13	Fri Feb 27 21:40:10 UTC 2015	1001
14	Fri Feb 27 21:40:11 UTC 2015	1001

	A	B	C	D	E
1	TimeStamp	Athelete	activity	breathingrate	breathingratestat
2	Feb 27 2015 21:39:59	1001	0	10	
3	Feb 27 2015 21:40:00	1001	0	10	
4	Feb 27 2015 21:40:01	1001	0	10	
5	Feb 27 2015 21:40:02	1001	0.0234375	10	
6	Feb 27 2015 21:40:03	1001	0.015625	10	
7	Feb 27 2015 21:40:04	1001	0.01953125	10	
8	Feb 27 2015 21:40:05	1001	0.015625	13	
9	Feb 27 2015 21:40:06	1001	0.015625	21	
10	Feb 27 2015 21:40:07	1001	0.015625	29	
11	Feb 27 2015 21:40:08	1001	0.015625	32	
12	Feb 27 2015 21:40:09	1001	0.01953125	32	

Figure 39. An example of the incorrect date format for MySQL. The picture on the left shows the incorrect date format from the raw data. The picture on the right show the date after the program handles the format problem.

Next, the Survey Data was missing some of the attributes in the combination file, however the missing data was in the separate raw data files. One example is the end timestamp when the user completed the surveys. How do you fix this, should I go through each survey data file by hand and fix over 500 dates for each user? This answer is no when you are a program. Anything that can be done manually can also be done by a program much faster. I manually did around two or three of the dates by hand, in order to determine the procedure, and then figured out how to write a program to do this for me. The Java program would go through each line of the

combination survey data file, parse the line, find the missing attributes, find the raw data file, if it exists then open it, find the corresponding attributes, re-write the parsed line to the combination file, then go to the next line to see if there were any missing attributes. Figure 40 below shows an example of the missing End Time Stamps (EndTS).

Type	ScheculedTS	Reminder1	Reminder2	Reminder3	StartTS	EndTS
RSSchedule	2/11/2015 13:33					
RSSchedule	2/11/2015 15:58					
RSSchedule	2/11/2015 17:14					
RSSchedule	2/11/2015 18:26					
RSSchedule	2/11/2015 20:07					
RSSchedule	2/11/2015 21:45					
missed MR						
undo MD						2/17/2015 16:26
RS3	2/17/2015 16:53	2/17/2015 16:53			2/17/2015 16:53	3
RS4	2/17/2015 19:15	2/17/2015 19:15			2/17/2015 19:15	4
RS5	2/17/2015 20:56	2/17/2015 20:56			2/17/2015 20:56	5
Bedtime	2/18/2015 9:00				2/17/2015 21:00	2/17/2015 21:01
RSSchedule	2/18/2015 11:06					
RSSchedule	2/18/2015 12:49					
RSSchedule	2/18/2015 15:57					
RSSchedule	2/18/2015 17:15					
RSSchedule	2/18/2015 20:14					
RSSchedule	2/18/2015 21:23					
MR	2/18/2015 9:00	2/18/2015 9:00			2/18/2015 9:00	2/18/2015 9:01
RS1	2/18/2015 11:06	2/18/2015 11:06			2/18/2015 11:06	1
RS2	2/18/2015 12:49	2/18/2015 12:50			2/18/2015 12:50	2
RS3	2/18/2015 15:57	2/18/2015 15:57			2/18/2015 15:58	3
RS4	2/18/2015 17:15	2/18/2015 17:15			2/18/2015 17:15	4
RS5	2/18/2015 20:14	2/18/2015 20:14			2/18/2015 20:15	5
Bedtime	2/19/2015 7:00				2/18/2015 21:00	2/18/2015 21:00

Figure 40. An example of the missing data in the survey files. The highlighted cells are missing the end time stamp (EndTS) for that particular survey. A program was created to determine which data was missing, find the corresponding raw data file based off the date, find the corresponding raw data, and then insert it into the combination file to fix the problem in over 500 files.

Once each line had been parsed I would output the result to a new file. This code took some time to write, I had to use the User ID and then convert the date in order to find the raw data file, then parse the entire raw data file to find the corresponding attributes. Once the code was finished, it fixed all the files in less than 10 seconds. So the combination of the time to think about how the procedure for the code, to write the code, and the execution time was much shorter than

doing the work manually. In addition, if someone wants to continue my work, this code can be reused for new data that is also missing attributes in the combination file.

In total, there were over 8.5 million sensor data records in the database. After inserting the data into a database, I combined all of the different sensor metrics together into one table. Once all the metrics were together, based off their timestamp and athlete values, I also inserted the survey mood data into its own table. The data combination procedure will be discussed in the proceeding section, 5.2.2 Data Combining.

5.2.2 Data Combining

The sensor data from Hexoskin and the survey data from our study were from two different and completely separate sources. However, when running the machine learning pipeline and when analyzing the data, these separate data sources need to be combined into one source. There were over 8.5 million sensor data records in the database. There were approximately 160 mood dysregulation labels in the survey data. After combining the data, only 5601 sensor records matched or corresponded with the survey data. Therefore, most of the survey data did not line up with the sensor data. Some of the associated reasons were: The sensor was not worn by the user during the time of the survey, the sensor data collected during the time of the survey had a low confidence rate, during the time of the survey the sensor did not have a good connection, and many other possible reasons.

To combine the data, each sensor record had a timestamp associated with it. Each sensor record corresponded to one second in the day. Each survey record had a start timestamp and end timestamp which corresponds to the time the user started and completed the survey. The average time the user took to complete the survey was approximately three minutes. Therefore, the code I wrote would take the start and end time of each mood dysregulation survey for a particular user

and determine if there were corresponding sensor records between these two times. If there were, the code would mark mood dysregulation as true for all the sensor records between these two times.

After combining the data, more subjects were crossed off the list, meaning I could not use their data. The main reason was because no survey data corresponded to the sensor data. Before combining the data there were 16 users that were being analyzed, after combining the data only 10 subjects were being analyzed, because 6 users did not have any corresponding survey data with sensor data. There were 8.5 million sensor records and after combining the data there were only 5601 sensor records that corresponded with the survey data records. This is less than a 1% overlap between the two data sources. Next data cleaning will be discussed in section 5.2.3.

5.2.3 Data Cleaning

Several preprocessing techniques are used to prepare the data for training machine learning algorithms. However, the raw data itself looked pretty good on graphs and other data visualization methods. Hexoskin performs data cleaning automatically on their server and when collecting the data on the sensor. Therefore, even though the term raw data is being used here, technically it has been pre-processed already from Hexoskin. For example, the heart rate is averaged over the last 16 beats in order to calculate the current heart rate for that second in time. Breath rate is averaged over the last 7 respiration cycles in order to calculate the current breath rate for that second in time. Activity is calculated by (accX ^ 2 + accY ^ 2 + accZ ^ 2) ^ 0.5 and high passed at 2.65Hz on the 3 axis independently. Activity is also averaged over the last second to measure the G units for that particular second in time. However, even though this processing is done, further processing and cleaning need to be performed in order to compare results with the

raw data given from Hexoskin. Below in Figure 41 are some raw data examples from subject 1009 on April 18.

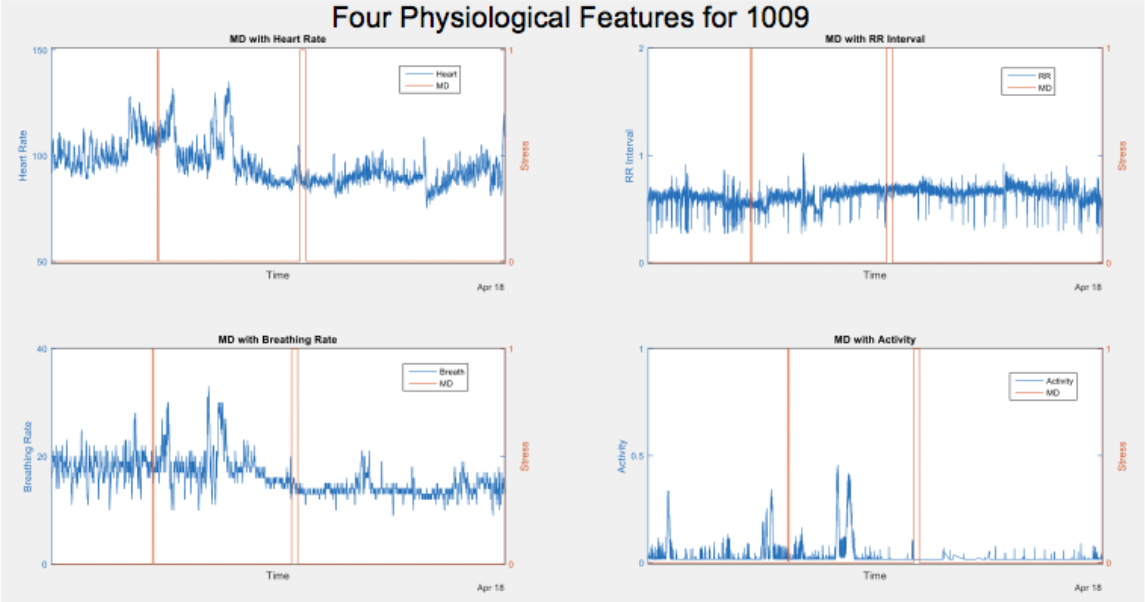

Figure 41. Examples of raw data from one subject in the study. It is shown how well the raw data is without any cleaning performed from our research. However, Hexoskin does some preprocessing on this data before we receive it.

The data cleaning model can be broken down into a pipeline itself. First we take the raw data: Time stamp, activity, breathing rate, breathing rate status, cadence, expiration, heart rate, heart rate status, inspiration, minute ventilation, RR interval, and tidal volume. We perform a locally weighted scatterplot smoothing (LOESS) to fit a regression model on the data. This is done to find outliers two standard deviations (2SD) away from the mean of each minute. Each outlier found will be flagged and ignored in the training of the machine learning models. Two standard deviations away from the curve was recommended in the literature and was thoroughly tested in this pipeline. One standard deviation was also tested in this pipeline, however the final results were lower because of the loss of valuable data, and therefore 2SD was implemented in the finalized pipeline. After outliers are detected and ignored, the base features are normalized in

order to help wide between-person differences before computing any statistical features from the dataset.

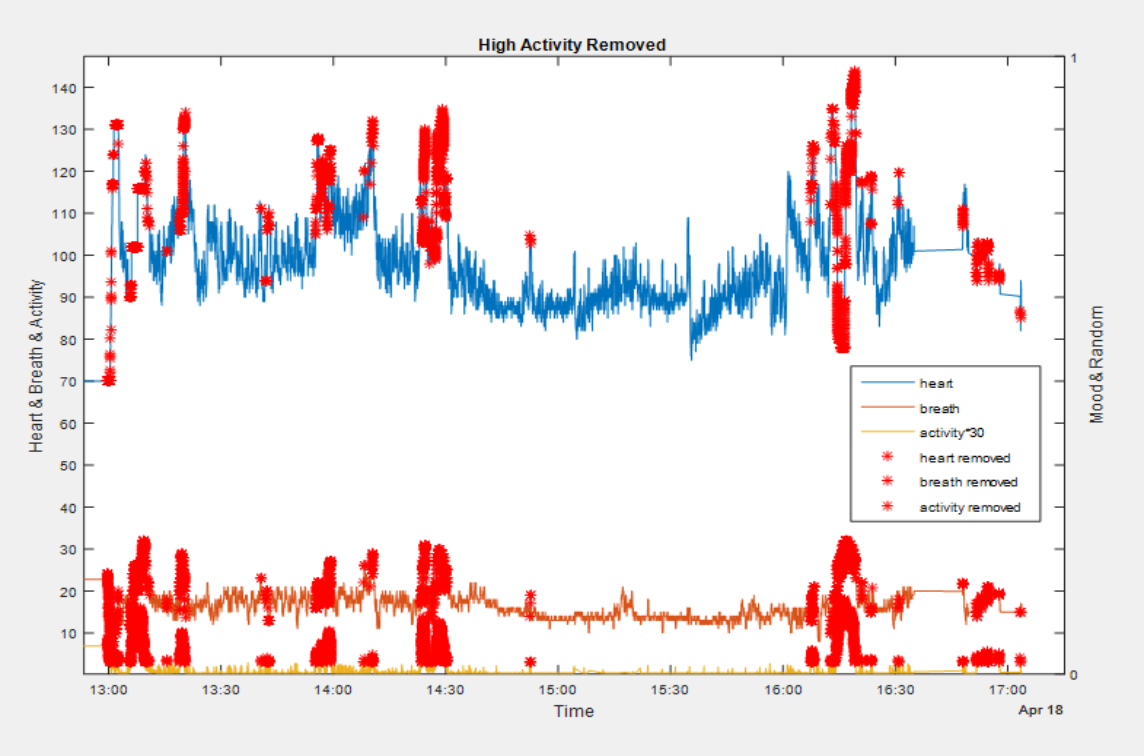

Figure 42. Example of removing high levels of activity. The literature suggests high levels of activity mask mood dysregulation which can mislead the machine learning algorithms. Therefore, data collected during high levels of activity are removed.

Next, data consisting of concurrent high levels of activity were removed because physical activity including motion can overwhelm the sensors and physiological response to mood dysregulation. Removing high levels of activity can be seen in Figure 42. In other research such as [2], they remove two minutes following physical activity since they found that the physiology returns to baseline within two minutes after activity [2]. It is even reported that high activity levels will mask mood dysregulation, causing the machine learning models to have more false positive prediction values to mood dysregulation, which is further reason to remove high activity levels [2]. Proceeding, unreliable and non-valid heart and breath rate were removed from the data since

attributes cannot be reliably computed for these minutes of the data. Below in Figure 43 shows a visual representation of the cleaning pipeline thus far.

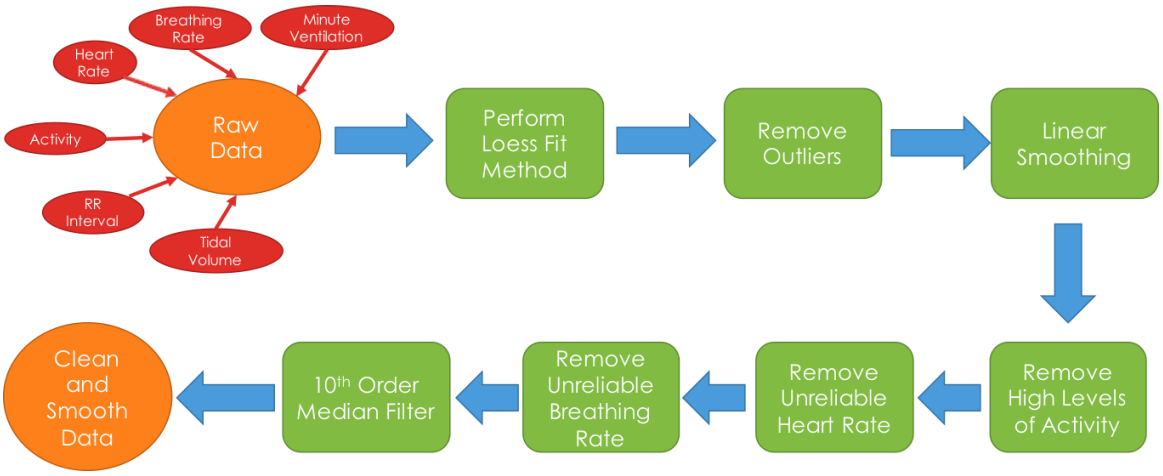

Figure 43. Visual representation of the first cleaning pipeline, a.k.a. Naïve Pipeline. This pipeline adds an additional unnecessary median filter module which smooths the data too much loosing valuable data.

From the figure, we can see the cleaning pipeline needs to add an additional median filter after removing unreliable data in order to remove some of the outliers. However, after the median filter, much of the valuable data is lost which means the cleaning pipeline needs to have further research performed in order to find out the correct order. There were many tests performed on the median filter module in order to remove the outliers which remained after Loess and the removal of unreliable data. The default value for N, N=3, was applied and the outliers remained. Next, an N value of 10 was tested but the outliers that needed to be removed still remained. Proceeding, N=100 was tested and failed. Finally, the outliers were removed after N was equal to 400, which removed most of the data available. Figure 44 below shows a visual representation of this process to see why this pipeline was not very well designed.

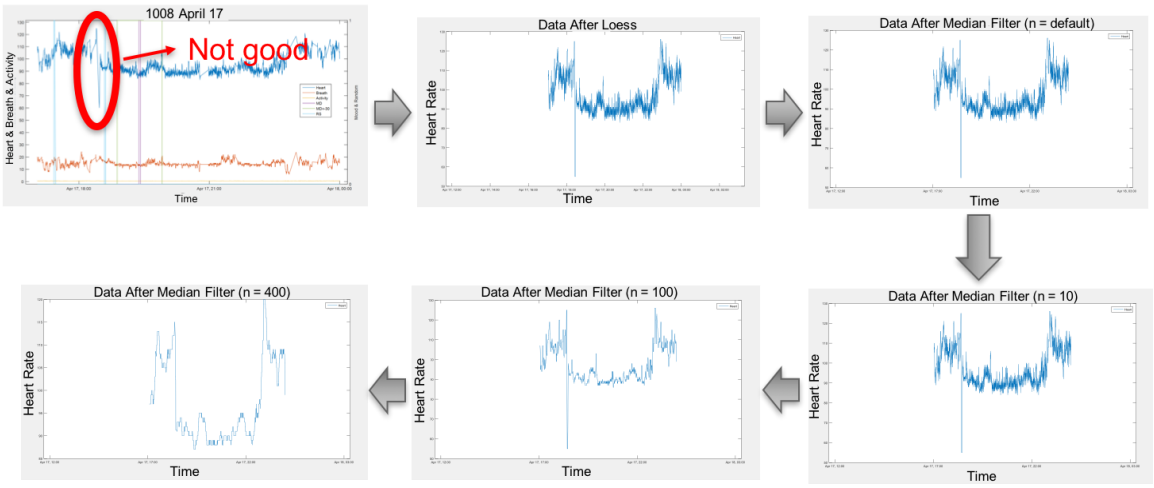

Figure 44. A visual representation of the results from the "Naïve Cleaning Pipeline." It is shown that the outliers in the top left graph is not removed until N=400 on a median filter. However, this removes most of the valuable data and thus is not a good approach.

After these results, a re-construction of the entire cleaning pipeline took place. A logical approach was taken, by reordering the modules. Instead of removing the unreliable data from heart and breath rate after Loess, these two modules were re-order to the beginning of the cleaning pipeline. So the order of the new cleaning pipeline follows: First, we take the raw data from the physiological measurements and remove the unreliable ECG measurements. This can be determined by a heart rate status attribute presented by the hardware of the Hexoskin sensor. Next, the same approach was taken by removing unreliable RIP measurements. After removing all the unreliable data, approximately 20.63% of the data was removed.

Proceeding, high activity levels were removed and approximately 8.78% of the data is removed at this model. After, a Loess Model is made to make a linear fit for the data and any data 2SD away from the curve is removed. Approximately 11.27% of the data is removed at this module. The total percentage of the data removed from the cleaning pipeline is 40.68%, approximately 30% fewer than some of the related work presented in [2] research. Figure 45 shows a visual of the final cleaning pipeline.

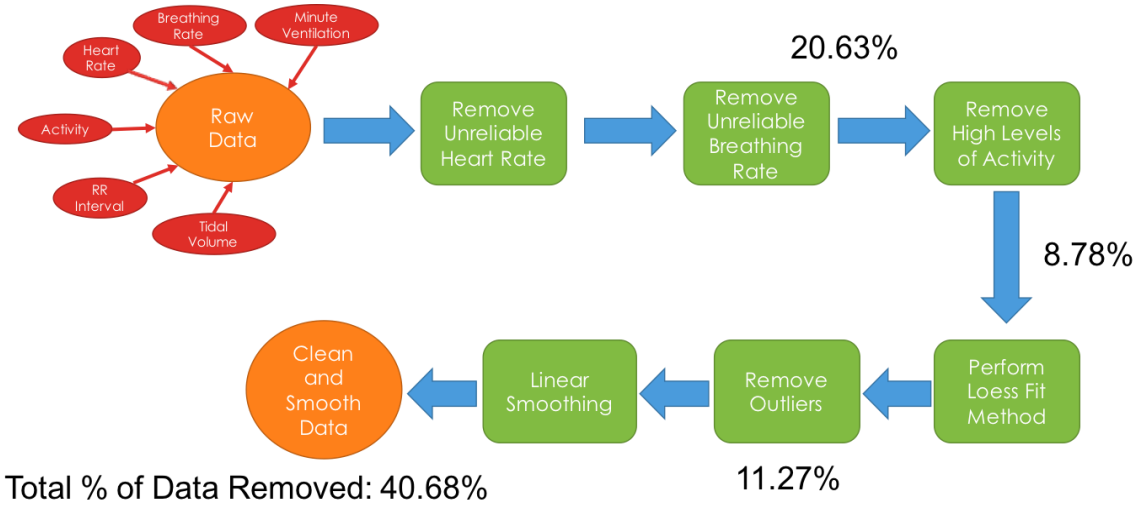

Figure 45. A flow chart representing the Final Cleaning Pipeline. Compared with the naïve cleaning pipeline, this pipeline does not need a median filter module, due to a better ordering of the entire pipeline and each of its modules. After removing unreliable data, approximately 20% of data was removed. After removing high levels of activity, approximately 9% of data was removed. After removing outliers, approximately 11% of data was removed. In total, approximately 40% of the data was removed.

Compared with the Naïve Cleaning Pipeline, it can be seen there is no need for a median filter module. Moreover, the ordering of the modules has changed, which increased the cleaning pipeline results and caused the median filter module to be unnecessary. In Figure 46 below, the same data was used in the naïve cleaning pipeline with the same outliers. In the final cleaning pipeline results, after removing the unreliable data, high activity, and then performing Loess, the outliers are removed. Comparing the naïve cleaning pipeline results and the final cleaning pipeline results, all of the valuable data remains in the final cleaning pipeline which is why it is implemented.

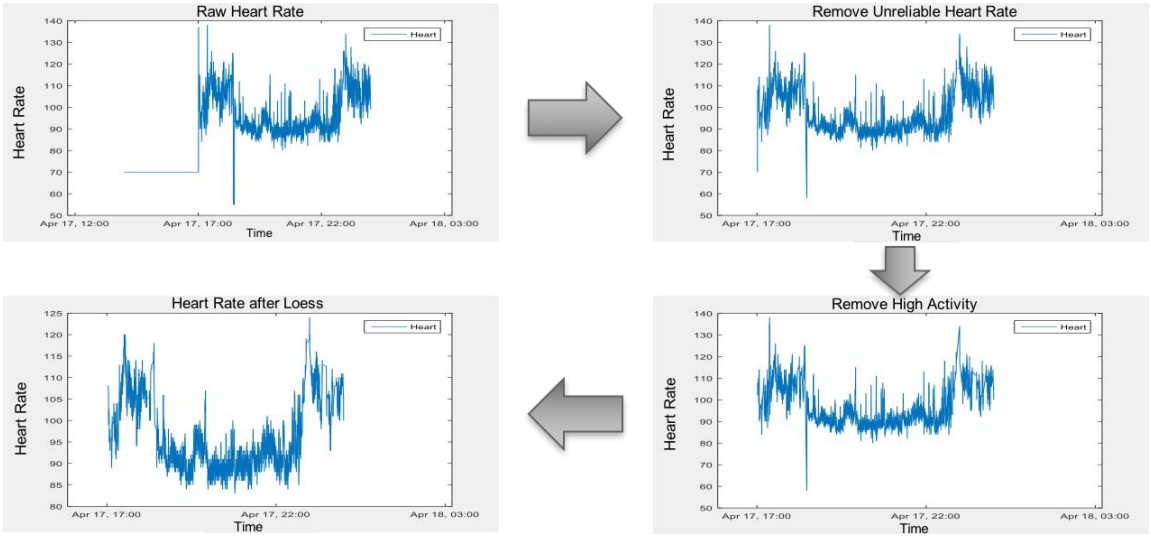

Figure 46. A visual representation of the Final Cleaning Pipeline Results. It is proven that the median filter module is unnecessary and all of the valuable data remains with this approach.

5.2.4 Data Smoothing

With the remaining data from the sub-pipeline, the Data Cleaning Pipeline, a locally weighted scatter plot smoothing (LOESS) model is implemented to fit smooth surfaces to the remaining data. From William Jacoby's research in 2000, he states Loess is a powerful but simple strategy for fitting smooth curves to empirical data [49]. Jacoby proceeds by saying the term "loess" is an acronym for "local regression" and the entire procedure is a fairly direct generalization of traditional least-squares method for data analysis. Loess is nonparametric, in the sense that the fitting technique does not require an a priori specification of the relationship between the defendant and independent variables [49]. Although it is used most frequently as a scatterplot smoother, loess can be generalized very easily to multivariate data [49]. There are also inferential procedures for confidence intervals and other statistical tests [49]. Therefore, for everything listed above, loess is a useful tool for data smoothing in AMD's pipeline.

The smoothing process is considered local because, like the moving average method, each smoothed value is determined by neighboring data points defined within the span [50]. The

process is weighted because a regression weight function is defined for the data points contained within the span [50]. In addition to the regression weight function, a robust weight function can be used to make the process resistant to outliers. Finally, the method are differentiated by the model used in the regression, loess uses a quadratic polynomial [50].

The local regression smoothing process follows these steps for each data point [50]:

1. Compute the *regression weights* for each data point in the span. The weights are given by the tricube function shown below:
    a. $w_i = \left(1 - \left|\frac{x - x_i}{d(x)}\right|^3\right)^3$
    b. Where x is the predictor value associated with the response value to be smoothed. x_i are the nearest neighbors of x as defined by the span, and $d(x)$ is the distance along the abscissa from x to the most distant predictor value within the span. The weights have these characteristics:
        i. The data point to be smoothed has the largest weight and the most influence on the fit.
        ii. Data points outside the span have zero weight and no influence on the fit.
2. A weighted linear least-squared regression is performed. For loess, the regression uses a second degree polynomial.
3. The smoothed value is given by the weighted regression at the predictor value of interest.

If the smooth calculation involves the same number of neighboring data points on either side of the smoothed data point, the weight function is symmetric. However, if the number of neighboring points is not symmetric about the smoothed data point, then the weight function is not symmetric [50]. Note that unlike the moving average smoothing process, loess span never

changes. For example, when you smooth the data point with the smallest predictor value, the shape of the weight function is truncated by one half, the left-most data point in the span has the largest weight, and all the neighboring points are to the right of the smoothed value [50].

After loess is used to fit a smooth model on the data, the residual and outliers are found from the data. The difference between the observed value of the dependent variable (y) and the predicted value $(\hat{y})$ is called the residual (e). Each data point has one residual [51].

a. Residual = Observed value – Predicted value

b. $e = y - \hat{y}$

Both the sum and the mean of the residuals are equal to zero. That is, $\Sigma\, e = 0$ and $\bar{e} = 0$. The residual is used to find the outliers of the data. If the data is 2SD away from the residual, then this data point will be removed. The symbol for standard deviation is σ (the Greek letter sigma). Below is the formula for standard deviation [52]:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

If the data point is removed, its value will be replaced with a null value in order to keep all data series the same length. However, null values are not allowed in the machine learning prediction model. Therefore, a one-dimensional (1D) linear interpolation is used in order to fill in the outlier values so each data series will be returned as the same length as the original input. The original loess regression model is used to with linear interpolation also known as regression imputation [53]. If this is not done, each data series will have different lengths, which will make it difficult to combine all the physiological data before it is passed to the machine learning algorithm. Linear interpolation is a basic way to fill in the "holes" in a series of data points [54]. If

two known points are given by the coordinates $(x_0\ ,\ y_0)$ and $(x_1\ ,\ y_1)$ then the linear interpolant is the straight line between these points [55]. For a value x in the interval $(x_0\ ,\ x_1)$ the value y along the straight line is given from the equation [55]:

$$\text{a)}\quad \frac{y-y_0}{x-x_0} = \frac{y_1-y_0}{x_1-x_0}$$

Solving this equation for y, which is the unknown value at x, gives the formula below [55]. The formula below is the formula for linear interpolation in the interval $(x_0\ ,\ x_1)$:

$$\text{b)}\quad y = y_0 + \frac{(y_1-y_0)}{x_1-x_0}\ (x-\ x_0)$$

This procedure is performed for all physiological attributes such as Activity, Breathing Rate, Heart Rate, Minute Ventilation, RR Interval, and Tidal Volume, all of which are a discontinuous time series of rapidly varying mobile sensor data. This procedure can be seen in _. In the next section, 5.2.5, the feature creation and extraction will be discussed in greater detail.

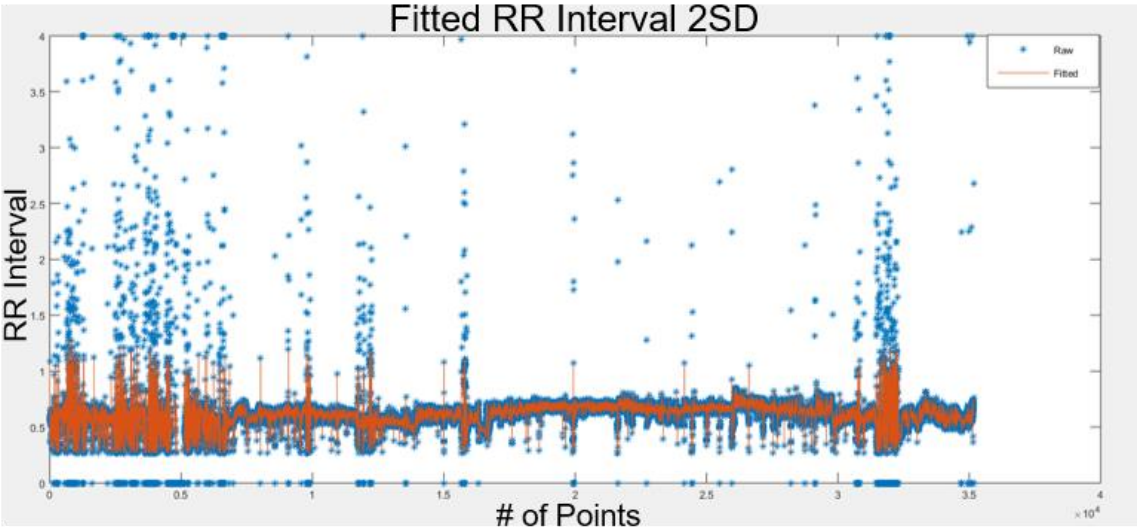

Figure 47. Data smoothing procedure outcome. The blue points are the raw that and the red line is the fitted data. After a locally weighted scatter plot smoothing (LOESS) model is implemented to fit smooth surfaces to the remaining data, the residual and outliers are found from the data that are 2 standard deviations away. Regression imputation is implemented to fill in the holes so all data series return the same length as the original input.

5.2.5 Feature Creation and Extraction

There were several experiments done to improve the machine learning model prediction. Based off previous literature [15], time is an important factor for mood dysregulation and stress. However, if we feed the Time Stamp attribute with the data into the machine learning algorithms, then the model will base everything off the timestamp, and the accuracy will be 99%. This is overfitting, thus the time stamp attribute cannot be used directly. In addition, if one model is going to be used across multiple users, an attribute with date and time will not work because most users are not in the study during the same date and time. However, if you do not consider time, then the accuracy will be very low approximately 70%. So, there needed to be some creative thinking on how to build machine learning models that considered time but did not have the time stamp attribute included.

There were four main ideas implemented and tested for creating time attributes. 1) To create a time category value. 2) Create a time only attribute which consisted of hour, minute, and second of day. 3) Create an hour only attribute which only consisted of the hour of the day. 4) Create an only hour and minute attribute which only consisted of the hour and minute of day.

The first idea of creating a time category value had some different approaches. The first approach was to create three categories for the entire day consisting of morning, afternoon, and night. Since there are 24 hours in a day and 3 categories, 24/3 = 8, so each time category should consist of 8 hour blocks. If we start at 12am, the morning time category would be from 12am to 8am, the afternoon category would be from 8am to 4pm, and the night category would be from 4pm to 12pm. This doesn't make sense, since we usually do not say 8am is in the afternoon and we usually do not say 4pm is night. Therefore, a shift in the time categories must take place. After shifting the time categories where morning starts from 2am and ends at 10am, afternoon is from

10am to 6pm, and night is from 6pm to 2am, we can determine this is also not a good fit. Usually we do not say 6pm is in the afternoon and 2am is in the night.

The second approach is to create 4 time categories in a day. If there are 4 categories for a 24 hour period then 24/4 = 6 and thus, each category must consist of 6 hour blocks. The 4 categories consist of morning, afternoon, evening, and night. Morning will start at 5am and end at 11am, afternoon will start at 11am and end at 5pm, evening will be from 5pm and end at 11pm, night will be from 11pm to 5am. By doing this each time category is evenly spaced and makes more sense to humans. The machine learning models do not like string representations for the time categories, therefore in the code, morning = 1, afternoon = 2, evening = 3, and night = 4.

The second idea was to create a time only attribute. To do this, a program was created which basically took the date time attribute and removed the date. For example a value of 2016-07-15 13:23:24, which is a standard SQL date time format, was transformed into 13:23:24. Even though at the beginning we thought this would be a good idea, the probability one patient is participating in the study the exact same time another subject is participating in the study is extremely low. Therefore, this idea was an extreme failure.

The state space for the machine learning algorithm was too large. There are 60 seconds in a minute, 60 minutes in an hour, and 24 hours in a day. If you multiple 60 * 60 * 24 there are 86,400 different states for this one particular attribute, not even calculating the other 10+ physiological attributes and all their possible values. When running the machine learning model, the program would crash every time. I even installed 8GB of additional ram, borrowed from a friend in the lab, calculating to a total of 20GB, increased the heap size to 16GB, and only using 1 subject and only 1 day for that subject, the program would still crash. This attribute was kept, however it was not used in the final analysis.

The third idea was to create an hour only attribute which consisted of the hour of the day. A program was created to remove the date, minute, and second from the time stamp. For example, 2016-07-15 04:20:00 would be transformed into 4. The benefits of this idea is it could be used across subjects, there was a high probability one subject participated in the study the exact same hour another subject participated in the study, there is a high possibility subjects have mood dysregulation during the same hour of the day as other subjects, and it is very likely that the same subject will have mood dysregulation at a similar time across days. This ideas was a success and was kept in the final analysis.

The fourth and last idea tried was to create an only hour and minute attribute. A program was created to remove the date and second from the time stamp attribute. For example, 2016-04-20 13:23:24 was transformed into 13:23. We realized this was not as likely for two subject to have mood dysregulation at the exact hour and minute of the day, however testing was done to scientifically determine if this would improve the results. Since the state space was not as large as the second idea, the program was able to run. There are 60 minutes in an hour and 24 hours in a day so there was 1,440 states for this particular attribute. However, when examining the final results, the training set was over fitted by this attribute, which was similar to running the regular timestamp attribute. The accuracy was 98% and the states were only created by this attribute. This attribute was kept but not used in the final analysis.

In addition to creating the previously mentioned features, we added a 30 minute window to the start and end time of each survey. This was because the start time and end time of the self-initiated mood dysregulation surveys were too short and did not include enough data for the prediction. Based off all the mood surveys used, the average time taken to complete a mood dysregulation survey was approximately three minutes and twenty one seconds (3:21). Each

sensor record corresponds to one second of the survey data because the sensor data is collected at a 1Hz frequency sampling rate after it is preprocessed by Hexoskin.

To try to analyze such a short time period, less than 3 minutes in some cases, would be difficult. Based off the psychologist and physiologist we were working with, it is good to look at least 30 minutes before the start of a mood dysregulation episode in order to see the sensor reading tendencies for any particular user. Therefore, a window size of 30 minutes before the start time of the survey was added and a window size of 30 minutes after the survey end time was also analyzed. Adding the 1 hour window size, 30 minutes before and after, helped with the small overlap of sensor and survey data talked about in section 5.2.2. Before the added window size there were 5601 corresponding sensor and survey data records, now there were 74,042 corresponding records from 10 users. Figure 48 shows how the mood dysregulation survey normally looks at approximately 3 minutes interval and then how the 30 minute window was added to the start and end time of the mood dysregulation survey. The three sensor reading shown on the graph are heart rate which is blue, breath rate which is orange, and activity which is yellow. The first two vertical lines are random surveys and the last vertical line is a mood dysregulation survey. The second graph shows how the start time and end time were extended 30 minutes.

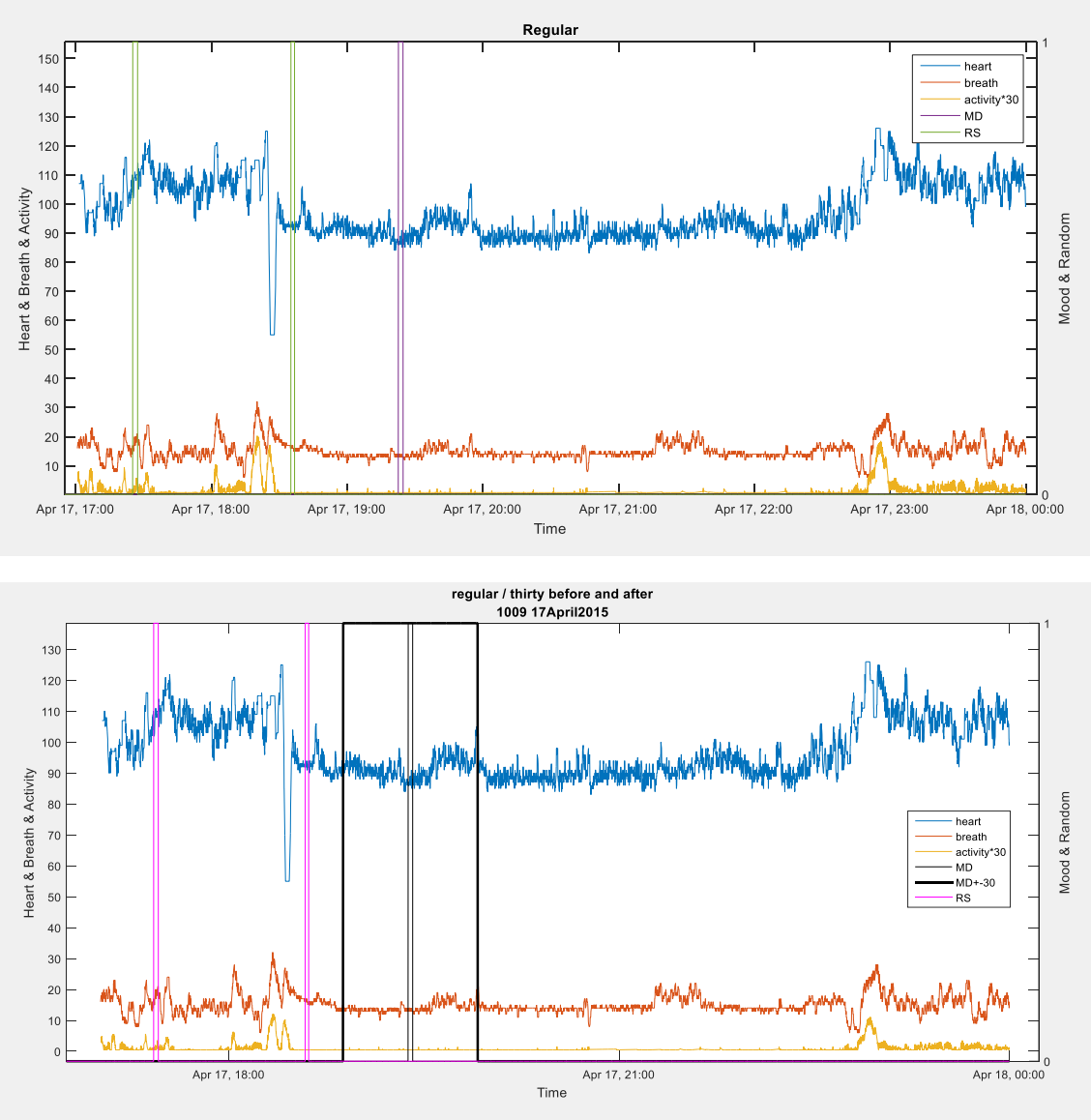

Figure 48. An example showing the 30 minute window extension to the start and end time of each survey. This was done to increase the results from the machine learning algorithms since the average survey time was only 3 minutes which is not enough data to analyze. The top graph shows the regular survey, vertical bars. The bottom graph show the same surveys with the third vertical bar, the black bar, showing the original survey and the 1 hour extension.

For future work, a researcher can also find mood dysregulation in some of the other surveys that were throughout the study. For example, there were many overlapping questions in the mood dysregulation parcel and the random assessment parcel items. It is possible to calculate the mean and standard deviation of responses for each user for these overlapping questions and

therefore calculate mood dysregulation in the random surveys. I did not have enough time to do this but if given more time this would be my next step in order to better balance the data.

Currently, there are way more non-mood samples than there are mood samples. There are approximately 8.5 million records in the database and only 74,042 records correspond to mood dysregulation labels to train the model after the 1 hour window is added to the mood dysregulation surveys. If a 30 minute window is extended on both the start and end time of the mood dysregulation survey then the machine learning algorithm has a larger window to predict when mood dysregulation is coming. The machine learning algorithm will also have more true positive samples. The more balanced the target class is, the better the model will be.

5.3 AMD Machine Learning Prediction

5.3.1 Evaluation Metrics

Our machine learning prediction performance is measured using classification metrics based off suggestions from the literature. Performance is measured using many metrics including accuracy, kappa, confusion matrices, and receiver operating characteristic (ROC) area. Accuracy of the correctly classified instances is interpreted as the correctly classified instances divided by the total number of instances. The incorrectly classified instances accuracy was also examined during AMD Machine Learning Prediction module. Kappa coefficient is a statistical measure of inter-rater reliability. Kappa is usually a better and more robust measure than accuracy due to kappa measuring the correlation between predictions and ground truth while taking into consideration the probability occurring by chance [2]. Kappa is used not only to evaluate a single classifier but can also be used to evaluate classifiers amongst themselves.

If we assume Kappa, κ, is a measure of agreement between categorical variables X and Y. Kappa is calculated from the observed and expected frequencies on the diagonal of a square

contingency table [56]. Suppose that there are n subjects on whom X and Y are measured and suppose there are g distinct categorical outcomes for both X and Y. Let f_{ij} denote the frequency of the number of subjects with the i^{th} categorical response for variable X and the j^{th} categorical response for variable Y [56]. Then the frequencies can be arranged in the following $g\ x\ g$ table:

Table 1. The frequencies of the number of subjects for each distinct categorical outcome.

	Y = 1	**Y = 2**	...	**Y = g**
X = 1	f_{11}	f_{12}	...	f_{1g}
X = 2	f_{21}	f_{22}	...	f_{2g}
| |	| |	| |	... ...	| |
X = g	f_{g1}	f_{g2}	...	f_{gg}

The observed proportional agreement between X and Y is defined as:

$$p_0 = \frac{1}{n}\sum_{i=1}^{g} f_{ii}$$

And the expected agreement by chance is:

$$p_e = \frac{1}{n^2}\sum_{i=1}^{g} (f_{i+})(f_{+i})$$

Where f_{i+} is the total for the i^{th} row and f_{+i} is the total for the i^{th} column. Thus, the kappa statistic is:

$$\hat{\kappa} = \frac{p_0 - p_e}{1 - p_e}$$

Cohen's kappa statistic is an estimate of the population coefficient:

$$\kappa = \frac{P_r[X = Y] - P_r[X = Y \mid X \text{ and } Y \text{ independent}]}{1 - P_r[X = Y \mid X \text{ and } Y \text{ independent}]}$$

Generally, $0 \leq \kappa \leq 1$, although negative values do occur on occasion [56]. Cohen's kappa is ideally suited for nominal categories [56]. In this research, our nominal category is mood dysregulation.

The confusion matrix is a 2x2 matrix which consists of the number of true positives and true negatives on one diagonal and the number of false positives and false negatives on the other diagonal. The confusion matrix can also be referred to as an error matrix and has a specific layout that allows the performance of a supervised learning algorithm to be visualized [57]. This research uses the confusion matrix to see if AMD is confusing two classes or commonly mislabeling one class as another. ROC is used as a visual representation of the sensitivity of the machine learning model. Sensitivity can be classified as true positive rate vs. false positive rate for a binary machine learning classifier as the discrimination threshold is varied [2]. The curve can be generated by plotting the true positive rate against the false positive rate for settings at various thresholds.

5.3.2 Prediction Models

The selected physiological features mentioned above and the mood dysregulation labels are used in [58], Waikato Environment for Knowledge Analysis (WEKA), to train several machine learning classifiers. In these research, several classifiers were trained and tested, however we will

discuss the top four in great detail. Therefore, we implemented four types of machine learning classifiers, Naïve Bayes, Bayesian Network, J48 Decision Tree, and Random Forest. One reason for choosing Naïve Bayes is because it is great to use for a baseline machine learning prediction model. We use Naïve Bayes to see what the simplest and quickest result would obtain and compare that result with the other machine learning methods.

Next, we decided to choose J48 decision tree because it is also very simple to implement and requires minimal computation resources when compared to other available classifiers for nominal attributes [59]. Since it does not require much computation, this model can be implemented on a smart phone for future work of AMD. From Cristina Petri's work in 2010, decision trees are a class of data mining and machine learning techniques that have roots in traditional statistical disciplines such as linear regression [60]. She stated decision trees also share roots in the same field of cognitive science that produced neural networks. Decision trees are a simple, but powerful form of multiple variable analysis [60]. Petri shared some advantages of using Decision Trees include: being able to visual the result, very efficient, can easily modify, and decision trees can handle both nominal and numeric attributes. However, one disadvantage with decision trees are most of the algorithms, like the one used in this research, requires the target attribute to only have discrete set of values [60].

The Random Forest machine learning algorithm is one of the best among classification algorithms because it is able to classify large amounts of data with high accuracy, as stated by Walker in 2013 [61]. Random Forests are an ensemble learning method for classification and regression that constructs a different number of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. That is why Random Forest can be thought of as a form of nearest neighbor search algorithms [61].

Walker continues by saying, Random Forests are a combination of tree predictors where each tree depends on the values of a random vector sampled independently with the same distribution for all trees in the forest [61]. The basic principle is that a group of weak predictors can come together to form a strong predictor group. Random Forests are a great tool for making predictions since they do not over fit the data because of the law of large numbers while introducing the right kind of randomness allows them to accurately create classifiers and repressors [61]. The one drawback found in this research is that Random Forest have a consistently linear efficiently time with J48 as the data gets larger, e.g. the total execution time consistently gets larger as the data gets larger and is consistently slower than J48.

5.3.3 Experiments on Prediction

There were many different cases and experiments done on the four machine learning models discussed in the previous section. When using Naïve Bayes, Bayesian Network, J48 Decision Tree, and Random Forest the two main categories of a balanced and non-balanced target class was used. Figure 49 below show the distribution of mood dysregulation without balancing the target class for all the data composed of one user.

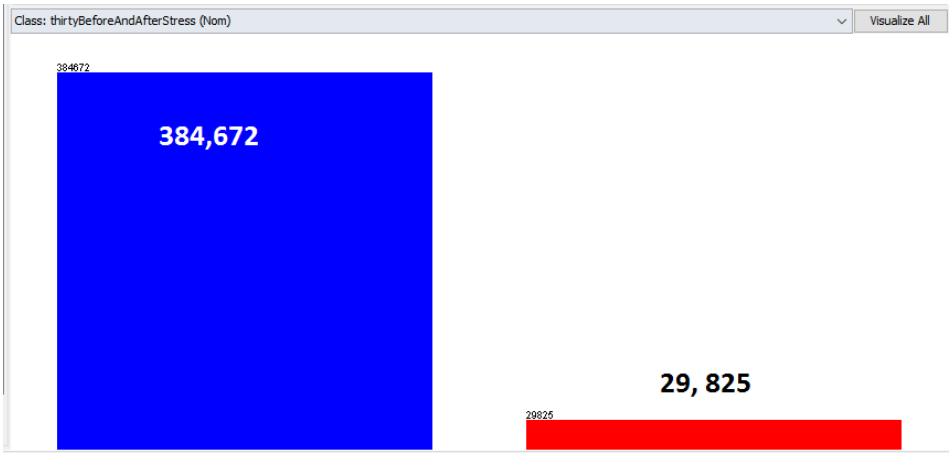

Figure 49. Bar graph to show the non-balanced mood dysregulation class. It is shown how many mood labels are available and the distribution between the two classes. The blue bar is the negative non-mood dysregulation samples and the red bar is the positive mood dysregulation samples.

When using a non-balanced class, it brings some trouble analyzing the result. For example, after running a J48 decision tree, the baseline would be 80.52% and the overall accuracy would be 89.94%. The baseline is when the algorithm always predicts false for mood dysregulation, if this were to happen on the above example then the results would be 92%. By only looking at the accuracy, one might assume the final results are good. However, if the baseline is so high, then the accuracy does not have a clear meaning. Therefore, a balanced class must be used in order to fully know the results of the machine learning algorithm and to judge the algorithm based off accuracy alone.

Before a balanced class is used, there were several cases performed on a non-balanced class such as comparing the three different data sets: clean data versus the pre-cleaned data versus the raw data from Hexoskin. When we use the words “clean data” we mean AMD’s cleaning pipeline was ran on the data and the unreliable data was removed, the outliers were determined by Loess, and perform regression imputation to fill in the outliers. When we use the words “pre-cleaned data” we mean only run half of AMD’s cleaning pipeline and the unreliable data was removed, however this is the only cleaning done. So basically “pre-cleaned data” is only removing the unreliable data. When we use the words “raw data” we mean the data that was provided by Hexoskin servers. Hexoskin does some pre-processing on the data before we receive it which was discussed in previous sections, however it will be known here as raw data. With a non-balanced class, all the three data sets were compared with using time and when time was not used. The time comparison will be talked about further in the next paragraph.

Next, a balanced mood dysregulation class is analyzed. By balancing the class, the baseline prediction, predicting all zeros for mood dysregulation, will be approximately 50%. Since the baseline is 50%, the overall accuracy makes more sense. Figure 50 below is a visual representation

of a balanced mood dysregulation class. If the prediction algorithm predicts all zeros, then the result will obtain 50.1% accuracy for this particular example.

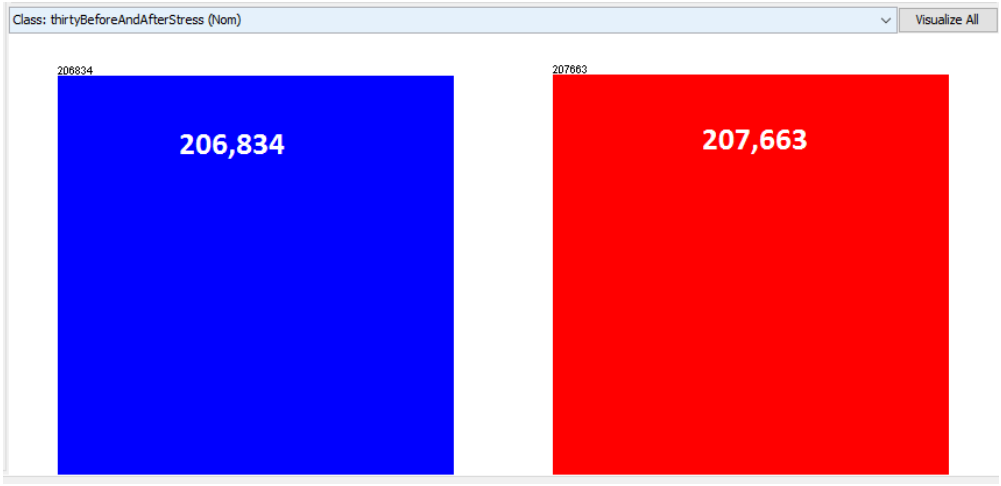

Figure 50. A bar graph showing a balanced mood dysregulation class. If a machine learning prediction algorithm predicts all zeros to determine a baseline, the result will obtain 50.1% accuracy. This helps the researcher to understand the overall accuracy better when compared to a non-balanced class.

When using a balanced class, the three data sets were compared: clean data versus pre-cleaned data versus raw data. While comparing all three data sets, time was taken into consideration as well. So for each data set, an analysis was done using time and another analysis was done without using time. In the previous section 5.2.5, we discussed how time attributes were created in more detail. So for each of the three datasets, we compared when using time categories and the hour of the day. We tested using one or both of the time attributes and compared results. For both balanced and non-balanced classes, using time obtained better accuracy. More results will be discussed in the next chapter, Chapter VI, Analysis and Experimental Results.

6. ANALYSIS AND EXPERIMENTAL RESULTS

We now present the results of applying and evaluating AMD's machine learning pipeline using the physiological features to predict mood dysregulation on the field data where the user is in their natural environment. For all of the experiments performed and analyzed, results were obtained and analyzed. Multiple experiments were performed and multiple models were trained and tested. The four models discussed in this research are Naïve Bayes, Bayesian Network, J48 Decision Tree, and Random Forest. We used 10-fold cross validation to obtain the performance measures of all four machine learning algorithms. First a comparison is made between the three datasets: Clean Data versus Semi-Clean versus Raw Data. These three datasets are described in the previous section 5.3.3 and are clearly defined. Next, a comparison is made between having a non-balanced class versus a balanced target class. Proceeding, a comparison is analyzed between using time and not using time in the machine learning model. Lastly, mathematical and statistical explanations are provided to allow readers to understand why the results are improved dramatically.

6.1 Different Approaches for Model

6.1.1 Clean vs. Semi-Clean vs. Raw Data

Once AMD's cleaning pipeline was finished, there needed to be some comparison in order to determine if the cleaned data analyzed with the same machine learning algorithm performed greater than the raw data from Hexoskin which has some pre-processing done already. Each of the three data sets here are discussed in detail and clearly defined in the previous section 5.3.3. When running the clean data in a J48 decision tree with a non-balanced class, the baseline was 80.52% and the accuracy was 88.95% for 79,696 instances. Since the baseline is so high, it is hard to understand the accuracy, therefore we will be looking at the kappa values to analyze the three

data sets. The kappa value for the cleaned data achieved a value of 63.16%. Kappa was discussed and clearly defined in the previous section 5.3.1 and has greater meaning than overall accuracy.

Next, we analyzed the results when semi-cleaned data was used as a dataset. The semi-cleaned dataset has the same number of instances as the clean data. When running the semi-cleaned data in a J48 decision tree with a non-balanced class, the baseline was 80.52% and the accuracy was 88.48% with 79,696 instances. This result is a little bit lower than the cleaned data which is logical because less cleaning has been done and thus the data is not as good. When looking at the kappa value, the semi-cleaned data received a value of 61.31% which has a little more separation than the overall accuracy. From the cleaned data to the semi-cleaned data there was a 1.85% decrease in kappa value. Therefore, the cleaned data performs better than the semi-cleaned data.

Last, we analyzed the results from the raw data from Hexoskin versus the cleaned data. Remember, the raw data is technically not raw from the sensors. Hexoskin does some pre-processing and averaging of all available sensor metrics. This is discuss and clearly defined in the previous sections 5.3.3 and 5.1. For example, the raw RR interval from Hexoskin's sensor has a frequency of 265Hz which supplies 256 samples per second. When we receive the data from Hexoskin's server, the RR interval signal is received at 1Hz or 1 sample per second because it is averaged down to produce a more consistent signal. The heart rate is calculated from RR interval, thus the heart rate is averaged over the last 16 beats to obtain a 1 sample per second value, hence 1Hz. By doing this, Hexoskin's data is technically not raw, but we do not do any cleaning or processing on this data, therefore it is referred to as raw data in this research.

When running the raw data in a J48 decision tree with a non-balanced class, the baseline was 80.57% and the accuracy was 89.94% for 153,576 instances. As described, the raw data has a

larger number of instances due to the fact that no cleaning or removing was done on this data. Also, the data has a higher accuracy than the cleaned data or the semi-cleaned data. This really surprised us and actually hurt a little bit because the cleaning pipeline took several months to complete. Then after the cleaning pipeline is complete, we find that the results are higher with the raw data. Some say, your results are higher because there are more samples and instances. This is not true. The kappa value obtained from the raw data is 66.26% which is a 3.1% increase from the clean data. Therefore, the raw data obtains better results than the cleaned data. Moreover, the raw data obtained the best results and will be used in the proceeding approaches for the model. Mathematical and statistical explanations will be further examined in 6.1.5. Figure 51 below is a visualization representation between the three datasets and their outcomes as described above.

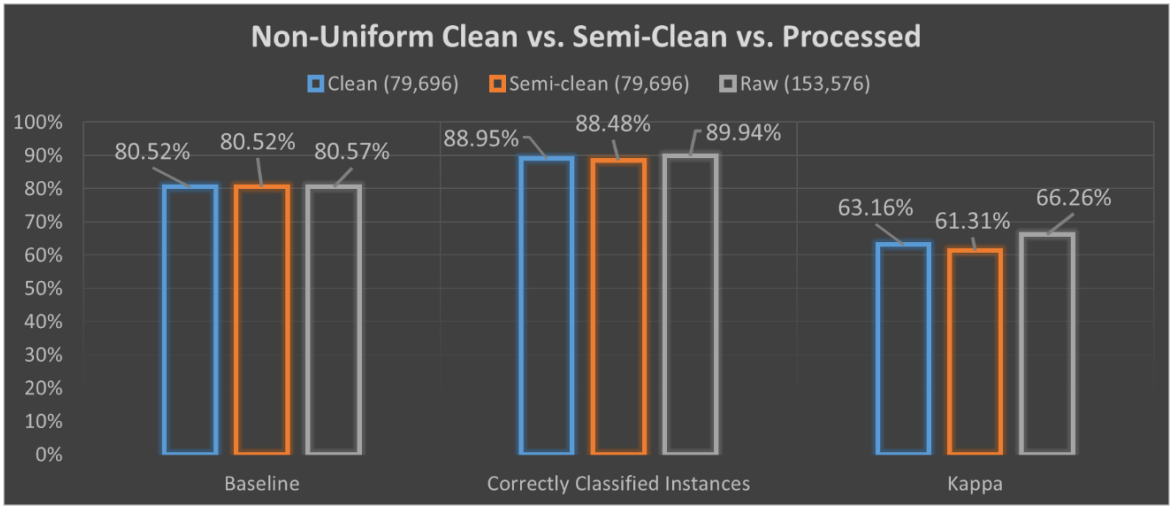

Figure 51. Bar graphs for each of the three datasets to analyze and compare their results: Clean, Semi-Clean, and Raw. Each dataset was analyzed with a J48 Decision Tree and a non-balanced target class. Surprisingly, the raw data obtained the highest results and is analyzed further in section 6.1.2.

6.1.2 Non-Balanced vs. Balanced Class

After examining the three datasets in the previous section, it was starting to be apparent the target class needed to be balanced in order to understand the results. Without a uniform and

non-balanced class, the baseline might be 92% and the accuracy might be 95% which can be seen in Figure 52. When you first see the accuracy, one might assume the classifier is almost perfect but after further examination and research, the kappa statistic had a result of 58% which is a moderate magnitude when following some guidelines that have appeared in the literature such as Landis and Koch [62]. Based off Fleiss's equally arbitrary guidelines, any kappa value over 0.75 is excellent agreement, 0.4 – 0.75 is fair to good agreement, and below 0.40 is poor, the kappa value of 58% would be in-between fair and good [63]. Therefore, the accuracy does not have a clear meaning when the baseline is so high. From the baseline to the accuracy there is only a 3% difference, thus a different approach is applied and results are compared.

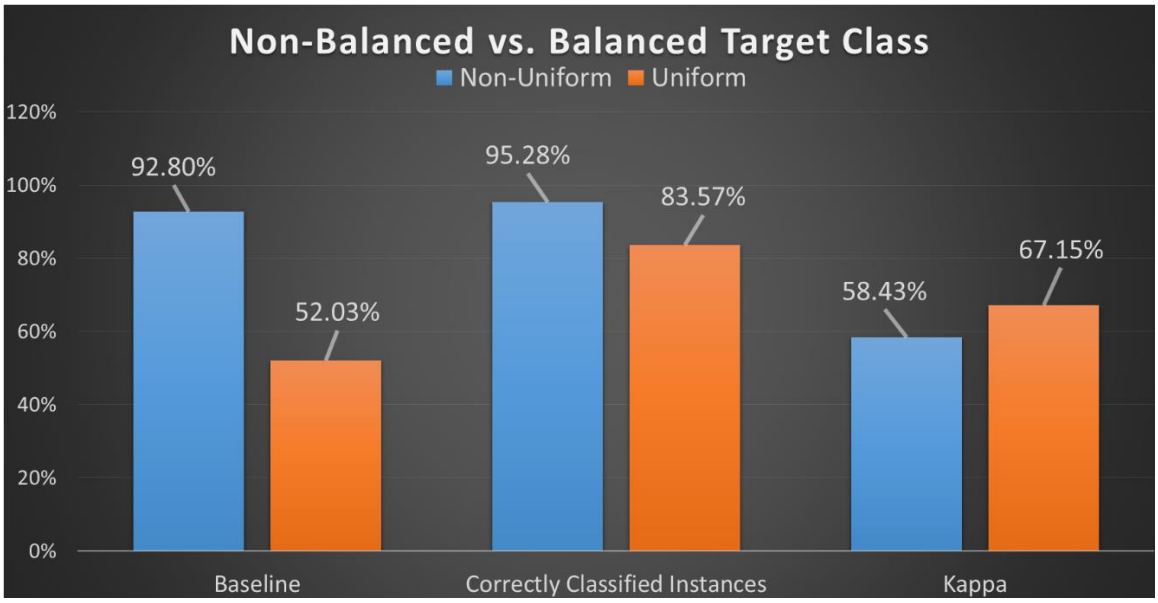

Figure 52. Results for non-balanced versus balanced target class. With a non-balanced target class, the baseline is 92% and the accuracy is 95%. If only analyzing the accuracy the results are skewed and do not make sense. Therefore, resampling to achieve a balanced class are done to better understand the results obtained.

Resampling is done in order to make the target class balanced and uniform. There are many options when resampling and each of these options were tried, tested, and compared. For example, with resampling there is an option to have replacement or without replacement. The essence of resampling is to use only the sample data and to resample from that data to create

different realizations of the experimental results [64]. Resampling can be used on the data sample itself. By resampling with replacement you can make samples of n by drawing repeatedly from n events, and get different samples each time. Sampling can be done from the parent ensemble either with replacement or without replacement. Drawing with replacement means that all draws are identical, and the same event maybe be drawn more than once in the same subsample. Drawing without replacement means that an event may not occur more than once in a particular sample, though it may appear in several different samples [65].

Resampling with and without replacement was used in order to obtain a balanced and uniform target class. The purpose of resampling was to get the positive mood dysregulation samples approximately the same amount as the negative mood dysregulation samples. Some people resample to get more positive samples however this was not the case in this research. The main purpose was to decrease the amount of negative mood dysregulation samples. After running 100 resamples with replacement and 100 resamples without replacement the mean of the increase of the accuracy was approximately 1% for a J48 Decision Tree when the target class is balanced. The mean of the increase for the kappa statistic was 1.56%. This means they two results are practically the same. Since we wanted to leave the data as original as possible and not draw two of the same sample twice, we decided to go with no-replacement. Figure 53 below is a visual for the comparison between resampling techniques.

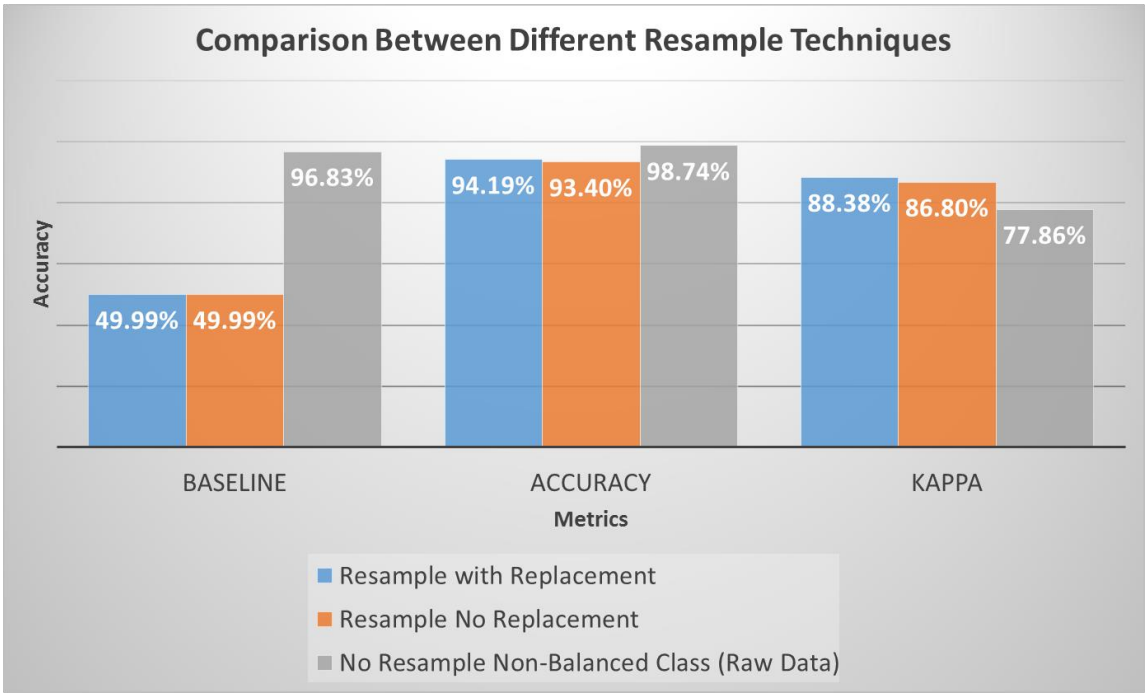

Figure 53. Bar charts showing visual results when comparing the different resampling techniques. It is show that when resampling with or without replacement receives very similar results. After running both methods 100 times, the average increase in accuracy is only 0.79%.

After resampling, retaining the same amount of positive mood dysregulation samples, the baseline is approximately 52%. By baseline, we mean when the machine learning algorithm predicts all zeros for a binary target class, the result will be 52% accuracy. With the baseline around 50% we can now analyze the accuracy normally as well as the other statistical formulas. After resampling, the accuracy of a J48 decision tree with a balanced class is approximately 83%. The accuracy is moderate here because we are not using time when analyzing the data, this accuracy is formed from only the physiological data. However, for mood, time of the day is essential. After resampling, the kappa value increases to approximately 67%. This is a 9% increase when compared to the non-balanced class. This makes sense because there are less negative mood dysregulation values causing the kappa value to increase slightly. Overall, the balanced and uniform class has higher kappa results and makes more sense, therefore it is used in the proceeding sections and in this research.

6.1.3 Time vs. Without Time Features

For all the results up to this point, none of the time features have been included. Therefore, without time features, we were able to get an averaged accuracy of approximately 83% and a kappa value of approximately 67%. This is moderately good considering mood dysregulation will usually happen during similar times for each individual. Since we do not include time, we are withdrawing one of the most valuable pieces of information from our classifier. For example, if someone is sleeping from 11pm to 7am, then the probability for mood dysregulation to occur is very slim. Based off each user in our study, the probability is zero for a user to have mood dysregulation during the morning category, however, it could be possible. For every subject in the study, most mood dysregulation occurred during the afternoon and evening categories. The afternoon category is anytime between 11am and 5pm. The evening category is any time between 5pm and 11pm. These finding will be discussed more in the proceeding section 6.1.5.

There were two main time features created and used, they were the Hour of Day and Time Category, both integer values. These two features were created from the original timestamp and added for all the users. The creation of these features is discussed deeply in the previous section 5.2.5. After adding these features, the accuracy improved approximately 10%. This shows that time is a very important factor for determining mood dysregulation from physiological data. When adding time attributes, the accuracy went from 83.57% increasing to 93.46%, which is an increase in accuracy of 9.89%. We were all very excited when this occurred. After adding time attributes, the kappa value went from 67.15% increasing to 86.93%, which is an increase in kappa by 19.78%. Figure 54 below show visual representation of the results obtained with and without time attributes.

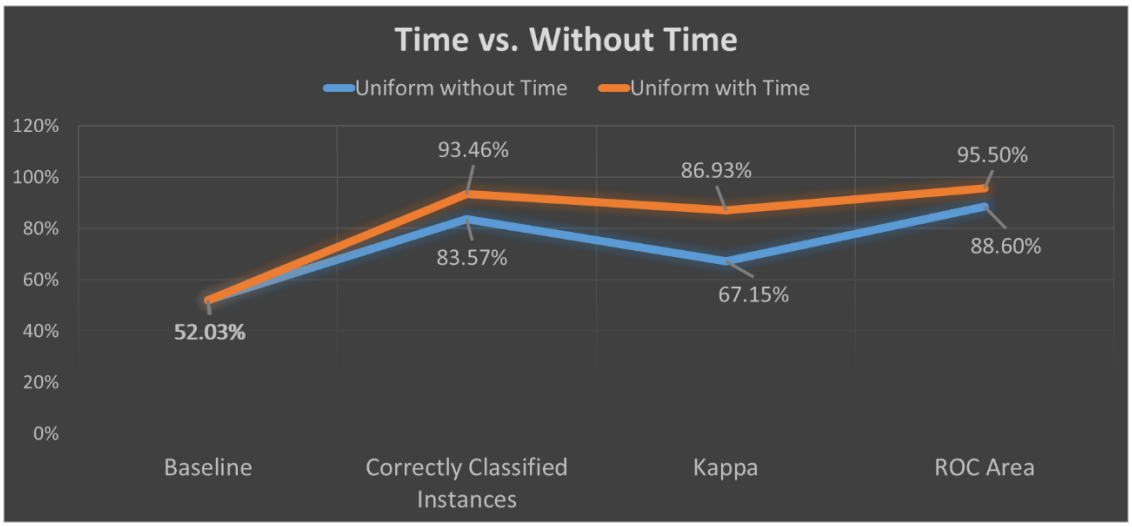

Figure 54. Visual representation of the results when using time attributes compared to not using time attributes. Time is an important factor in mood dysregulation episodes, therefore using time attributes increases the accuracy by approximately 10% for a J48 Decision Tree.

Based off Fleiss's equally arbitrary guidelines, any kappa value over 0.75 is excellent agreement [63]. Moreover, following Landis and Koch's guidelines, any kappa value between 0.61 – 0.80 has substantial agreement and any kappa value between 0.81 – 1 is almost perfect agreement [62]. Therefore, our kappa value of 86.93% after adding time features is an incredible and very valuable find. However, we were able to increase the results after performing feature selection algorithms discussed in the next section 6.1.4.

6.1.4 Feature Selection Algorithms

After running J48 Decision Tree many times on all these different approaches and experiments, we started to notice trends with the tree structure. Figure 55 below shows a visual example of the J48 decision tree structure for a real subject in our study. When running the model on each subject, we noticed the same seven attributes were being generated at the most significant positions of the tree structure. These features included: Time categories, expiration, inspiration, heart rate, hour of day, breathing rate, and minute ventilation. This was seen visually by the tree structure, later we ran feature selection algorithms to verify the most important

features which will be discussed next. The top five features were: Inspiration, Expiration, Hour of Day, Time Categories, and Heart Rate.

```
J48 pruned tree
------------------

timeCategories <= 1: 0 (76001.0)
timeCategories > 1
|   expiration <= 26165
|   |   inspiration <= 26202
|   |   |   heartRate <= 179
|   |   |   |   onlyHour <= 13
|   |   |   |   |   heartRate <= 68
|   |   |   |   |   |   onlyHour <= 1
|   |   |   |   |   |   |   expiration <= 7355
|   |   |   |   |   |   |   |   breathingRate <= 14
|   |   |   |   |   |   |   |   |   minuteVentilation <= 6706.4
|   |   |   |   |   |   |   |   |   |   activity <= 0
|   |   |   |   |   |   |   |   |   |   |   onlyHour <= 0
|   |   |   |   |   |   |   |   |   |   |   |   minuteVentilation <= 3439.52: 0 (1425.0)
|   |   |   |   |   |   |   |   |   |   |   |   minuteVentilation > 3439.52
|   |   |   |   |   |   |   |   |   |   |   |   |   heartRate <= 60
|   |   |   |   |   |   |   |   |   |   |   |   |   |   breathingRate <= 7: 1 (2006.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   breathingRate > 7
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   minuteVentilation <= 6148.64: 0 (69.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   minuteVentilation > 6148.64
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   heartRate <= 56: 1 (354.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   heartRate > 56
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   rrInterval <= 1.148438: 0 (17.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   rrInterval > 1.148438: 1 (17.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   heartRate > 60
|   |   |   |   |   |   |   |   |   |   |   |   |   |   breathingRate <= 6
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   tidalVolume <= 717.12: 0 (63.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   tidalVolume > 717.12
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   rrInterval <= 0.800781
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   breathingRate <= 5: 0 (2.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   breathingRate > 5: 1 (12.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   rrInterval > 0.800781: 1 (169.0/1.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   breathingRate > 6: 0 (650.0)
|   |   |   |   |   |   |   |   |   |   |   |   onlyHour > 0
|   |   |   |   |   |   |   |   |   |   |   |   tidalVolume <= 252.32
|   |   |   |   |   |   |   |   |   |   |   |   |   minuteVentilation <= 2204.48
|   |   |   |   |   |   |   |   |   |   |   |   |   |   breathingRate <= 6: 0 (35.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   breathingRate > 6
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   heartRate <= 61
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   heartRate <= 54
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   minuteVentilation <= 1938.88: 1 (42.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   |   minuteVentilation > 1938.88: 0 (40.0)
|   |   |   |   |   |   |   |   |   |   |   |   |   |   |   heartRate > 54
```

Figure 55. The J48 Decision Tree structure from the model used on all subjects. The important features and the path of each node can be visualized. This is important to see if the decision tree is generating a useful representation of the data and can be used to determine the most valuable features.

At first we wanted to see if choosing the top 5 features would produce similar results. All the results discussed here are using a J48 decision tree, balanced predicted target class with no replacement, for all the users in the study, and on all days these users were in the study. The

original results we obtained were 93.46% accuracy and 86.93% kappa by using all 11 features: Activity, breathing rate, cadence, expiration, heart rate, inspiration, minute ventilation, RR interval, tidal volume, time categories, and hour of day. After using the top five features selected by correlation, revealed in the previous paragraph, we obtained an accuracy of 86.37% and a kappa value of 72.75%. This was a slight decrease in the results, however the results were consistent with the original results, which suggests that the each prediction model is correctly assembled.

The top five features were selected by a correlation-based feature subset selection algorithm for machine learning. This correlation algorithm evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them. Subsets of features that are highly correlated with the class while having low inter-correlation are preferred [66]. A good feature selection subset is on that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other [66].

Definition: A feature V_i is said to be relevant IFF there exists some v_i and c for which $p(V_i = v_i) > 0$ such that

$$p(C = c \mid V_i = v_i) \neq p(C = c)$$

If the correlation between each of the components in a test and the outside variable is known, and the inter-correlation between each pair of components is given, then the correlation between a composite test consisting of the summed components and the outside variable can be predicted from:

$$r_{zc} = \frac{k\overline{r_{zi}}}{\sqrt{k + k(k-1)\,\overline{r_{ii}}}}$$

Where r_{zc} is the correlation between the summed components and the outside variable, k is the number of components, $\overline{r_{zi}}$ is the average of the correlations between the components and the outside variable, and $\overline{r_{ii}}$ is the average inter-correlation between components [66].

Next we thought we would try adding a few of the features back, since we were curious if we could obtain better results by removing some of the features from the total set. After performing another feature selection algorithm, wrappers for feature subset selection, we found the top 7 features and ran another analysis. An example output of the tree is presented below in Figure 56. The wrapper subset evaluates attribute sets by using a learning scheme. Cross validation is used to estimate the accuracy of the learning scheme for a set of attributes [67]. Then best first search algorithm is used on the set of attributes to obtain the resulting subset of 7 attributes: Time categories, inspiration, expiration, heart rate, hour of day, breathing rate, and minute ventilation. Once these 7 attributes are used to train and test the machine learning J48 decision tree, we obtain accuracy results of 94.08% and kappa results of 88.16%. This is an increase of 0.62% on the accuracy and an increase of 1.23% on kappa statistic. This was the best result we could obtain using a J48 decision tree, with the same positive samples as the original raw data, using a balanced target class without replacement, on all the users of the study.

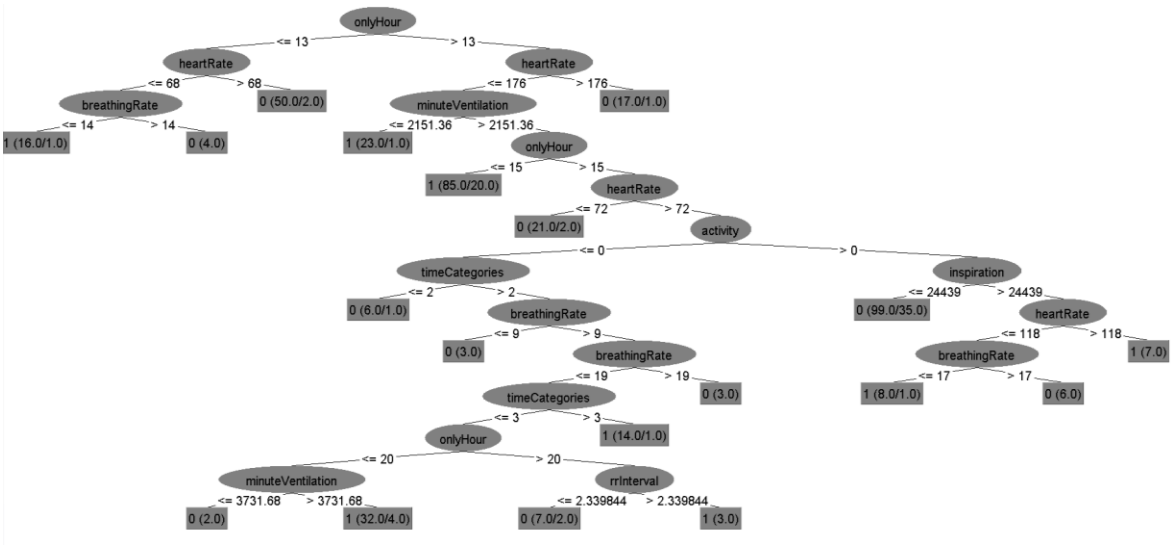

Figure 56. J48 tree structure from top features selected. After performing wrappers for feature subset selection, we found the top 7 features. Then best first search algorithm is used on the set of attributes to obtain the resulting subset of 7 attributes: Time categories, inspiration, expiration, heart rate, hour of day, breathing rate, and minute ventilation.

6.1.5 Discussion

Theoretically, it is logical to think that adding time to the machine learning pipeline, as two separate attributes, would increase the accuracy of the model. However, we wanted to provide some evidence to why this is true. Based off the seven subjects that made it to the machine learning predication module, we ran a simple statistical analysis on their mood dysregulation episodes. For each user, most mood dysregulation happens during the afternoon and evening categories. The afternoon category is anytime between 11am – 5pm and the evening category is anytime between 5pm – 11pm, 6 hour intervals. Figure 57 below shows bar graphs representing the number of samples in each time category for each subject in the study.

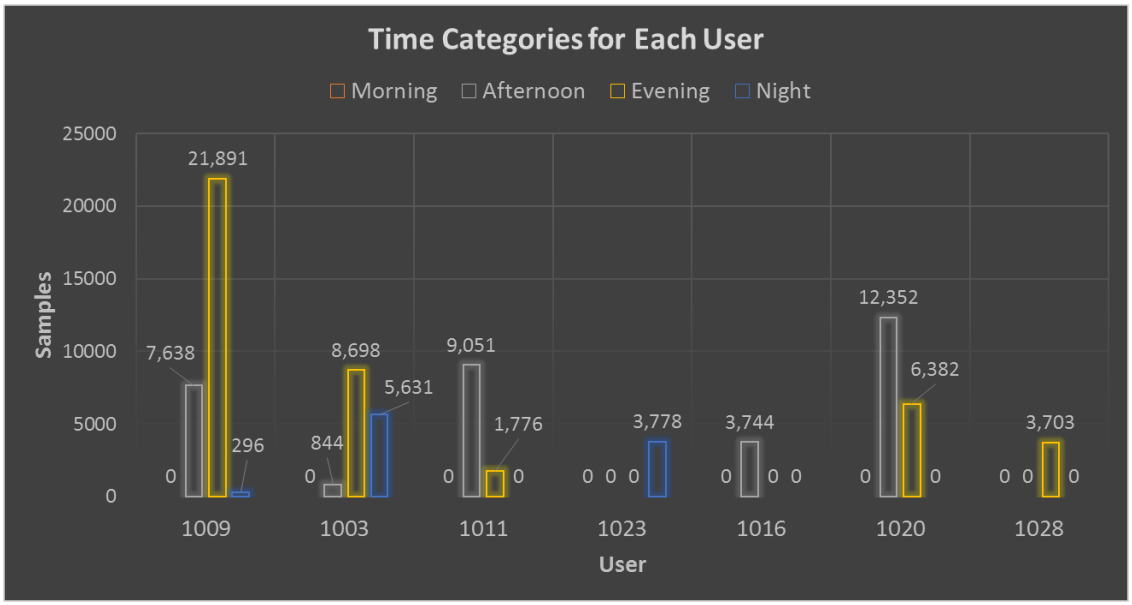

Figure 57. Bar graphs showing the total number of mood dysregulation samples in each time category for each of the users in the study. As shown, most users did not have mood dysregulation in the morning and the night. A majority of the mood dysregulation happens during the afternoon and evenings.

Looking at distributions for each time category, we can evaluate even deeper why time is such an important factor. The most frequent time mood dysregulation occurs for the seven users is the evening time category with a frequency distribution of 49.48%. The afternoon category is the second most frequent time mood dysregulation occurs with 39.20% distribution. The third time category mood dysregulation occurs in is during the night category. The night category is anytime between 11pm and 5am. The night category had a frequency distribution of 11.32%. Remaining is the morning category, which had a distribution of 0%. Meaning no mood dysregulation happened during the morning for all of the subjects in the study that made it to the machine learning prediction module. However, this does not mean mood dysregulation is impossible to happen during the morning for these subjects and it does not mean mood dysregulation will never happen in the morning for other subjects. All this means is mood dysregulation did not happen in the morning for these subjects during the time of their study. Figure 58 below shows the frequency distribution of all the subjects combined in the study.

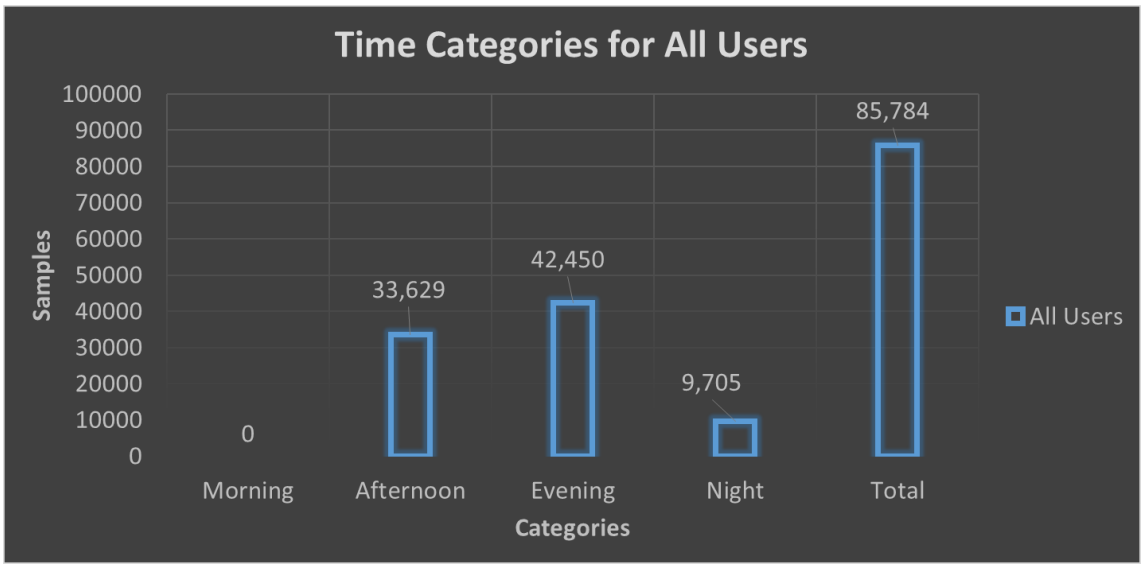

Figure 58. Bar graphs showing the frequency distribution for mood dysregulation episodes occurring to all subjects in the study. As shown, the distribution for afternoon and evening categories is 39.20% and 49.48%, respectively. Mood dysregulation rarely happened during the night category and never happened during the morning category, therefore time is a very important feature in the machine learning prediction.

6.2 Model Comparison

6.2.1 Accuracy

After understanding the data more, choosing the best attributes, creating time attributes, and getting the pipeline in the correct order to obtain the highest results, we wanted to compare different machine learning algorithms in order to compare accuracy and efficiency. The four main machine learning algorithms discussed in this research are Naïve Bayes, Bayesian Network, J48 Decision Tree, and Random Forest. Each of these algorithms were trained and tested on the same dataset with the same settings. The valid number of data points were 2,707,252 (85,784 classified as stress and 2,621,466 as baseline). To avoid problems with unequal and non-balanced sample sizes for the target class, the sample sizes for each class were equalized before training the machine learning algorithms, by selecting a random sub-sample of 171,910 samples (85,784 classified as stress and 86,126 as baseline). The stress class was not modified, it was randomly sampled however it is the same as the original data set since the no-replacement parameter was

set to true. Figure 59 below shows the results of several machine learning algorithms testing or prediction performance on all 11 features (9 physiological and 2 time attributes). In the proceeding sections, we describe the accuracy and efficiency of these algorithms.

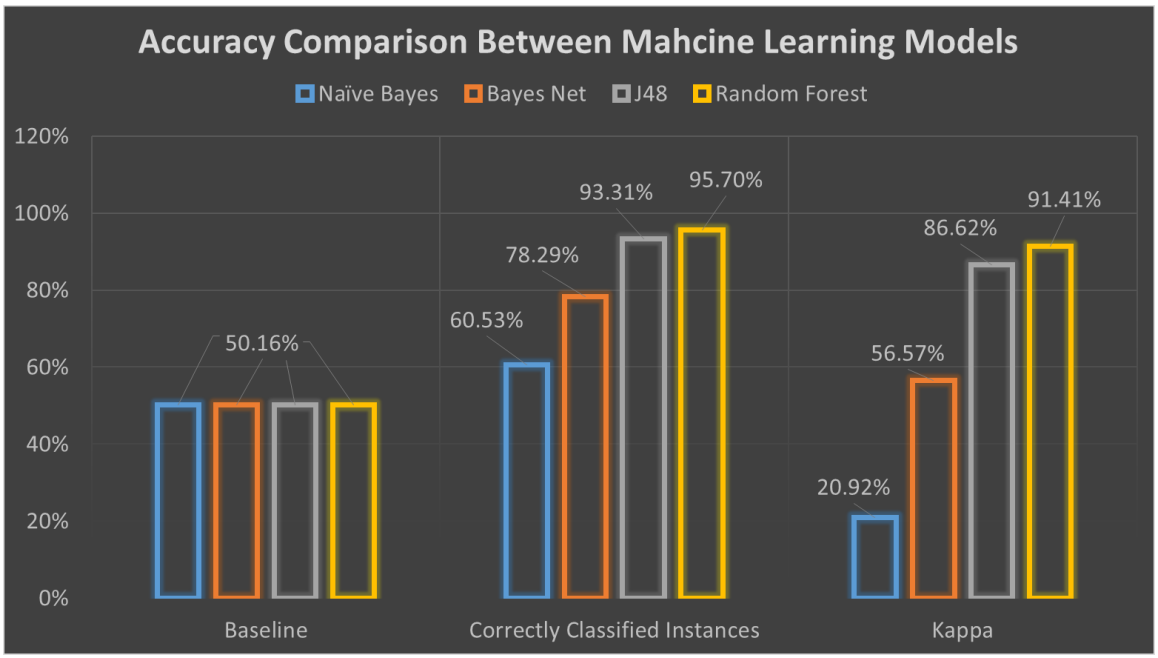

Figure 59. Visual comparison between four machine learning algorithms. Accuracy and kappa statistics are compared and Random Forest has the highest values.

Naïve Bayes was used as a baseline model for the accuracy and kappa values. Naïve Bayes is a classifier using estimator classes. Numeric estimator precision values are chosen based on analysis of the training data [68]. Naïve Bayes received an accuracy of 60.53% with a kappa value of 20.92%. When comparing with the literature, this kappa value is low and not acceptable in this research. However, Naïve Bayes was used as a baseline so this result was reported.

Next we used a Bayesian Network for classifying mood dysregulation. The Bayesian Network learning using various search algorithms and quality measures while providing data structures such as network structure, conditional probability distributions, etc. Facilities common to Bayesian Network learning algorithms like K2 and B. The Bayesian Network received an

accuracy of 78.29% and a kappa statistic of 56.57%. Any kappa statistic above 40% is moderate and acceptable. Bayesian network's performance is low but is reported for comparison between other machine learning algorithms.

When reading literature, many other research projects that were trying to predict stress episodes from physiological data recommend using J48 Decision Tree, which is why we used that in the previous section for determining all the different approaches. After running J48 compared to other machine learning algorithms, we can see why the literature suggests J48. It is because J48 is fast, simple, and accurate. The J48 decision tree received an accuracy of 93.31% and a kappa value of 86.62%. A kappa value above 80% is outstanding and almost perfect agreement. J48 Decision Tree, also known as C4.5, became really popular in the early 2000's after ranking #1 in the Top 10 Algorithms in Data-Mining after a paper was published in 2008 [69]. J48 builds decision trees from a set of training data using the concept of information entropy. J48 is an open source Java implementation of the C4.5 algorithm in the WEKA data mining tool. Table 2 below shows the confusion matrix produced by J48 Decision Tree using one model for one subject.

Table 2. Confusion matrix for J48 Decision Tree using one model for one subject. For this result, the total number of instances is 41,449. The accuracy is 93.38% and the kappa value is 86.77% for this confusion matrix. The receiver operating characteristic (ROC) area is 95.7% for this model.

Classified As	False	True
False	18,884	1,771
True	971	19,823

Next, we used Random Forest using the same dataset and conditions as the previous machine learning algorithms for constructing a forest of random trees. Random Forest had the

highest results of all the machine learning algorithms. Random forest received an accuracy of 95.70% which is optimal. Random forest obtained a really high kappa value of 91.41%. Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them [70]. Table 3 below shows the confusion matrix produced by Random Forest using one model for one subject.

Table 3. Confusion matrix for Random Forest using one model for one subject. For this result, the total number of instances is 41,449. The overall accuracy is 95.70% and the kappa value is 91.41% for this confusion matrix. The receiver operating characteristic (ROC) area is 99% which is slightly better than a J48 Decision Tree.

Classified As	False	True
False	19,443	1,212
True	569	20,225

All of the four models were tested with one user for one model and one model across all users. To compare one model across all subjects is to see whether we can build a generic model that will work for all the subjects in the study. Figure 60 below shows the accuracy results across all users. The results are consistent with one model for one user. Naïve Bayes obtained the lowest accuracy and is used for a baseline comparison for the other algorithms. Naïve Bayes reported an accuracy of 60.48% and a kappa value of 21.02%. The kappa value for this results has very low agreement and is not acceptable. Bayesian Network received the next best accuracy receiving 76.43% and a kappa value of 52.87%, any kappa value above 40% is moderate. J48 was the second

best algorithm for accuracy obtaining an overall accuracy of 93.46% and a kappa value of 86.93%. From the literature, any kappa value above 80% has almost perfect agreement. Random Forest obtained the best accuracy receiving 95.85% and a kappa value of 91.69%. All of these values were consistent with the single user single model approach which suggests the models were implemented correctly. Table 4 below shows the confusion matrix produced by J48 Decision Tree using one model for all subjects. Table 5 below shows the confusion matrix produced by Random Forest using one model for all subjects.

Table 4. Confusion matrix for J48 Decision Tree using one model for all subjects. For this result, the total number of instances is 171,910. The accuracy is 93.46% and the kappa value are 86.93% for this confusion matrix. The receiver operating characteristic (ROC) area is 95.5%.

Classified As	False	True
False	78,736	7,390
True	3,849	81,935

Table 5. Confusion matrix for Random Forest using one model for all subjects. For this result, the total number of instances is 171,910. The accuracy is 95.84% and the kappa value is 91.69% for this confusion matrix. The receiver operating characteristic (ROC) area is 99.2% which is slightly better than J48 Decision Tree.

Classified As	False	True
False	80,795	5,331
True	1,809	83,975

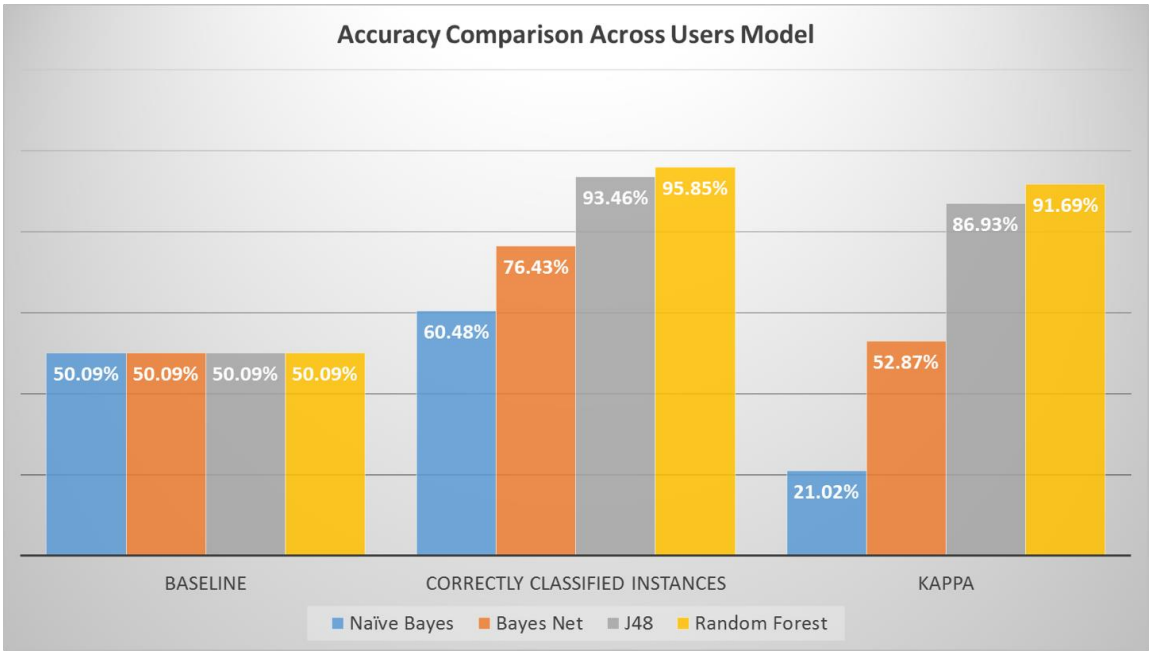

Figure 60. Bar graphs showing the accuracy and kappa statistic for each of the four models across all subjects. It is shown, Random Forest obtained highest accuracy as well as the highest kappa value. However, J48 Decision Tree achieved similar results and has better efficiency.

6.2.2 Efficiency

Accuracy is only one measurement to analyze a machine learning algorithm. However, the number of minutes or even hours necessary to train a model has a large variant between different algorithms. There does exist some correlation between training time and accuracy because one typically accompanies the other [71]. Moreover, when the number of data points are large, some machine learning algorithms are more sensitive and effected when compared to other machine learning algorithms. When time is of the essence and limited, this can lead one to choose a specific algorithm over another, especially when the dataset may be extremely large. Therefore, all of these reasons prove that efficiency must also be analyzed in order to fully examine the machine learning algorithms used. When analyzing the four algorithms used, the time taken to build the model is recorded as well as the total execution time which are averaged

with 7 runs for each algorithm, one for each user. The total execution time includes the time taken to build, train, test, and validate the model.

Since Naïve Bayes is one of the simplest algorithms, obtaining the least amount of accuracy as shown in the previous section, it is also the fastest algorithm. Naïve Bayes obtained an averaged time of 0.11 seconds to build the model and an averaged total execution time of 2.08 seconds. Naïve Bayes is fast. By the time we ran the program and looked down at the clock, the algorithm was finished. Next we report the efficiency for Bayesian Network (Bayes Net). Bayes Net obtained a better accuracy when compared to Naïve Bayes and thus it resulted in a lower efficiency. The averaged time taken to build the model for Bayes Net took 0.42 seconds. The averaged total execution time for Bayes Net received a time of 4.77 seconds. As one is reading this research, they might think the algorithms were fast because the dataset is small, however, this is not true. The dataset for each of these test contained approximately 386,000 records for each user and a total of seven users.

J48 Decision Tree is one of the most popular machine learning algorithms for tree structure algorithms because it is simple, accuracy, and fast, which is suggested in the literature, as well as using the least amount of memory. This was true when we were analyzing the algorithm. The J48 decision tree was the next best algorithm for accuracy which suggests that the efficiency will be lower. The averaged time taken to build the model for our J48 decision tree was 3.91 seconds. Again this is pretty fast for how much accuracy J48 decision tree received. J48 decision tree received approximately 15% better accuracy but it only took approximately 3 seconds longer to build the model. J48 decision tree received an averaged total execution time of 35.66 seconds for approximately 386,000 records for each user. When comparing algorithms, this research

recommends J48 decision tree because it is a great balance for accuracy and efficiency. If you want good results in a quick amount of time then J48 would be the way to go.

Random Forest was the most accurate algorithm used in this research, and based off the literature, it will be the slowest. This statement is accurate. In fact, random forest was consistently slower than all the other algorithms, whether we used one subject's data or all of the subjects combined. The averaged total time taken to build a model for random forest was 31.5 seconds. This is approximately 10 times longer than J48 but the random forest accuracy results are only 2% higher than J48. The averaged total execution time for random forest is 292.07 seconds. Again, this is approximately 10 times longer than J48 decision tree. Figure 61 below shows a graph representing the efficiency for each of the four algorithms analyzed in this research. The orange line is the total execution time and the blue line is the total time taken to build the model.

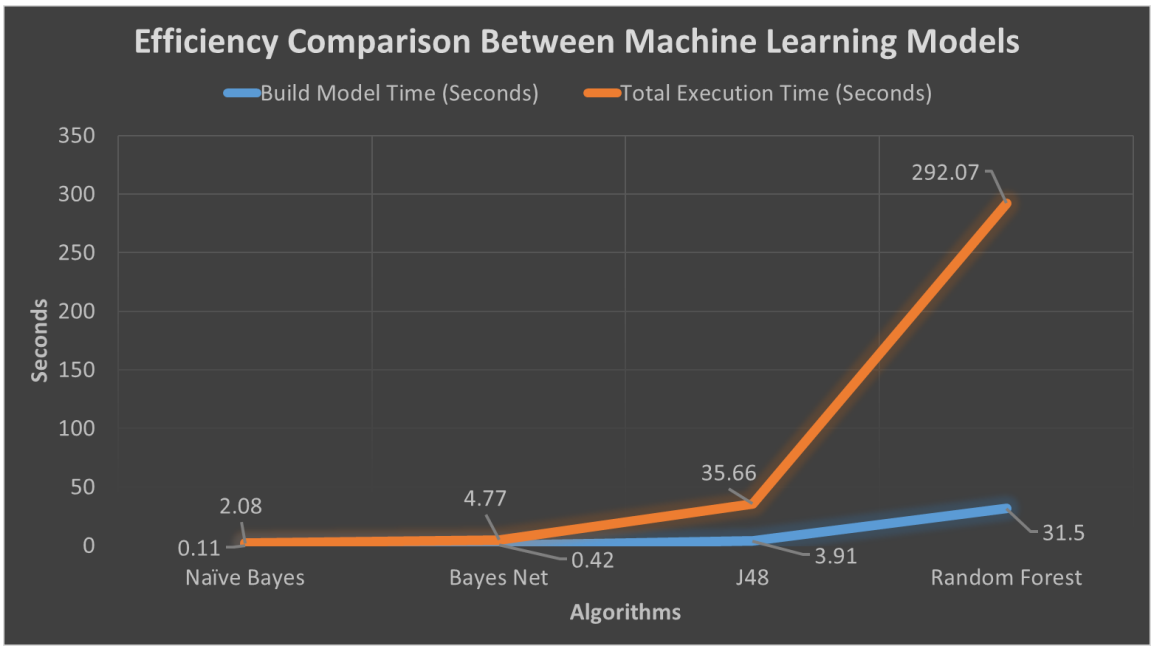

Figure 61. A visual representation for the efficiency comparison between machine learning models. It is shown Naïve Bayes is the fastest, however it received the lowest accuracy. Random Forest was the slowest algorithm but it received the highest accuracy. J48 is a great alternative receiving a similar accuracy to Random Forest with a 10 times faster total execution time.

Just like with accuracy, we tested the efficiency with one subject one model and one model across all subjects. This is to see whether we can build a generic model that will work across all subjects. Figure 62 below shows the efficiency comparison between models across subjects. The orange line represents the total execution time which includes the time taken to build the model, train, test, and validate the model. All times are averaged among 10 builds. Naïve Bayes receives an averaged time taken to build the model of 0.36 seconds and an averaged total execution time of 6.5 seconds. This was the quickest algorithm but also the one with the lowest accuracy. Next, Bayesian Network received the third best efficiency result with an averaged time taken to build the model of 2.67 seconds and an averaged total execution time of 38.61 seconds.

The second best algorithm for efficiency is J48 decision tree which is also the algorithm with the second best accuracy. J48 reported an averaged time taken to build the model of 34.53 seconds and an averaged total execution time of 601.41 seconds (approximately 10 minutes for all 7 users and approximately 3 million samples). Random forest received the slowest efficiency but received the best accuracy. Random forest averaged time taken to build the model was 221.02 seconds (approximately 3.5 minutes) and an averaged total execution time of 2,208.8 seconds (approximately 36.5 minutes). For the model built across all users, there were approximately 3 million records an averaged of approximately 386,000 records for each user.

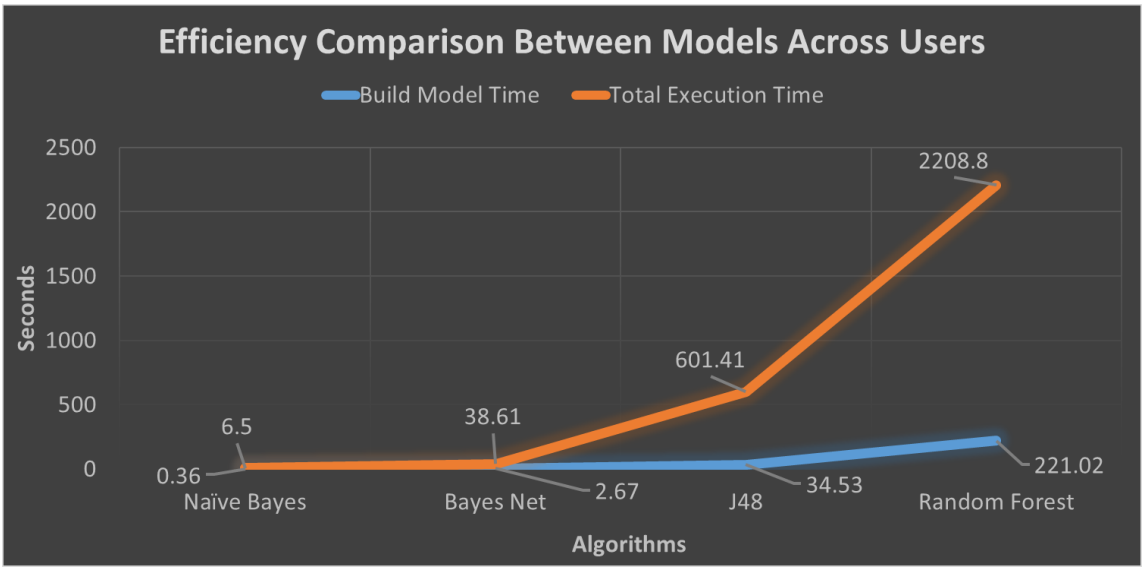

Figure 62. A graph showing the efficiency results compared across the four machine learning algorithms: Naïve Bayes, Bayesian Network, J48 Decision Tree, and Random Forest. These results are consistent with the one subject one model efficiency. Naïve Bayes has the best efficiency with the lowest accuracy and Random Forest has the worst efficiency with the best accuracy. J48 is a good alternative, getting great accuracy with pretty good efficiency.

Random forest produced the highest accuracy and kappa values. The kappa value was approximately 5% higher than J48 decision tree. However, the efficiency was approximately 10 times longer than J48. So a decision needs to be made which one to use between J48 and Random Forest. In this research, we would suggest using J48 decision tree. J48 is a great alternative to Random Forest because it received a similar accuracy to Random Forest with a ten times faster total execution time. In our opinion, J48 is a great algorithm to get quick results in order to determine some intermediate results. If time is not of the essence, then Random Forest should produce a higher accuracy. The dataset and time constraints on the project will be the best determinate of which machine learning algorithm to choose. If one is simply unsure, a wise decision will be to go with J48 decision tree.

7. DISCOVERIES AND KNOWLEDGE GAINED

In this research, we built, developed, and evaluated the first continuous prediction pipeline of mood dysregulation that can be consistently used in natural environments. I am very grateful to work on this project and spread the knowledge that we obtained. There was a lot of knowledge learned about properly comparing machine learning algorithms, finding trends to detect mood dysregulation, performing proper research, which has shaped us to be the computer scientist we are today. By sharing knowledge obtained in this research, I hope to influence others to continue this research project because there were many findings which makes it a great project for future work.

First, and one of the most important findings, is creating time categories for each day improves the accuracy of AMD by more than 10%. It is proven in this research that mood dysregulation happens at similar times of the day for the same subject. To help the machine learning algorithms, we also created another time attribute, Hour of Day, which was used in the final predictions. Both of the time attributes made were the most valuable attributes for detecting mood dysregulation. Before time was included, we obtained approximately 83% accuracy and after we included time attributes we obtained approximately 94% accuracy for a J48 decision tree with a balanced target class using resampling without replacement. This increase was consistent across our one subject one model experiments and with our one model across all subjects' experiments.

After adding time, we were able to build, train, and test a model that can consistently obtain an accuracy of 93.31% with a kappa value of 86.62% only taking an averaged total execution time of 35.66 seconds which consists of approximately 386,000 samples for each user. After running the model several hundred times, we started to find patterns in the attributes with the most

significance seen in the structure of the decision tree. Therefore we decided to run some feature selection algorithms. We found 7 attributes to have the most significance which includes: Time Categories, Inspiration, Expiration, Heart Rate, Hour of Day, Breathing Rate, and Minute Ventilation. Once these 7 attributes alone were used to train and test a J48 decision tree, we obtained a small increase in accuracy results consisting of 94.08% accuracy and kappa results slightly increasing to 88.16%. This is approximately 0.7% increase in accuracy and a 1.54% increase in kappa. These were the best results we could obtain using a J48 decision tree, with the same samples as the original raw data, using a balanced target class without replacement, using one model across all users of the study.

Using the raw data from Hexoskin achieved a higher accuracy when compared with using the cleaned data. When the term raw data is used here, it is technically not raw. Hexoskin does some pre-processing on the physiological sensors before we can get the data. For example, they average the heart rate among the last 16 heart beats. The increase in accuracy was low, approximately 1%, however it did exist and should be reported. Since the raw data received better results, we used the raw data throughout the model comparisons and our approach of one subject one model versus one model across all subjects. This proves that our cleaning pipeline was removing valuable information and Hexoskin already pre-processes the data well enough for predicting mood dysregulation using machine learning algorithms.

When comparing the one subject one model approach versus the one model across all subjects, we noticed the results were consistent between the two approaches. This means the machine learning model's accuracy and efficiency were consistent across the single user model and seven user model. Since these results were consistent, we could analyze the machine learning algorithms and recommend our model of choice. J48 Decision Tree would be our model of choice

because it is simple, accurate, and fast. J48 received a similar accuracy as Random Forest, our best model for accuracy, with an approximate difference of 2%, however, J48 was 73% faster than random forest and had a 10 times faster total execution time. Therefore, if time is an important factor in the analysis, we would recommend using a J48 decision tree because it is a great algorithm to get quick results with high accuracy.

When analyzing attributes, for the one model across all subjects, the most important attribute is Time Category. The second most important attribute for one model across all subjects is Inspiration. When experimenting on a one subject one model approach, the most important attribute is Hour of Day. The second most important attribute for one subject one model is Inspiration. When we were not considering time variables, on a one model across all subjects approach, the most important attribute is Inspiration and the second most important attribute is Heart Rate. Therefore, out of all the physiological measurements and only physiological measures, Inspiration is the most valuable measure for mood and heart rate is the second most valuable measure for predicting mood.

8. FUTURE WORK

Since we started this research and the machine learning analysis with AMD, more users have completed the study, therefore there is more data to be analyzed. The research performed in this Thesis can be duplicated on the new users that entered the study and thus can generate comparable results to be experimented on. This research only analyzes the times where the user initiated self-prompts for mood dysregulation, however mood dysregulation can be detected in many of the other surveys taken by the subjects, e.g. Random Prompt Surveys. If I had more time, I would have used AMD to predict mood dysregulation in the Random Prompt Surveys which is a good area for future work. There are many more positive mood dysregulation samples in the random surveys and the target class will be more balanced. With more samples, the results could be better and other machine learning algorithms can be tested. There are 62,353 Random Prompt Surveys out of 69,316 total surveys analyzed in this research, which is a lot more than the 5,601 Mood Dysregulation surveys analyzed in this research.

In our research lab, there are three similar studies happening simultaneously consisting of the Mood Toolkit Study, SLU HIV Study, and Alcohol Craving Study. AMD's pipeline can be used on the other datasets in order to obtain results and compare machine learning algorithms on different datasets. The Mood Toolkit Study is taking a look at mood dysregulation just like this study, the SLU HIV study is analyzing alcohol and drug usage with HIV positive subjects, and the Alcohol Craving Study is looking at alcohol use and trying to use machine and deep learning in order to predict alcohol use and cravings. AMD can be used on all three of these studies and can be compared with other pipelines being developed for the same study.

Lastly, AMD's pipeline is in the cloud running on machines with unlimited resources. Since J48 efficiency is so good with high accuracy, it would be a great algorithm to implement on a

mobile device. Therefore, future work could be done in order to implement AMD on a mobile device with limited resources. Since J48 Decision Tree has a Java implementation, it would be great research to try to implement AMD on an Android device while allowing the user to run the application all day using minimal battery power. In addition, instead of using machine learning, deep learning could be applied in AMD's pipeline to see if better results could be obtained. The deep learning algorithms' results could be compared with the current machine learning algorithms in AMD. If better results are obtained, then deep learning algorithms could also be implemented on a mobile device to predict mood dysregulation on the fly without uploading data to the cloud.

9. REFERENCES

[1] B. S. McEwen, "Protection and Damage from Acute and Chronic Stress: Allostasis and Allostatic Overload and Relevance to the Pathophysiology of Psychiatric Disorders," *Annals of the New York Academy of Sciences,* vol. 1032, pp. 1-7, 2004.

[2] K. Plarre, A. Raij, S. M. Hossain, A. A. Ali, M. Nakajima, M. al'Absi, E. Ertin, T. Kamarck, S. Kumar, M. Scott, D. Siewiorek, A. Smailagic and L. E. Wittmers, Jr. , "Continuous Inference of Psychological Stress from Sensory Measurements Collected in the Natural Environment," in *Information Processing in Sensor Networks (IPSN) 2011 10th International Conference* , Chicago, IL, 12-14 April 2011.

[3] J. T. Cacioppo and L. G. Tassinary, Principles of psychophysiology: Physical, social, and inferential elements, New York, NY: Cambridge University Press, 1990, pp. 216-252.

[4] H. Ursin and R. Murison, "Classification and description of stress," *Neuroendocrinology and psychiatric disorder,* pp. 123-132, 1984.

[5] R. Rosmond, M. F. Dallman and P. Bjorntorp, "Stress-Related Cortisol Secretion in Men: Relatoinships with Abdominal Obesity and Endocrine, Metabolic and Hemodynamic Abnormalities," *Journal of Clinical Endocrinology and Metabolism,* vol. 83, no. 6, 2009.

[6] B. S. McEwen and E. Stellar, "Stress and the Individual: Mechanisms Leading to Disease," *Archives of Internal Medicine,* vol. 153, no. 18, pp. 2093-2101, 1993.

[7] G. P. Chrousos and P. W. Gold, "The Concepts of Stress and Stress System Disorders: Overview of Physical and Behavioral Homeostasis," *The Journal of the American Medical Association (JAMA),* vol. 267, no. 9, pp. 1244-1252, 1992.

[8] M. Al'Absi and D. K. Arnett, "Adrenocortical responses to psychological stress and risk for hypertension," *Biomedicine and Pharmacotherapy,* vol. 54, no. 5, pp. 234-244, 2000.

[9] R. Rosmond and P. Bjomtorp, "Endocrine and metabolic aberrations in men with abdominal obesity in relaiton to anxio-depressive infirmity," *Metabolism,* vol. 47, no. 10, pp. 1187-1193, 1998.

[10] A. Steptoe, G. Fieldman, O. Evans and L. Perry, "Cardiovascular Risk and Responsivity to Mental Stress: The Influence of Age, Gender and Risk Factors," *European Journal of Preventive Cardiology,* vol. 3, no. 1, pp. 83-93, 1996.

[11] J. P. Henry, "Stress, neuroendocrine patterns, and emotional response," *American Psychological Association,* pp. 477-496, 1990.

[12] M. Al'Absi, Stress and Addiction: Biological and Psychological Mechanisms, Academic Press, 2007.

[13] M. A. Enoch, "Pharmacogenomics of Alcohol Response and Addiction," *American Journal of Pharmacogenomics,* vol. 3, no. 4, pp. 217-232, 2003.

[14] M. A. Enoch, "Genetic and Environmental Influences on the Development of Alcoholism," *Annals of the New York Academy of Sciences,* vol. 1094 Resilience in Children, pp. 193-201, 2006.

[15] R. W. Carpenter and T. J. Trull, "Components of Emotion Dysregulation in Borderline Personality Disorder: A Review," *Current Psychiatry Reports,* vol. 15, 2013.

[16] P. Sun, N. M. Wergeles and C. Zhang, "ADA - Automatic Detection of Alcohol Usage for Mobile Ambulatory Assessment," in *Smart Computing (SMARTCOMP), 2016 IEEE International Conference*, 2016.

[17] U. Ebner-Priemer, "Home Ambulatory Assessment," 2014. [Online]. Available: http://www.saa2009.org/.

[18] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in *Handheld and Ubiquitous Computing*, Springer Berlin Heidelberg, 2001, pp. 304-307.

[19] T. E. Starner, "Wearable Computing and Contextual Awarenesss," Massachusetts Institute of Technology (MIT) Media Lab, 1999.

[20] R. Shi, C. Zhang, H. Wang, P. Sun, T. Trull and Y. Shang, "mAAS - A Mobile Ambulatory Assessment System for Alcohol Craving Studies," *Computer Software and Applications Conference (COMPSAC),* pp. 282-287, 2015.

[21] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, P. Klasnja, K. Koscher, J. A. Landay, J. Lester, D. Wyatt and D. Haehnel, "The Mobile Sensing Platform: An Embedded Activity Recognition System," *IEEE Pervasive Computing,* vol. 7, no. 2, pp. 32-41, 2008.

[22] H. Lu, W. Pan, N. D. Lane, T. Choudhury and A. T. Campbell, "SoundSense: scalable sound sensing for people-centric applications on mobile phones," *MobiSys Proceedings of the 7th international conference of Mobile systems, applications, and services,* pp. 165-178, 2009.

[23] E. W. Boyer, R. Fletcher, R. J. Fay, D. Smelson, D. Ziedonis and R. W. Picard, "Preliminary Efforts Directed Toward the Detection of Craving of Illicit Substances: The iHeal Project," *Journal of Medical Toxicology,* vol. 8, no. 1, pp. 5-9, 2012.

[24] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng and A. T. Campbell, "Sensing meets mobile social networks: the design, implementation and evaluation of the CenceMe application," *SenSys Proceedings of the 6th ACM conference on Embedded network sensor systems,* pp. 337-350, 2008.

[25] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West and P. Boda, "PEIR, the personal environmental impact report, as a platform for participatory sensing systems research," *MobiSys Proceedings of the 7th international conference on Mobile systems, applications, and services,* pp. 55-68, 2009.

[26] S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, I. Smith and J. A. Landay, "Activity sensing in the wild: a field trial of ubifit garden," *CHI Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* pp. 1797-1806, 2008.

[27] A. Thiagarajan, L. Ravindranath, K. LaCurts, S. Madden, H. Balakrishnan, S. Toledo and J. Eriksson, "VTrack: accurate, energy-aware road traffic delay estimatoin using mobile phones," *SenSys Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems,* pp. 85-98, 2009.

[28] E. L. v. d. Broek, V. Lisy, J. H. Janssen, J. H. D. M. Westerink, M. H. Schut and K. Tuinenbreijer, "Affective Man-Machine Interface: Unveiling Human Emotions through Biosignals," *Biomedical Engineering Systems and Technologies,* pp. 21-47, 2009.

[29] J. Healey, L. Nachman, S. Subramanian, J. Shahabdeen and M. Morris, "Out of the Lab and into the Fray: Towards Modeling Emotion in Everyday Life," in *Pervasive Computing*, 2010.

[30] J. A. Healey and R. W. Picard, "Detecting Stress During Real-World Driving Tasks Using Physiological Sensors," *IEEE Intelligent Transportation Systems Society,* vol. 6, no. 2, pp. 156-166, 2005.

[31] J. Kim and E. Andre, "Emotion recognition based on physiological changes in music listening," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, no. 12, pp. 2067-2083, 2008.

[32] S. D. Kreibig, "Autonomic nervous system activity in emotion: A review," *Biological Psychology,* vol. 84, no. 3, pp. 394-421, 2010.

[33] S. D. Kreibig, F. H. Wilhelm, W. T. Roth and J. J. Gross, "Cardiovascular, electrodermal, and respiratory response patterns to fear- and sadness-inducing films," *Psychophysiology,* vol. 44, no. 5, pp. 787-806, 2007.

[34] M. Myrtek and G. Brugner, "Perception of emotions in everyday life: studies with patients and normals," *Biological Psychology,* vol. 42, no. 1-2, pp. 147-164, 1996.

[35] P. Rainville, A. Bechara, N. Naqvi and A. R. Damasio, "Basic emotions are associated with distinct patterns of cardiorespiratory activity," *International Journal of Psychophysiology,* vol. 61, no. 1, pp. 5-18, 2006.

[36] C. L. Stephens, I. C. Christie and B. H. Friedman, "Autonomic specificity of basic emotions: Evidence from pattern classification and cluster analysis," *Biological Psychology,* vol. 84, no. 3, pp. 463-473, 2010.

[37] Y. Shi, M. H. Nguyen, P. Blitz, B. French, S. Fisk, F. D. l. Torree, A. Smailagic, D. P. Siewiorek, M. Al'Absi, E. Ertin, T. Kamarck and S. Kumar, "Personalized Stress Detection from Physiological Measurements," *International symposium on quality of life technology,* pp. 28-29, 2010.

[38] P. A. Fournier, 2016. [Online]. Available: http://www.hexoskin.com/.

[39] L. Constantine and H. Hajj, "A survey of ground-truth in emotion data annotation," *Pervasive Computing and Communications Workshops (PERCOM Workshops) IEEE International Conference,* pp. 697-702, 2012.

[40] G. Miller, "The Smartphone Psychology Manifesto," *Perspectives on Psychological Science,* vol. 7, no. 3, pp. 221-237, 2012.

[41] S. M. Hossian, A. A. Ali, M. Rahman, E. Ertin, D. Epstein, A. Kennedy, K. Preston, A. Umbricht, Y. Chen and S. Kumar, "Identifying drug (cocaine) intake events from acute physiological response in the presence of free-living physical activity," *IPSN Proceedings of the 13th international symposium of Information processing in sensor networks,* pp. 71-82, 2014.

[42] W. James, The principles of psychology, vol. 1, New York, NY: Henry Holt and Co, 1890.

[43] P. A. Fournier, 2016. [Online]. Available: http://www.hexoskin.com/pages/key-metrics-delivered-by-hexoskin.

[44] Hexoskin Wearable Body Metrics, "Hexoskin Health Research," Hexoskin, 2016. [Online]. Available: https://cdn.shopify.com/s/files/1/0284/7802/files/How_it_works_2014-09-15_EN_Researchers.pdf?11535. [Accessed 4 December 2016].

[45] D. V. Sheehan, "The Mini-International Neuropsychiatric Interview (MINI): the development and validation of a structured diagnostic psychiatric interview for DSM-IV and ICD-10.," *Journal of clinical psychiatry,* pp. 22-33, 1998.

[46] B. Pfohl, Structured Interview for DSM-IV Personality (SIDP-IV), Washington, DC: American Psychiatric Press, Inc., 1997.

[47] Wikipedia, "Wikipedia The Free Encyclopedia QRS Complex," Wikipedia Foundation, Inc., 17 November 2016. [Online]. Available: https://en.wikipedia.org/wiki/QRS_complex. [Accessed 4 December 2016].

[48] P. A. Fournier, "Biometric Resources Hexoskin API," 2016. [Online]. Available: https://api.hexoskin.com/docs/index.html.

[49] W. G. Jacoby, "Loess: a nonparametric, graphical tool for depicting relationships between variables," *Electoral Studies,* pp. 577-613, Dec 2000.

[50] MathWorks, "Documentation," The MathWorks, Inc. , 2016. [Online]. Available: https://www.mathworks.com/help/curvefit/smoothing-data.html. [Accessed 4 December 2016].

[51] Star Trek, "Residual Analysis in Regression," 2016. [Online]. Available: http://stattrek.com/regression/residual-analysis.aspx?Tutorial=AP. [Accessed December 2016].

[52] MathIsFun, "Standard Deviation Formulas," 2014. [Online]. Available: http://www.mathsisfun.com/data/standard-deviation-formulas.html. [Accessed December 2016].

[53] H. Sarker, M. Tyburski, M. Rahman, K. Hovsepian, M. Sharmin, D. Epstein, K. L. Preston, C. D. Furr-Holden, A. Milam, I. Nahum-Shani, M. al'Absi and S. Kumar, "Finding Significant Stress Episodes in a Discontinuous Time Series of Rapidly Varying Mobile Sensor Data," *CHI Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems,* pp. 4489-4501, 2016.

[54] Florida State University, "Linear interpolation," [Online]. Available: http://www.eng.fsu.edu/~dommelen/courses/eml3100/aids/intpol/. [Accessed December 2016].

[55] WikiPedia, "Wikipedia The Free Encyclopedia Linear Intrepolation," Wikipedia Foundation, Inc., 2016. [Online]. Available: https://en.wikipedia.org/wiki/Linear_interpolation. [Accessed December 2016].

[56] The Pennsylvania State University, "Stat 509 Desgin and Analysis of Clinical Trials Lesson 18: Correlation and Agreement," 2016. [Online]. Available: https://onlinecourses.science.psu.edu/stat509/node/162.

[57] Wikipedia, "Confusion Matrix," Wikipedia Foundation, Inc. , 2016. [Online]. Available: https://en.wikipedia.org/wiki/Confusion_matrix. [Accessed December 2016].

[58] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explorations Newsletter,* vol. 11, no. 1, pp. 10-18, 2009.

[59] J. R. Quinlan, C4. 5: Programs for Machine Learning, San Mateo, California: Morgan Kaufmann, 1993.

[60] C. Petri, "Decision Trees," Computer Science Department of Babes-Bolyai University Romania, Cluj Napoca, 2010.

[61] M. Walker, "Random Forest Algorithm," 24 September 2013. [Online]. Available: http://www.datasciencecentral.com/profiles/blogs/random-forests-algorithm.

[62] J. R. Landis and G. G. Koch, "The Measurement of Observer Agreement for Categorical Data," *Biometrics,* vol. 33, no. 1, pp. 159-174, March 1977.

[63] J. L. Fleiss, B. Levin and M. C. Paik, Statistical methods for rates and proportions, New Jersey: John Wiley & Sons, Inc. , 2013, pp. 598-626.

[64] R. W. Stephenson, A. G. Froelich and W. M. Duckworth, "Using Resampling to Compare Two Proportions," *Teaching Statistics An International Journal for Teachers,* vol. 32, no. 3, pp. 66-71, 5 August 2010.

[65] R. Barlow, SLUO Lectures on Statistics and Numerical Methods in HEP Lecture 6: Resampling and the Bootstrap, Manchester: Lecture notes, 2000.

[66] M. A. Hall, "Correlation-based Feature Selection for Machine Learning," Hamilton, 1999.

[67] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence,* pp. 273-324, 1997.

[68] P. Langley, "Estimating continuous distributions in Bayesian classifiers," *Artificial Intelligence,* pp. 338-345, 1995.

[69] X. Wu, V. Kumar, R. J. Quinlan and J. Ghosh, "Top 10 algorithms in data mining," *Knowledge and Information Systems,* pp. 1-37, 2008.

[70] L. Breiman, "Random Forests," *Machine Learning,* pp. 5-32, 2001.

[71] B. Rohrer, "How to choose algorithms for Machine Learning," Microsoft Azure, 2016. [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice. [Accessed December 2016].

[72] E. E. Ghiselli, "Theory of psychological measurement," New York : McGraw-Hill, 1964.

[73] Pennsylvania State University, "STAT509 Design and Analysis of Clinical Trials," PennState Eberly College of Science , 2016. [Online]. Available: https://onlinecourses.science.psu.edu/stat509/node/162. [Accessed December 2016].

10. VITA

Nickolas M. Wergeles was born in Torrance, California in 1989. He started to grow up there but in 1992, the Los Angeles (LA) riots occurred due to reactions of the Rodney King trial. This caused arsons, shootings, and many tragic situations. Nickolas' parents decided it was not a good place to raise a family, therefore they moved to Missouri where they knew some family friends.

In Missouri, Nickolas continued to grow up. In elementary, Nickolas started playing percussion for music class. He also started competitive sports such as Basketball, Football, and Bicycle Motocross (BMX). Later in high school, Nickolas continued his musical talents and competitive sports participating in high school band, football, track-&-field, and BMX. Nick really started progressing in Mathematics and sports. During his senior year, he obtained track & field records, was one of the team captains for his football team, was the lead snare drum percussionist, and was taking several dual-credit college classes. Towards the end of his high school career, Nick's Uncle told him, "Since you are good with mathematics and you like technology, you should consider studying Computer Science." This changed Nick's life forever and pointed him in the right direction.

During Nick's undergraduate degree at the University of Central Missouri (UCM), he started taking computer science (CS) courses. Nick received serval scholarships such as the National Science Foundation (NSF) scholarship, S.M.A.R.T. Grant, and the academic competitiveness grant. Towards the end of his freshman year, Nick was thinking about switching degrees. He started the theory of CS and thought, "This is not for me," since the outcome of his results were small. However, Nick went to a smaller University, thus he was close with his professors, which encouraged him to continue. Therefore, Nick continued his CS degree and created one of the best applications his software engineering professor had ever seen, at that

time. Nick created a hybrid application which used Google maps, calculated the fastest walking route to each building on the UCM campus, and would open fire-escape plans for a particular building to help guide the user along the quickest route to class, all the way to the classroom door.

During the summer of 2011, Nickolas applied for a research developer position at the University of Missouri where he was one of 10 selected out of more than 145 applicants in the Midwest region. There he developed and designed a remote object localization system using Android devices. Nick helped to server as a team leader and collaborator. He recorded multimedia weekly briefings and conducted code review for the CS department. At the end of the research experience, the project he worked on received first place out of all the projects at Mizzou. This summer really opened Nick's eyes to see he needed to go to Graduate school and he was really interested in research. During Nick's last undergraduate semester, he became a teaching assistant (TA) for multiple college algebra courses. Later, he finished his degree with a 3.8 GPA, mathematics minor, and graduated cum laude. Nick was on the Dean's list 8 out of 9 semesters. Nickolas had some jobs lined up, however his family convinced him he needed to get his graduate degree.

Nickolas applied to the University of Missouri (Mizzou) in Columbia. During graduate school, Nickolas became a TA for computer organization and assembly language. The students started liking him and reported good things to the CS chair. He continued to do outstanding work and became a research assistant (RA) for Professor Yi Shang. Nick was a TA and RA at the same time, however, later he decided to only continue the RA, so he could focus on his research and course work. Nick was involved with four concurrent research projects and started mentoring undergraduate research during the summers. He also started the collaboration between the Mizzou CS and Missouri Conservation Departments (MDC). While working on research projects

for his RA, Nick started research for his thesis. His thesis research was a collaboration between the Mizzou CS and psychology department studying emotional dysregulation using artificial intelligence.

Towards the end of his graduate school career, Nickolas received Mizzou's 2016 Most Outstanding Master's Student Award for the College of Engineering. He was involved with honor societies such as Upsilon Pi Epsilon (UPE) and the Association for Computing Machinery (ACM) where he was Treasure. Later, he had two research papers accepted in the Institute of Electrical and Electronics Engineers (IEEE) Smart Computing 2016 International Conference. Continuing, Nickolas was asked to be the Instructor for the Web Application Software Engineering course at Mizzou. During the Fall 2016 semester, there were approximately 110 students and only one student dropped from the course. Nick received very positive reviews and seemed to have a strong connection with the students. Nick led and controlled the rolls of three TAs and created the curriculum for the course, all while finishing his thesis.

Right now, Nickolas will attend the December 2016 graduate commencement with a GPA of 3.6. There he will graduate with UPE honors and continue for his PhD in Computer Science. With a PhD degree, Nickolas will pursue teaching and try to become a Professor in Computer Science. During Nick's time at graduate school, he learned that he wanted to do research and he had a positive outcome when teaching students. Nick is so grateful for all the advice he has received from his family and all of his professors. Without Professor Yi Shang, Nick would not be the person he is today. Professor Yi Shang guided, taught, and mentored Nick. Nick also knows that his family was the most positive force in his life. Without his family, Nick would not even be here. Family is the most important thing in this world. Friends will come and go but family will be forever, and without family we have nothing.