USING THE APPLE WATCH FOR DETECTION AND PREVENTION OF SUDDEN UNEXPECTED DEATH IN EPILEPSY DURING SLEEP

BY

DAVID RAYMOND CONNELL

Submitted in partial fulfillment of the
requirements for the degree of
Master of Science in Biomedical Engineering
in the Graduate College of the
Illinois Institute of Technology

Approved _______________________
Advisor

Chicago, Illinois
December 2017

ACKNOWLEDGMENT

I would like to thank Dr. David Mogul for his suggestion of the project and guidance throughout my thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Symbol	Definition
$\hat{\sigma}^2$	Variance estimator.
I	Heart beat interval
$\bar{I}$	Average heart beat interval between data points collected by the Apple Watch
L	Standard deviation of consecutive heart beat intervals
T	Total standard deviation of all heart beat intervals
HR	Heart rate

ABSTRACT

Sudden unexpected death in epilepsy (SUDEP) is a serious concern on the epileptic community. SUDEP is estimated to make up 12-17% of all epilepsy related deaths; however little is known about the disease or how to prevent it [4]. Limitations in current health monitoring methods prevent the frequent data gathering needed to record biometrics during a SUDEP event. Without this data many studies related to SUDEP are restricted to using witness accounts and post-mortem diagnoses only, restricting research progress to a slow pace at best. To gain better insight on SUDEP, new methods of data collection are needed. Wearable devices, such as the Apple Watch, may provide a medium for this data gathering. The Apple Watch comes with an accelerometer and heart rate monitor which can be run with little inconvenience to the user. An app, described in this paper, has been written for this device to record motion data from the accelerometer along with heart rate in epileptics during sleep. Since 94% of SUDEP events occur during sleep, monitoring is designed for sleep only as lively movements throughout the day may set off the alarm. Since the limited SUDEP research has not provided the data required to construct SUDEP specific detection methods, indirect methods are needed. In witnessed cases, between 75 and 80% of SUDEP victims had a seizure immediately prior to death making seizure onset a potential SUDEP predictor [9]. The app enlists two independent detection methods based on seizure onset, one for each of the monitored metrics. The accelerometer's detection is based on the variance in a user's movement. During normal sleep, movements are generally small and kept to a minimum creating a low variance. In the event of a convulsive seizure, rapid movements increase the variance to beyond healthy levels. An event is detected if the variance in any of the spacial dimensions (x, y, or z) reaches a threshold value. The heart rate method uses a more complex algorithm. The app produces a cardiac sympathetic index (CSI) which is a measure of heart rate variability, comparing the total beat-interval variance to

the variance of consecutive beat-intervals [5]. In the moments approaching a seizure total variance becomes large compared to inter-beat interval variance [5]. Taking this into account the app searches for increases in CSI to determine seizure onset. Unfortunately, the watch's heart rate monitor does not provide continuous data but rather an average heart rate over the past few seconds. Preventing the data collected from being used to calculate true CSI values. Although there is some trace of a trend in the watch's approximate CSI values which may be useful. Future studies are still required to determine if this trend will be appropriate for seizure detection, before it can be used in the app. Along with detecting SUDEP events, the app sends recorded data to a database for study. A python module has been written to provide a simple interface for interacting with this database. It is the desire these collected metrics will provide the needed data to improve the current detection methods and provide insight on the cause of SUDEP.

CHAPTER 1
INTRODUCTION

Sudden unexpected death (SUD) is the unlikely and unexplained sudden death of an otherwise healthy individual. Of special interest is SUD in the epileptic population: sudden unexpected death in epilepsy (SUDEP). SUDEP occurs at a higher frequency than SUD in the general population. After controlling for co-morbidities, Hoist et al. found SUD to be 16.3 times as prevalent in epileptics compared to the general public [3]. Furthermore, SUDEP is the cause of between 12-17% of all epilepsy related deaths. In high risk patients that number may be as large as 40% [4].

Despite the frighteningly high risk of SUDEP there is still much left unknown about the disease, leaving the underlying mechanism of action for SUDEP a mystery. It has been suggested that SUDEP may be the result of seizure related hypoventilation or heart rate dysrrhythmia [9]. However, Lhatoo et al. have brought attention to cases of SUDEP which do not show evidence of a related seizure and do not appear to be caused by cardiac distress [7]. The differences between cases lead to the possibility there may not be a single mechanism of action but, various physiological pathways which might cause SUDEP [9].

This insufficient understanding of SUDEP has prevented an accurate method for predicting the onset of SUDEP. Some risk factors have been proposed including: missed anti-epileptic drug doses, having at least one seizure per month, and being between the ages of 20-45 [4]. However, while these may help determine who is at high risk to be affected by SUDEP, it has yet to be determined if these factors can help predict when an incident will occur.

As its name implies sudden unexpected death occurs without warning, making it difficult to study. It is rare for a victim to be in the hospital or monitored in some way during a SUDEP event. As a result, there is little biometric data detailing physiological changes during SUDEP. More data may shed light on methods for predicting and ultimately preventing SUDEP. Unfortunately, prohibitory costs and inconvenience have made regular monitoring of epileptics' biosignals unrealistic. Recently, however, wearable devices such as smart watches equipped with sensors, notably with accelerometers and heart rate monitors, have become increasingly present in the market, bringing a new, non-restrictive and relatively cheap, mode of biometric monitoring and recording.

These watches may be capable of aiding SUDEP research. This paper provides a description of the SUDEPmonitor app designed for the Apple Watch as part of my master's thesis research. This uses the watch's sensors to monitor an epileptic user's heart rate and motion during sleep in order to detect possible SUDEP events. If the watch detects indicators of SUDEP it will send an alert to a caregiver to check on the user. Additionally, the app will send accelerometer and heart rate data, along with event markers, to a database with open access for research purposes. Ideally, this will provide the needed data to better understand SUDEP and produce better prevention methods.

CHAPTER 2
DETECTION METHODS

SUDEPmonitor aims to detect, prevent, and record data on SUDEP. Due to a limited understanding of SUDEP, detection and prevention appear to be daunting tasks. There are, however, some leads which may provide a base for a detection algorithm. Events of SUDEP occur predominantly after a seizure: between 75% and 80% of witnessed cases were during or directly following a seizure [9]. It is therefore hypothesized: detecting seizures will detect the onset of many SUDEP cases. As such, seizures will be used as the basis for the SUDEPmonitor's detection algorithm. From the literature, two types of seizure detection algorithms stand out: movement based and heart rate based.

2.1 Accelerometer Based Detection

For seizure detection, the accelerometer is an obvious choice but requires the seizure to cause muscle spasms. Beniczky et al. used a simple threshold based algorithm for detecting generalized tonic-clonic seizures (GTCS) and found it detected nearly 90% of the GTCSs, thus showing the potential for an accelerometer based detection method [1].

Viewing accelerometer data from the author's sleep, (figure 2.2 top), it can be seen that the accelerometer signal is not necessarily zero at rest. The accelerometer detects the acceleration due to gravity as well as the acceleration due to movement. Because of this, the angle of the wrist may affect whether the same movement will set off the alarm when using a basic threshold system. Since the angle of the wrist is unlikely to provide information about seizure onset, the relationship between wrist angle and detection should be minimized. To do this, the difference of the accelerometer data (the $i_{th}+1$ point subtracted by the i_{th} point) was used in-place of the raw data.

When viewing the histogram for the difference of the signal from figure 2.2 top, the data appears to be well approximated as a zero mean Laplacian random variable (figure 2.1). The variance of this distribution was estimated using eq 2.1. The mean characteristics of the estimated variance over a series of three sleep tests can be found in table 2.1.

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{n}|x| \tag{2.1}$$

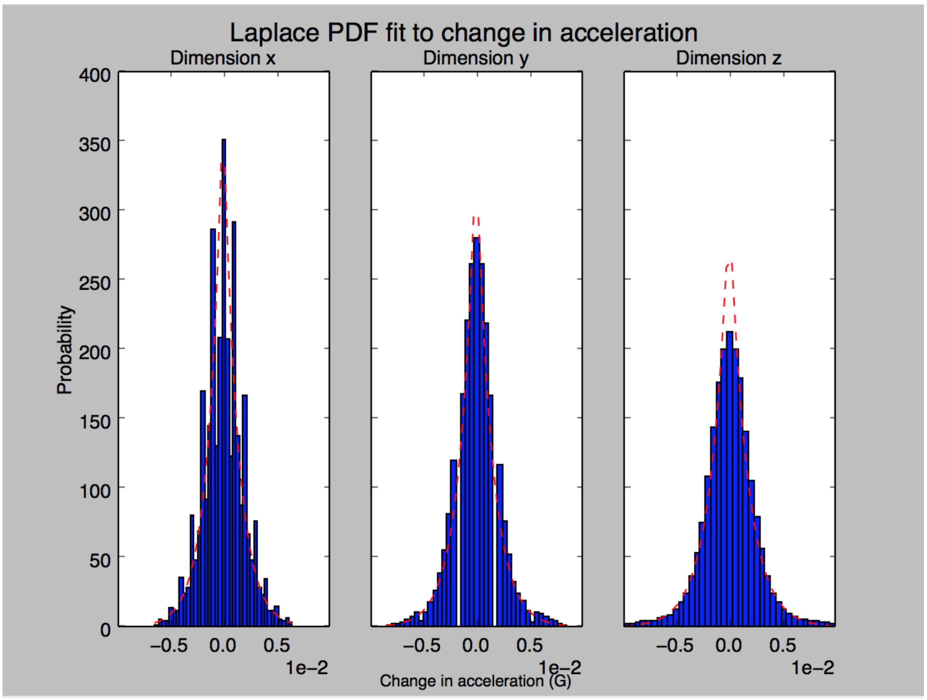


Figure 2.1. The histogram for the difference of the author's accelerometer data during sleep fit to a Laplace distribution.

Finding the distribution for the seizure state was not as easy due to a lack of seizure data. In an attempt to simulate a seizure, tests were performed where the arm was shaken rapidly during a SUDEPmonitor recording. The results found

Table 2.1. Average characteristics of the accelerometer variance for three of the author's sleep sessions.

Accelerometer axis	Max (G^2)	Min (G^2)	Average (G^2)	Variance (G^4)
X	2.9835e-01	3.1988e-07	6.1938e-04	6.2113e-05
Y	3.3800e-01	2.3347e-07	1.7000e-03	2.5191e-04
Z	5.1200e-01	3.4721e-07	1.0728e-03	2.0419e-04

the mean to be near zero under this case as well, possibly due to movements being equally likely to travel in a positive or negative direction. However, the variance for these tests was much greater as expected. Of course, since the tests were only on the scale of 10s of seconds, the high "seizure" activity made up a much greater fraction of the session than it would in a larger session, across the time-span of hours. It would be presumed that the full session would hide the high variance of the seizure behind the low-activity normal sleep. Since the interest is in predicting when a seizure is underway approximating the variance under the seizure case as the sample variance during only the seizure, rather than the whole session, will likely be more beneficial. As expected, this variance was much larger than the variance during control (non-seizure), allowing the use of sample variance to be used as a test statistic.

The detection algorithm was then set as a threshold test, continually comparing the sample variance from the previous 5 seconds against the threshold for each of the dimensions (x, y, and z). If the variance for any of the dimensions exceeds the threshold, the alarm is set-off. Both sleep data and fake seizure data were tested with this method (figure 2.2). No false alarms were produced during these tests while high activity was successfully identified during the fake seizure. The detection delay in figure 2.2 bottom is due to the sample variance using data from the previous 5 seconds.

The detection may not be as successful for real seizures, particularly ones which do not produce muscle spasms. Such a method may also be susceptible to false alarms during intense physical activity; however the app is intended for use during sleep, minimizing this potential downfall.

2.2 Heart Rate Based Detection

In addition to use of the accelerometer, a second detection method will use the watch's heart rate sensor to catch seizures missed by the previous method, including non-convulsive seizures. This method incorporates properties of the Lorenz plot, a plot reflecting the variance of beat intervals, to determine seizure state. The Lorenz plot technique has been capable of detecting seizures without relying on motion, allowing for identification of seizures without muscle spasm [5].

The Lorenz plot is a plot of the i_{th} R-R interval, I_i, against the i_{th} + 1 R-R interval, I_{i+1}. This plot provides two characteristics of heart rate variability (HRV): the standard deviation of the points' distances form the $I_i = I_{i+1}$ line (L) and the standard deviation of the points' distances from the axis perpendicular to the $I_i = I_{i+1}$ (T) (figure 2.3) [5, 2].

The value T may be interpreted as the variance of consecutive heart beat intervals while, L is the total variance for all the intervals. These characteristics can be calculated using equations 2.2 and 2.3 modified for simplicity from Piskorski et al. [8].

$$L = \sqrt{\frac{1}{2N}\sum_{i=1}^{N}[(x_i + y_i) - (\bar{x} + \bar{y})]^2} \tag{2.2}$$

$$T = \sqrt{\frac{1}{2N}\sum_{i=1}^{N}[(x_i - y_i) - (\bar{x} - \bar{y})]^2} \tag{2.3}$$

Seizures often cause a large total beat interval variance compared to sequential beat interval variances. This leads to defining the Cardiac Sympathetic Index (CSI) as L/T[5].

CSI values during seizures have been found to be about 1.5 to 2.9 times larger than CSI values during control (non-seizure) when calculating CSI from the last 50 heart beats (CSI_{50}) [5]. However, the small sample size of the study leaves uncertainty in the validity of the method. This method also requires a training period to determine an estimate of the individuals non-seizure CSI. Nonetheless, CSI appears to be promising for the detection of seizures.

Due to limited access to the watch's heart rate monitor, the beat-to-beat intervals cannot be correctly calculated using an Apple Watch at this time. At best heart rate is obtained from the watch approximately every 5 seconds. These data is then converted to the average beat-to-beat interval over the time from the previous data point using equation 2.4.

$$\bar{I}(t) = \frac{60\text{sec}}{HR} \tag{2.4}$$

Where $\bar{I}(t)$ is the average beat-to-beat interval at the given time, in seconds, and HR is the heart rate in beats per minute. This value will be used as a single point allowing for the production of pseudo-Lorenz plots and CSI values. This is not ideal and clearly removes useful time resolution compared to true CSI values. Figure 2.4 show's the Lorenz plots obtained using the watch's heart rate monitor have similar shape to the Lorenz plot calculated from a pulse signal recorded simultaneously with a pulse transducer. The pulse transducer measures the pressure changes in the thumb over time resulting from blood flowing with the heart's beat.

2.2.1 Beat detection for the pulse transducer. To find the beat-intervals from the pulse transducer's signal a static threshold technique was used. First, the initial 5 seconds of the signal were cut since there was a lot of movement at the start of recording. Then the signal was normalized by subtracting out the minimum of the signal followed by dividing by the new maximum of the signal making the range of the signal 0 to 1. Every point greater than the threshold was given a value of 1 and all points below the threshold were assigned a value of zero. The difference of this binary signal was computed. At this point, the values where the signal crossed from below to above the threshold had a value of 1 and every value that crossed from above to below had a value of -1, all other values were 0. The times of the peaks were considered to be the points in the time array that corresponded to the 1s of the difference signal.

While this does not produce the exact times of the peaks but, rather, the times where the signal crossed the threshold, due to the high slope near the pulse's peak and the similarity of sequential peaks—causing the $i_{th}+1$ peak to be marked in a similar location of the wave as the i_{th} peak—this appears to be adequate for finding the time between the peaks. Lastly the beat-to-beat intervals were the difference between the beat times. Due to differences in signals the thresholds were manually selected for each sample. This simple algorithm worked well for most of the signals (figure 2.5 top). Although, there were issues with a few of the signals (figure 2.5 middle).

In file 6, the smaller peak following the main peak, in some locations, was larger than the main peak at other times. This leads to either additional false beats being detected or beats being missed, no matter what the threshold was set to. To obtain better beat detection in this case, a second dynamic threshold method was applied. This method was the same as above, except the static threshold was replaced by a threshold calculated for each point individually based on a window around that point. This method was able to correct the issue with the static threshold method (figure 2.5 bottom). However, this method had it's own problems. In noisy areas of a waveform the threshold can change drastically across nearby points leading to additional beats being detected. Secondly, some samples had a few abnormally tall peaks, inflating the threshold to above the trailing and leading peaks, causing missed beats. To solve the problem, the threshold method was selected for the tests individually based on which produced the better result. Even with the two detection methods, artifacts due to moving the thumb during recording caused some of the tests to be too noisy to use. These were simply discarded.

2.2.2 Testing the validity of the Watch's heart rate monitor. Heart rate was recorded using the SUDEPmonitor app along with a pulse transducer for four sets of tests: sitting, standing, supine, and active. The active test was recorded after performing a series of high jumps. The four tests were selected to compare the watch's results under different heart conditions to the pulse transducer's results. Each test was performed three times for between three and five minutes. During processing some of the samples were removed due to high levels of noise.

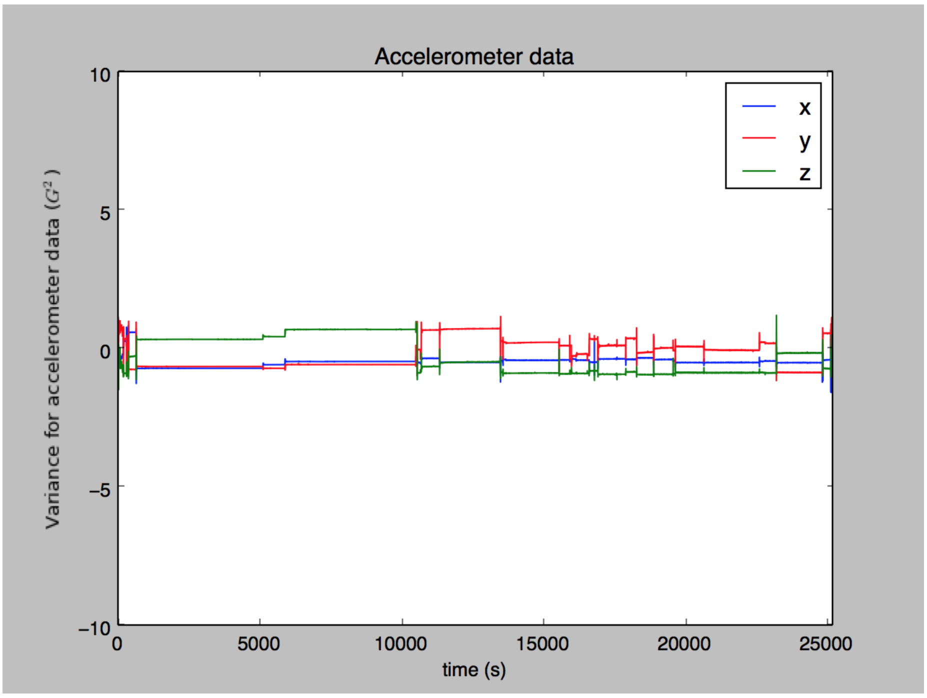
Table 2.2. Number of tests for each position after removing noisy files.

Test position	Number of tests	Total time (sec)
Sitting	2	440
Standing	2	385
Supine	2	355
Active	3	550

From figure 2.4 the two methods appear to give similar Lorenz plots. However, the watch's plots tend to miss some of the outlier terms which may be important for detecting seizure onset. Since the watch's heart rate monitor is the average of around 5 beats, it covers up unusually large beat-intervals surrounded by smaller intervals. As expected, at rest the beat-intervals vary around a central point (about 1 beat per second). After activity, the beat-intervals have a large total variance (L) compared to inter-beat variance (T). As the heart rate moves through a large range of frequency to return to resting rate from active state, giving rise to a large L/T ratio, or CSI, mimicking the seizure state found by Jeppensen et. al. [5].

If a large CSI is calculated from the watch's heart rate monitor during activity recovery compared to CSI at rest the Watch may be able to use this as a detection method. In the top figure of figure 2.6 there appears to be some relationship between the SUDEPmonitor's pseudo-CSI and the CSI calculated from a pulse transducer but this may not be consistent enough to use the SUDEPmonitor's approximation as an estimator for the true CSI (2.6 bottom). In the figure 2.6, CSI_{20} was calculated instead of the CSI_{30}, CSI_{50}, or CSI_{100} used in Jeppensen's study due to the shortness of the tests. The watch returns a heart rate value around every 5 seconds requiring 150 seconds to get even a single CSI_{30} value, making the trends difficult to interpret during the short tests due to few data points for CSI_{30} and no points for CSI_{50} or CSI_{100}. This is also the cause of the much larger delay for the first Watch CSI value compared to the first pulse transducer CSI value and why the watch has much fewer total points.

While the correlation between the two CSI signals may not be strong enough to treat the watch's CSI values as true CSI values there does appear to be difference's between the active and at rest tests. During the active test the Watch's initial CSI_{20} value is large followed by a steep downward trend. This same trend is found in the other two active tests but not in any of the other six tests. As stated before, the Lorenz plot from active recovery has the same shape as that found for Jeppensen's oncoming seizure giving some hope for the Apple Watch's limited heart rate monitor.



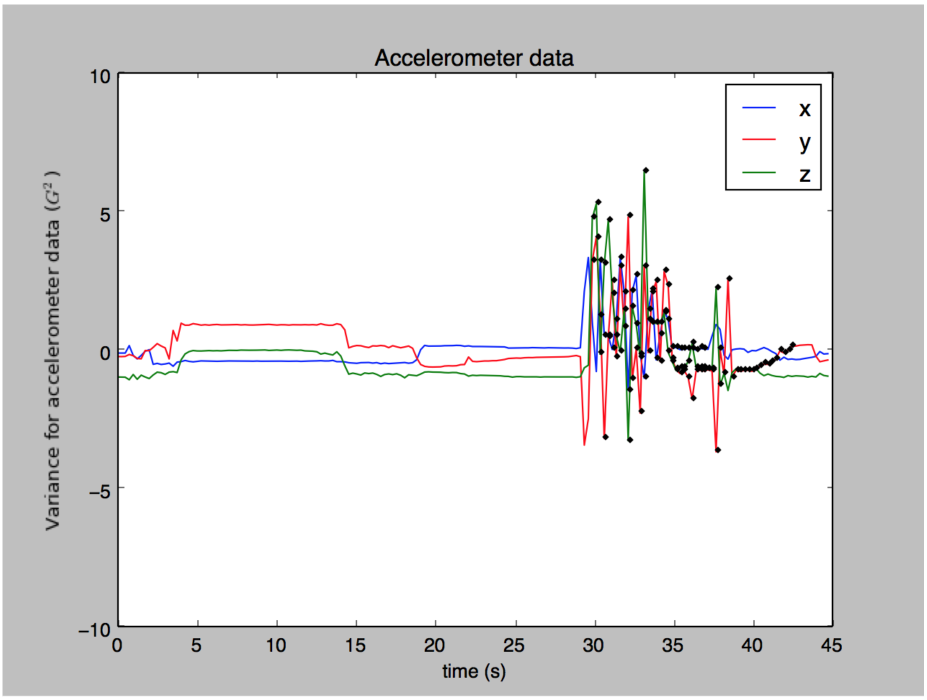


Figure 2.2. Detection of large variance in acceleration indicated by black dots. Acclerometer data of the author's sleep produced using the SUDEPmonitor app (no false alarms) (top). High levels of movement produced by shaking the watch during recorded setting of the alarm (bottom).

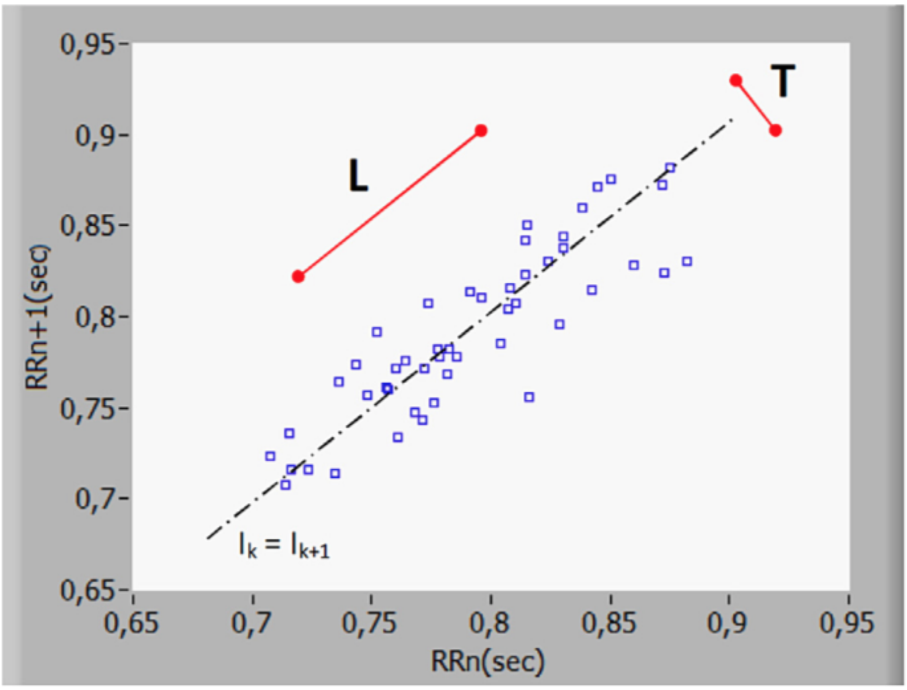
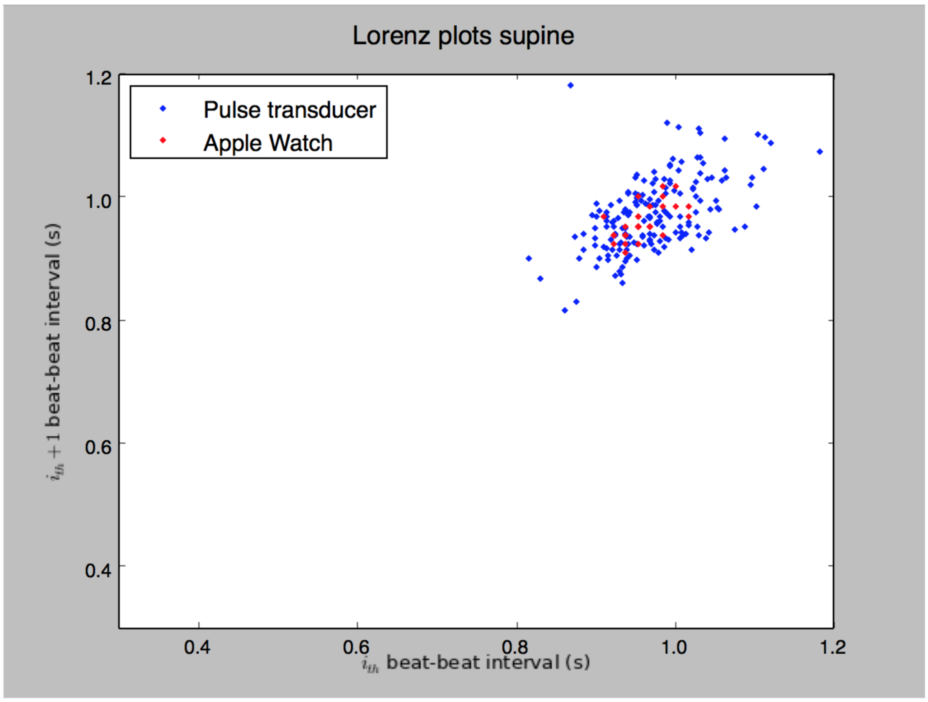


Figure 2.3. Lorenz plot for 50 consecutive heart beats reproduced from Jeppesen et. al. Where L is total variance between the beat-intervals and T is the variance between consecutive beat-intervals [5].



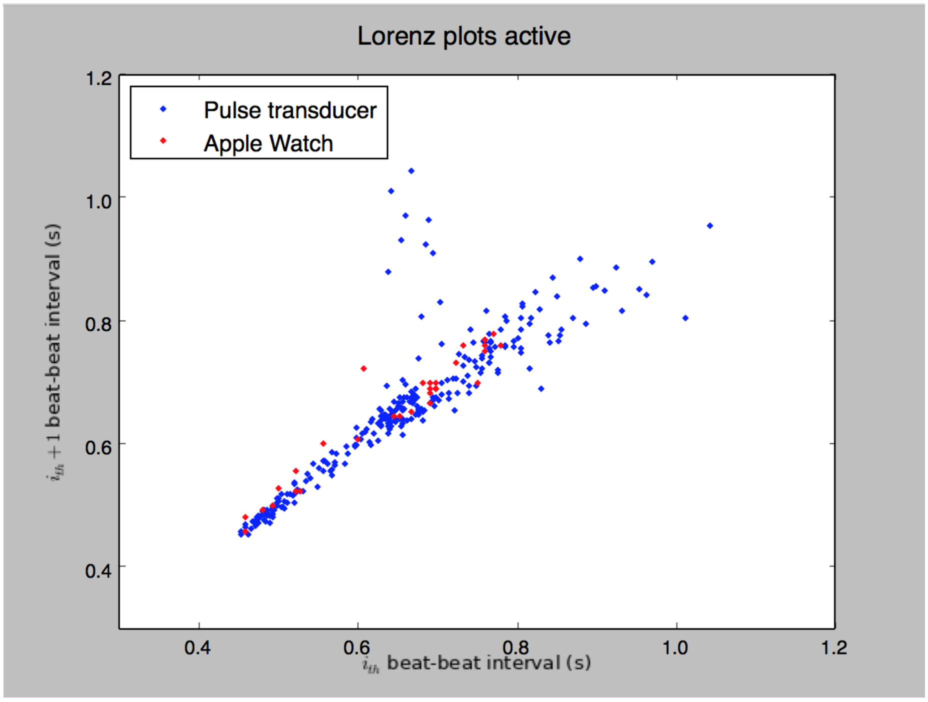


Figure 2.4. The author's Lorenz plots while in a supine position (top) and after light physical activity (bottom).

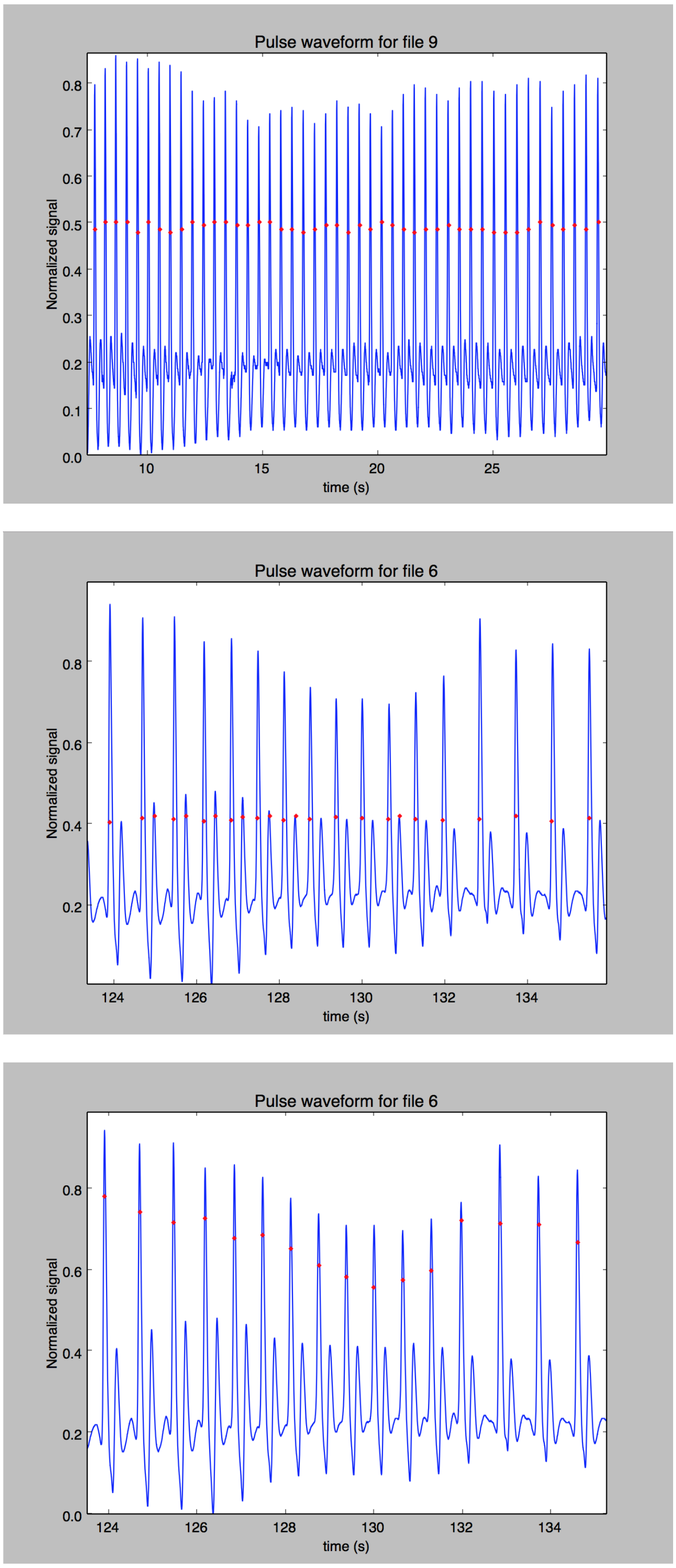


Figure 2.5. Beat detection for pulse transducer signal. Detected beats indicated by red marker. Successful beat detection using a static threshold (top). Unsuccessful beat detection using the same static threshold method on a different signal (middle). Corrected beat detection using the dynamic threshold method (bottom).

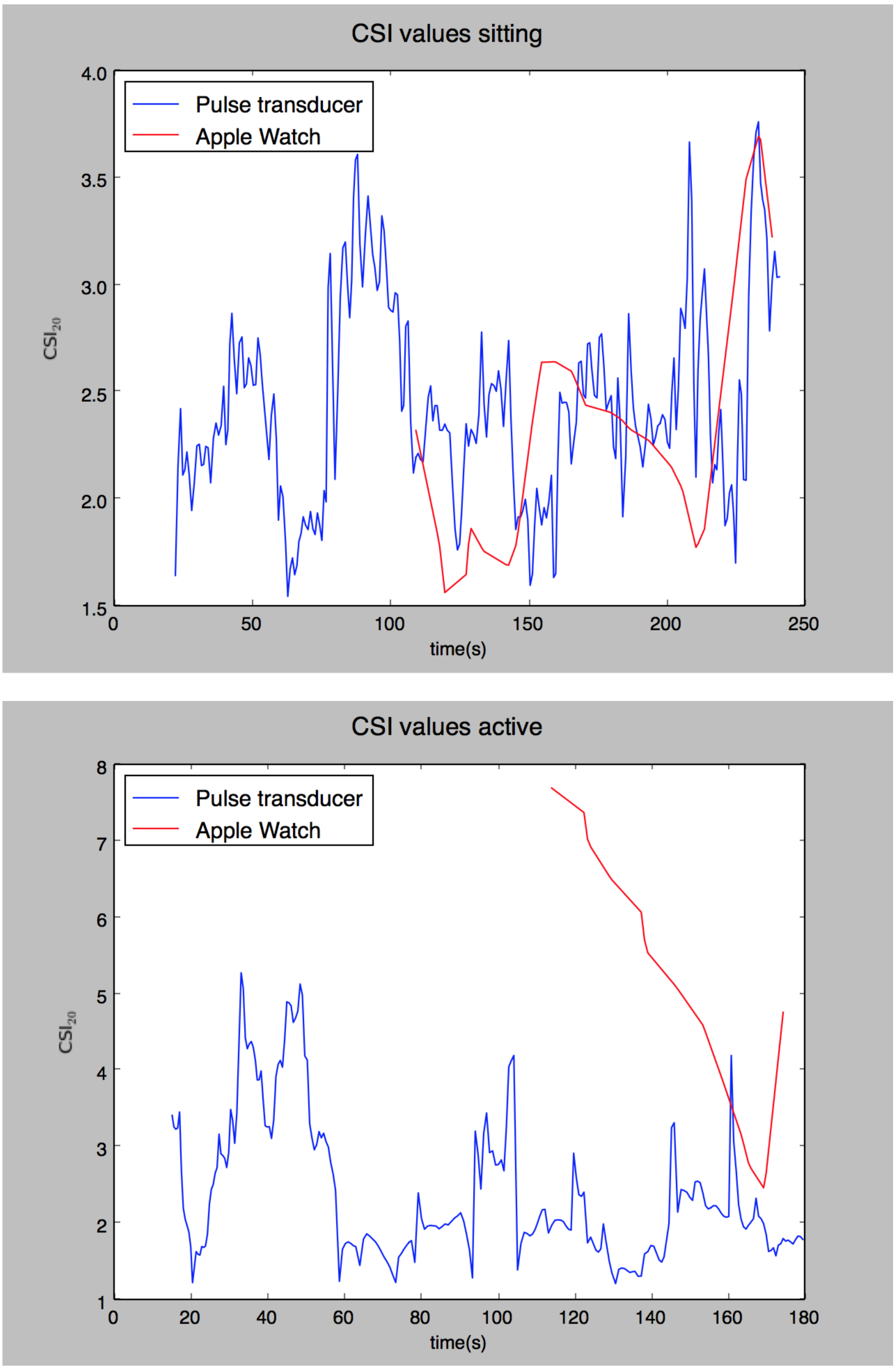


Figure 2.6. Comparison of CSI_{20} calculated using a pulse transducer (blue) and an Apple Watch running the SUDEP monitor (red). Showing the two methods appear to be related (top); however, are not consistently similar (bottom).

CHAPTER 3
APP USE

While the app uses the sensors from the watch, the accompanying iOS app must be downloaded on a linked iPhone before the Watch app can be used in order to give permission to access health data, send data to the server, and add user information. Once permission has been granted via the iOS app (figure 3.2 right) and the watch app has been downloaded the monitor may be used simply by selecting the "Start" button from the Watch app. The app will continue to process the user's biometrics until the "End" button is pressed (figure 3.1). The app is intended to be used specifically during sleep, when 94% of SUDEP cases occur [10].

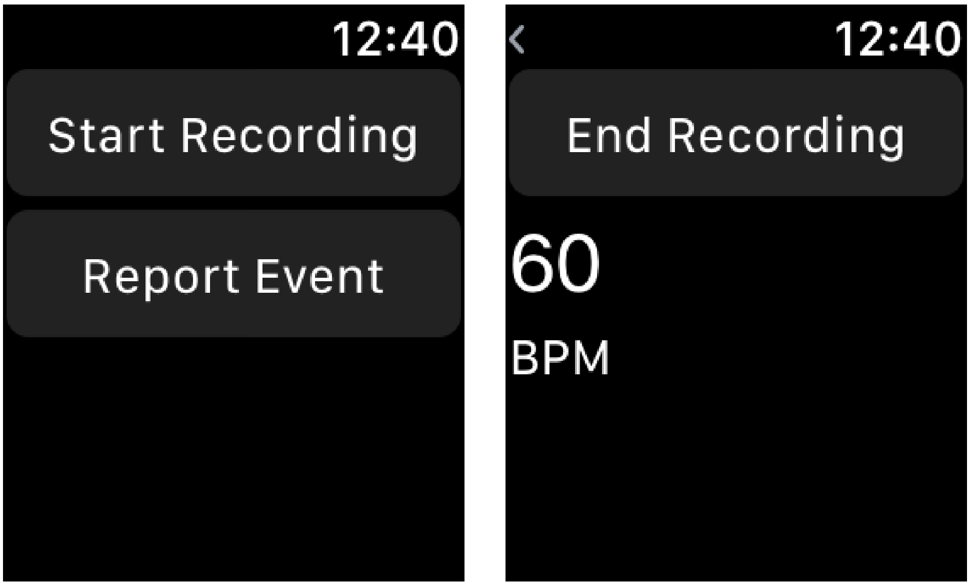


Figure 3.1. Recording session GUI for the Watch App

Upon first use of the iOS app a study information view will be displayed containing a description of the study's intent, how to use the app, and what is expected of the user (figure 3.2). This information will remain available at all time under the study info tab. After reading through the study info, the user will be asked to grant the app permission to read and write to the health store (figure 3.2 right). To use the app, only permission to read heart rate is required. Reading date of birth, height, weight, and sex are only used to auto-fill the profile data which, can be done manually instead. Once the app has been setup the user will receive a unique user ID, which can be found under the profile tab. This ID will be used to group all their data together in the database. Anyone who wants to follow the user sleep activity will need to enter this ID into the companion app.

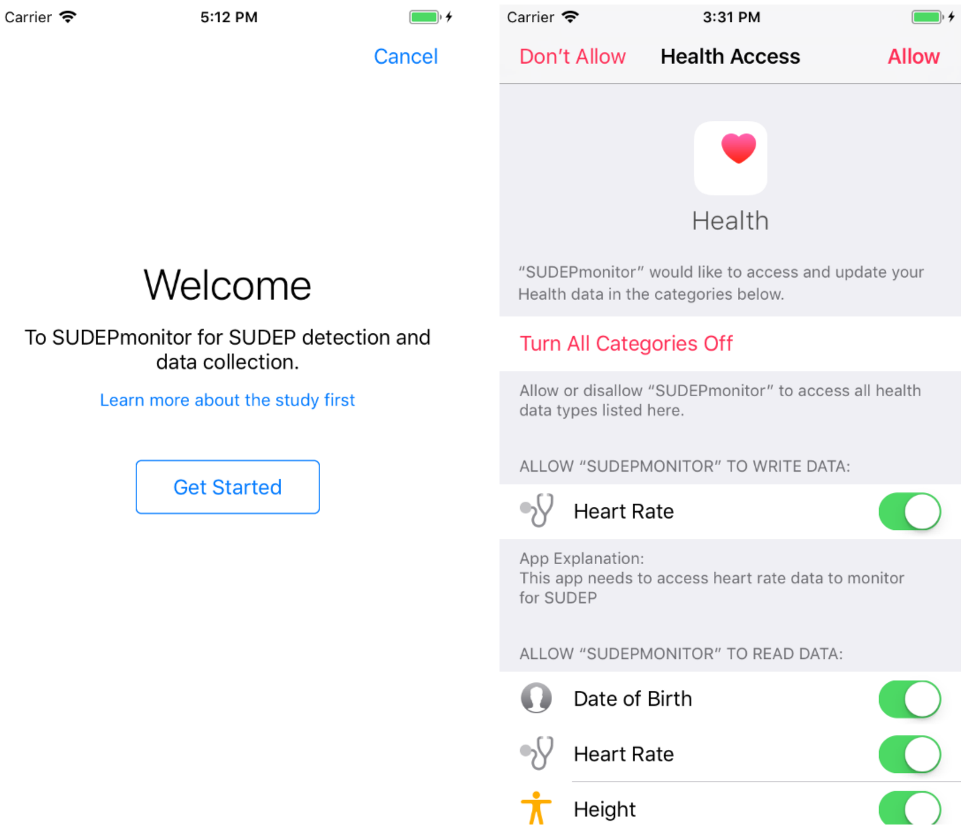


Figure 3.2. The apps welcome screen (left) and health store permissions screen (right).

During a recording session, if the app determines a seizure is occurring it will set off the alarm on the user's watch and linked iPhone and bring up the alarm view (figure 3.4 left) giving the option to silence the alarm or mark it as a false

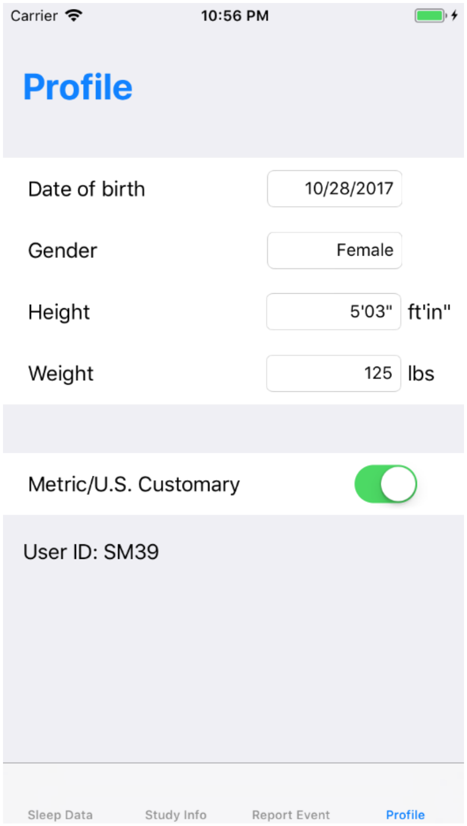


Figure 3.3. The profile view found in the iOS app.

alarm. Simultaneously, the app will send a warning notification to a nearby caregiver, with the help of a companion app, who can intervene. The presence of a close-by individual who is able to provide assistance has been shown to be a successful prevention measure and relatively few cases of SUDEP occur in the presence of a witness [9]. By detecting oncoming SUDEP events the watch should reduce the burden on a caregiver by allowing them to provide assistance without having to persistently watch-over the user. If the user indicates the event to be a false alarm the app will send a second notification to the care giver informing them of the mistake.

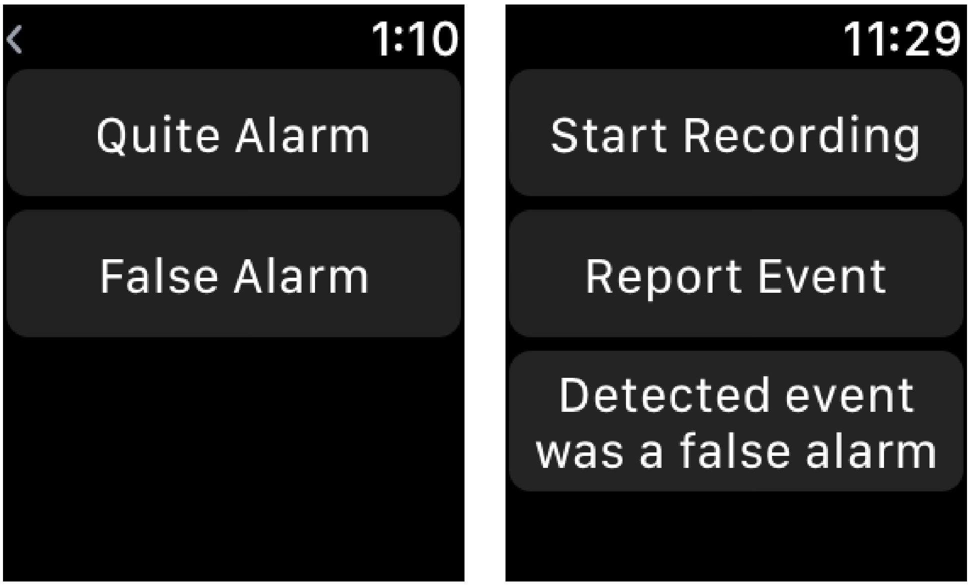


Figure 3.4. The alarm view displayed when an alarm is detected (left) and an additional option to tag an alarm as false after the app detects an alarm (right).

In the case of an event (seizure, SUDEP, or other) both the iOS and Watch apps provide a simple way to report the event. On the Watch the "Report Event" button sends the user to a screen with "Convulsive", "SUDEP", and "Other" options. To send an event tag, first select the option or options which best fit the event then press the "Exit" button. In using the iOS app, events can be reported in a similar manner, with the added ability of including an optional note and selecting the time of the event (the watch app assumes the event has occurred at the time of reporting), and by selecting the "Report Event" tab on the tab bar at the bottom of the screen. Reporting events will help researchers find areas of interest in the data. Therefore, it is important to properly report errors and false alarms. If an event is erroneously reported both the watch and iOS apps have a button to undo the last event report which, will appear on the screen following reporting an error. Similarly, if an alarm occurs the "Detected alarm was a false alarm" button will become available when the recording session is ended allowing for the user to mark a false alarm if they missed the chance during the alarm (figure 3.4 right).

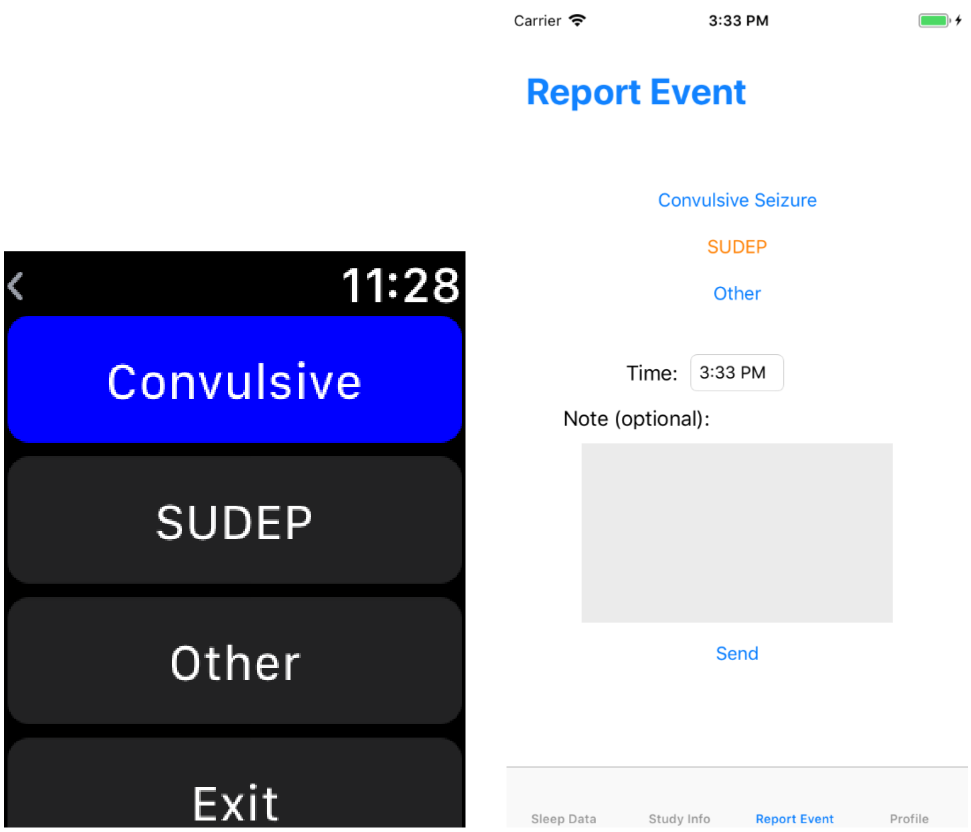


Figure 3.5. Event reporting on the Watch app (left) and iOS app (right).

After a session, a summary of the night's movement and heart rate over the course of the night can be viewed from the "Sleep Data" tab by swiping between the screens (figure 3.6). This provides the average values calculated over 15 minute periods. Heart rate is in beats per minute while the accelerometer data is unit-less as it has been normalized to provide integers rather than float values for clarity. Under the same tab there is a third screen which keeps track of the users events over the past week. The values displayed on the screen are saved locally instead of the entire dataset, to reduce the required on-device storage and prevent the need of recalculating the value each time the app is opened.

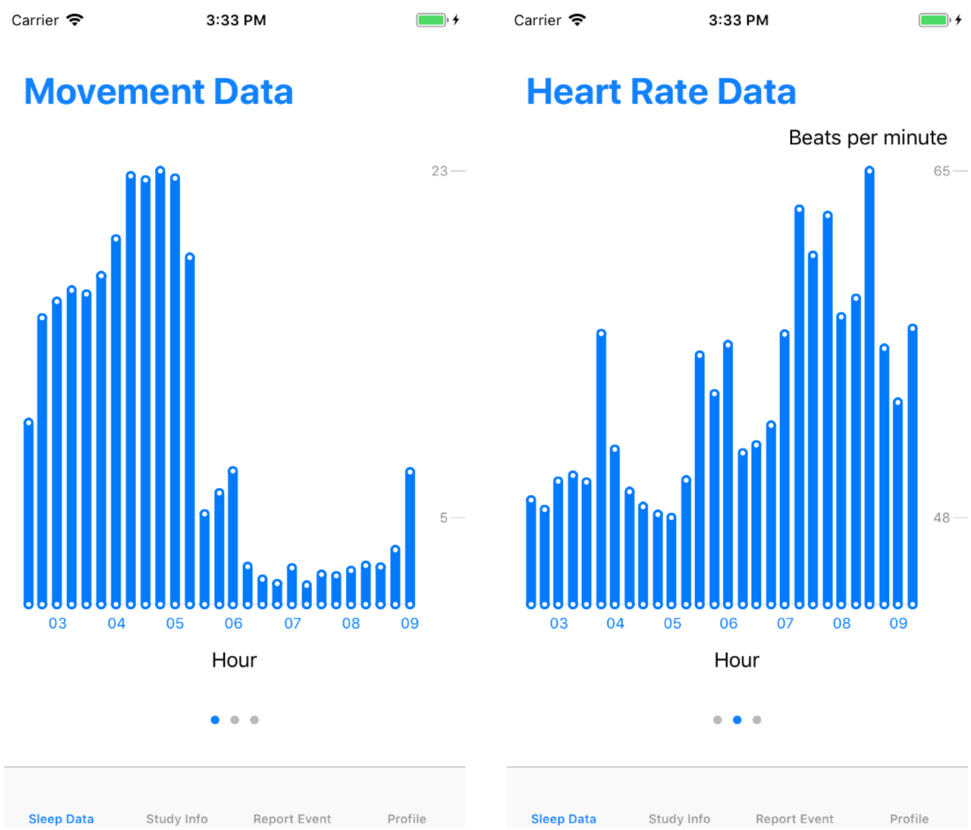


Figure 3.6. Session movement (left) and heart rate (right) display in the iOS app.

3.1 Companion App

In order to receive notifications for a specific user, a caregiver must download the observer app. Within the observer app any number of users can be added to the following list. The caregiver will receive notifications for each of the users in their following list. There are three different notifications the app will send: alarm started, alarm was a false alarm, and alarm ended. The alarm started notification is accompanied by an alarm. Since Apple prevents third-party apps from playing sounds when the phone's sound is off (unlike the native alarm app), to hear the alarm the phone's volume must be turned on.

The observer app displays the following list in the main view (figure 3.7 left). New followers can be added by selecting the "add" cell at the bottom of the following list. Once tapped the add followers view will appear (figure 3.7 right). New followers can then be added with their unique user ID, mentioned above, of the form "SMXY" for integer values X and Y. This can be found in the profile tab of the user's iOS app under the units switch (figure 3.3). If the ID is typed in wrong the app will indicate whether the ID is in the wrong format or does not exist in the database. Followers can also be deleted by selected the edit button on the top right of the main view and tapping the red button which appears next to the to-be-deleted follower or by simply swiping to the left on the cell containing the same follower and touching the delete button that appears.

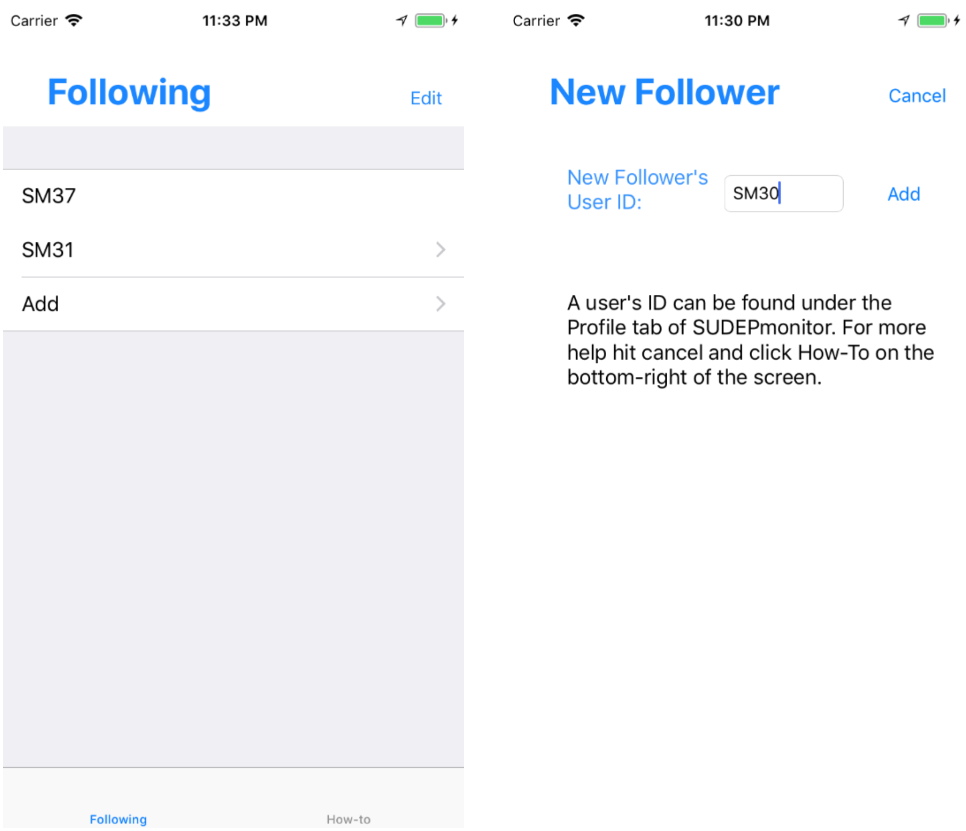


Figure 3.7. Viewing (right) and adding (left) followers within the observer app.

CHAPTER 4
DATABASE

The database is organized into branches for each user, named after their user ID, and a user list branch (figure 4.1). As the name suggests the user list branch provides a list of all the users in the database. This is used when creating new user names, when checking for users, and to provide a list of users for those accessing the database through the python file, sudep.py (see section 5, Accessing the database).

Figure 4.1. Outline of the databases structure.

Each individual user's branch contains their profile information (from the profile tab), sleep sessions, reported events, followers, and the current state of their app (whether there is an ongoing alarm). Each session is given it's own branch named by the date and time of the session (ddmmyyhhmmss) with the time in Coordinated Universal Time (UTC). The rest of the data is added to a meta-data subbranch within each user's main branch (figure 4.2 top). The events and dates branches within the meta-data branch contain the events and the dates of all the user's recorded session, used for similar purposes as the user list branch. Items in these two lists are named in the same manner as the individual sessions, allowing for events to be matched up with the sessions they occurred during.

When the app detects a seizure it automatically sends an event to the database with the current time and indicates the event was detected by the app, rather than reported by the user. If this event was determined to be a false alarm a note is appended to the event acknowledging the error (figure 4.2 bottom). The user should report the same event to confirm the event and add information about the type of event.

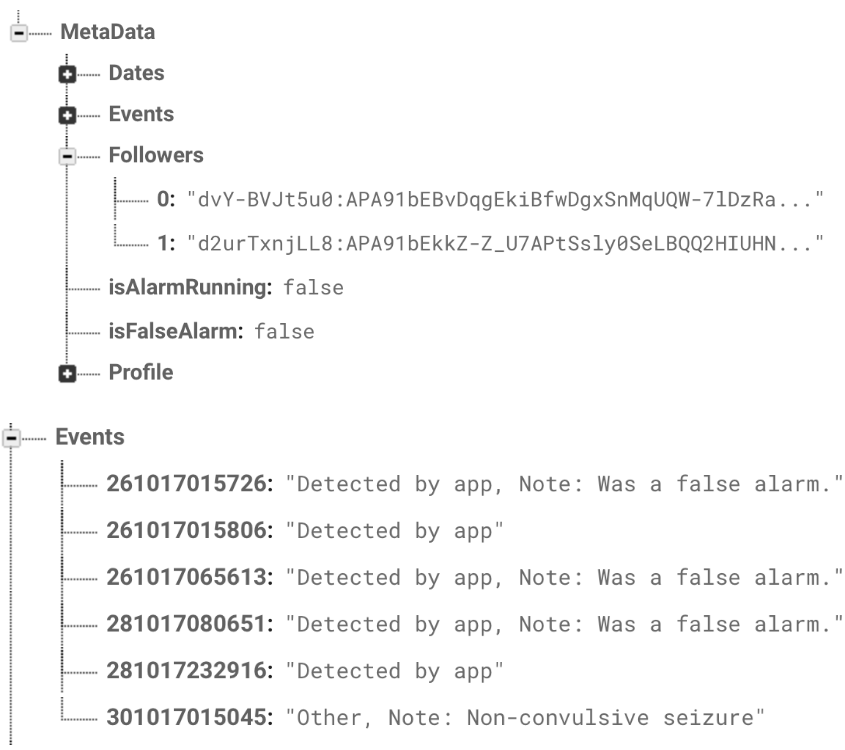


Figure 4.2. The database's hierarchy for storing meta-data (top) and events (bottom).

As well as sending an event tag to the database, the app also flips the "isAlarm-Running" flag to true when an event is detected (figure 4.2 top). By default this flag and the "isFalseAlarm" flag are set to false. When the database detects a change in the value of this flag it checks whether it changed from false to true or from true to false. When the former happens the database sends out a push notification to all devices in the user's followers branch (see subsection 3.1, companion app). This notification indicates an event has been detected. For the second case, the database ignores the switch. This flip is used to bring the database back to its normal state and allow for a new notification in the case of another event. If the event is determined to be a false alarm, the "isFalseAlarm" boolean is flipped leading to a similar check by the database which then sends out a notification explaining the previous event was a false alarm.

CHAPTER 5
ACCESSING THE DATABASE

Anyone is able to read the data from the database. This can be done simply with the help of sudep.py (a python module) containing several functions to import data from the database for use within the python environment. The module sudep.py is dependent on the requests, matplotlib, and numpy libraries so those packages must be installed before the module can be used. The simplest method to install these libraries is using python package manager pip (see appendix A for instructions). The sudep.py module contains functions for obtaining all the user IDs, displaying users' profiles, showing all of a user's sessions and events, and plotting user sessions, as well as storing heart rate and accelerometer data.

The module provides the list: user_list; functions: get_user, save, load, plot, get_events_for, get_profile_for, get_session_dates_for, and get_session_for along with classes: User, Session, and Profile. Although it contains all these objects, user_list, get_user, and plot should provide the full functionality of the module alone. The three classes have been added to provide details on what they store since instances of the User class then are returned from the get_user function, which in turn, contains instances of the other two classes, Session and Profile. While the rest of the functions were added to allow for additional flexibility if needed.

Like the database branch it gets its values from user_list, a list of the users contained in the database. These can be used as an argument when using the function get_user which accepts the name of a user and returns an instance of the class User. More so than that, get_user checks the local directory for a folder named "Users". If the directory does not exist the function creates it, downloads the user's data, and saves it to the newly created "Users" folder. If the folder does exist, it checks for a file for the entered user name, creating it by downloading and saving the file as previously stated if the file does not exist. If the file, does exist the function loads the file locally rather than pulling the data from the database. Pulls from the database are much more time consuming than loading the same data from a local drive. For this reason it is suggested get_user is used rather than initializing an instance of the User class directly. To replace the current file in the case new data has been added to the user's branch, the optional parameter "reload" can be set to true and get_user will write over the file if it already exists with the command below.

```
>>> get_user(user_list[0],reload=True)
```

The User class provides all the data from a user's branch organized in a similar manner to the database. This class contains five variables: self.user_name, self.profile, self.events, self.sessions, and self.dates. To view this data, print the class instance (figure 5.1). This provides all the dates of the sessions and events printed out in an easy to read format. The class's self.dates is a list of dates arranged in the same order as displayed under "Sessions" when printing the user. The variable self.sessions is a dictionary of Session instances with the dates as keys. As such for the instance of the User (named user below) in figure 5.1 the following command will return the first session for the 10th of October.

```
>>> user.sessions[dates[0]]
```

The Session class provides the monitor data from the session along with the duration of the session and the sampling frequency for the accelerometer. This data is summarized by printing the session (figure 5.2). Accelerometer and heart rate data are contained in separate dictionaries. The accelerometer's data is in a dictionary with keys: 'x','y', and 'z' corresponding to the three spacial axes. The values associated

```
User Name: SM36
Profile:
        Date of Birth: 10.23.2017
        Gender: Male
        Height: 1.78 m
        Weight: 78 kg

Events (onset (dd.MM.yy hh:mm:ss) || event type):
        26.10.17 06:56:13 || Detected by app, Note: Was a false alarm.
        26.10.17 01:58:06 || Detected by app
        28.10.17 08:06:51 || Detected by app, Note: Was a false alarm.
        28.10.17 23:29:16 || Detected by app
        26.10.17 01:57:26 || Detected by app, Note: Was a false alarm.

Sessions (date (dd.MM.yy hh:mm:ss) || duration):
        25.10.17 02:18:46 || 00:00:18
        25.10.17 02:58:38 || 00:00:43
        25.10.17 03:00:57 || 00:00:44
        25.10.17 03:09:24 || 00:00:36
        25.10.17 07:28:05 || 07:02:42
        26.10.17 01:57:19 || 00:00:29
        26.10.17 01:57:59 || 00:00:15
        28.10.17 08:06:43 || 03:26:38
        28.10.17 23:29:05 || 00:00:29
        29.10.17 09:40:37 || 04:49:56
```

Figure 5.1. A print out of the author's data from an instance of User.

with each key are numpy arrays of all the samples for the given axis. The heart rate dictionary has keys 'times' and 'heart_rate'. Again the dictionary values are stored as numpy arrays. The times array is needed for the heart rate data since the watch does not return the samples in regular intervals.

```
Accelerometer sampling frequency: 4 samples/s
Elapsed time of session: 17396.72 s
Average heart rate: 56.24 bpm
```

Figure 5.2. The author's most recent session (October 29th, 2017).

To get a quick view of a given session the plot function will plot the accelerometer and heart rate data over the course of the session (figure 5.3). This function uses the matplotlib module requiring the use of the plt.show command to display the plot.

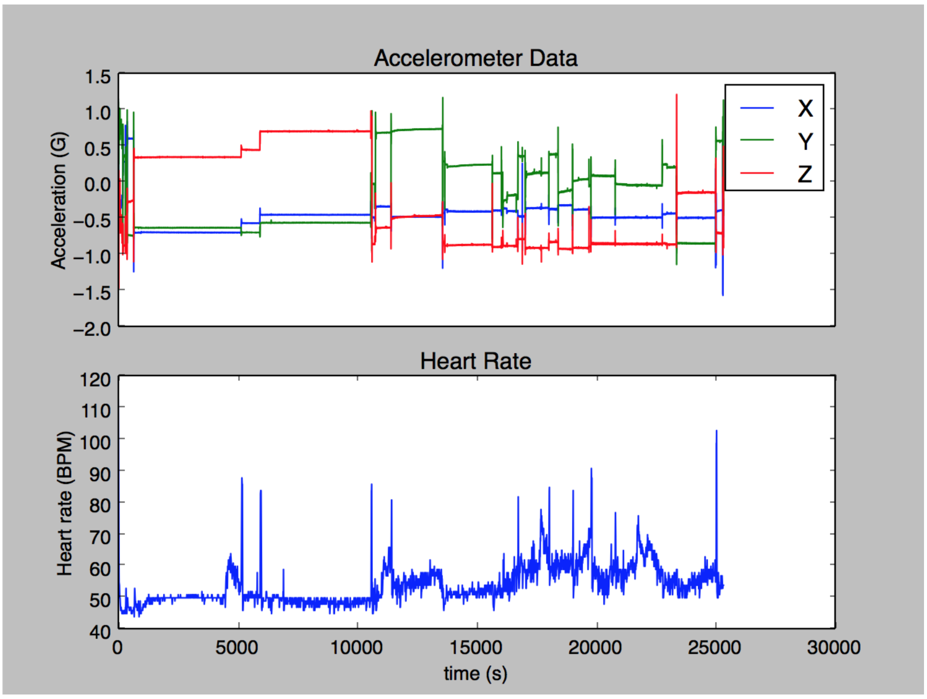


Figure 5.3. The figure produced by sudep.py's plot function for the author's most recent session.

CHAPTER 6
DISCUSSION

At the moment, the app attempts to detect SUDEP indirectly by detecting seizure. While studies have shown most observed cases of SUDEP occur after a seizure, there are still cases which do not, limiting the detection power of a seizure dependent SUDEP monitor [9]. There may be better predictors which have yet to be identified. The information gained from this app along with future outside research may shine light on patterns related to SUDEP onset to provide better predictors. However, while the risk of SUDEP is high relative to other causes of death the prevalence is still measured in cases per 100,000 person-years ($\sim$ 0.09-2.65 cases per 100,000 person-years) [3]. Meaning, even with 1,000 users, there may be one or less cases monitored over a period of 100 years. In order to provide insight on SUDEP a huge user base will be required. Furthermore, not all night-time seizures are potential SUDEP cases. It will be difficult to distinguish between a prevented SUDEP case and a non-deadly seizure from the information saved in the database.

Even with these challenges, SUDEPmonitor may still prove to be helpful in its current state, searching specifically for seizures. Data collected from the app along with future studies could allow the app to detect seizures and ultimately SUDEP better. Or more ideally, may provide a direct method of SUDEP detection.

Due to time and resource limitations, the app has yet to be tested on real seizures. Tests dealing with real seizure will be needed to better tune the detection methods and determine a relationship between the Apple Watch's heart rate monitor and seizure onset.

At the moment the accelerometer based detection method is based on a static threshold. Differences between individuals may cause this method to be useless in it's current state. By obtaining the sample variance for the difference of accelerometer data in many individuals (potentially through the app) it can be determined whether the variance is consistent throughout the population under normal sleeping conditions. If so, and sample variance during convulsive seizures are generally large compared to that of normal sleep, the simple threshold method may be an adequate convulsive seizure detector. However, if this is not the case, a training period may be required to produce a personalized threshold or another method altogether could be needed. As shown above, the difference in accelerometer data is approximately distributed as a zero-mean Laplace random variable. Using this, along with an estimate of variance under control and seizure conditions, it may be possible to devise a better test statistic using the Neyman-Pearson lemma [6].

As there are few studies specifically relying on the Apple Watch's heart rate monitor, additional studies will be required to properly utilize the heart rate method in the app. The heart rate method was not been applied to the SUDEPmonitor at the time of writing this paper since the slow sampling rate prevented it from providing the needed details to work with current heart rate based seizure detection methods. This issue may be bypassed in the future if additional access to the watch's raw photoplethysomogram (PPG) signal (which is used by the watch to calculate the heart rate) is given. In the event this does not happen the current heart rate signal from the watch may still be useful.

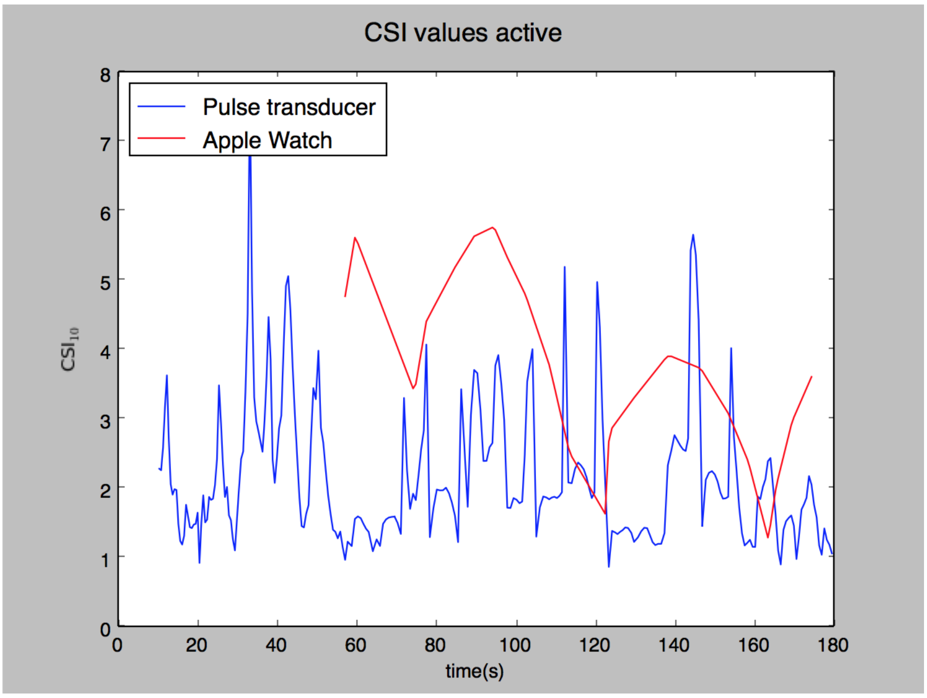
As mentioned in section 2 (detection methods), there is some indication of a trend unique to the active tests. During these tests the CSI_{20} values started off large and steadily fell back to normal values. More so, recovering from activity produced a similar Lorenz plot as those produced by Jeppensen [5]. It's possible seizures cause the opposite affect of the activty recovery tests (increase the heart rate over time rather than bring it down). In either case since the CSI values are functions of the Lorenz plot, the exact phenomenon for which the plot was produced should not affect the CSI obtained from it.

Before a detection method based on this potential phenomenon can be designed, longer tests are needed. The three to five minute tests used in this paper were not long enough to produce even CSI_{30} values. Viewing the active plots at smaller values of n, for CSI_n, the trend disappears (figure 6.1), suggesting larger values of n may work better. With longer tests there will be more flexibility to find the best n value. In addition, more tests are needed to imply the trend is more than an artifact of a small sample size.

With all this, it is clear more preliminary studies are needed before SUDEP-monitor can be of use. In addition to modifying the detection methods, there are also some issues with the data transferring. Currently. data is sent to the database through the phone. (The watch is not able to communicate directly with the database.) Because of this the phone has to be connected, through Bluetooth, with the watch to save data to the database. When the iOS app is in the background any large data files sent from the watch are handled on a background thread which is designed to optimize battery life and minimize processor burden resulting in transfers that may start tens of minutes after the initial request from the watch. Since the watch has limited storage and memory, it may not hold on to the session data once the SUDEP-monitor watch app has gone to the background potentially, causing lose of data. A simple solution is to open the iOS app after ending a session. But an ideal solution would not require any extra action by the user. A better fix would be to have direct communication between the watch and database. Not only would this minimize lost session data, it should also reduce lag between an event happening and the database reacting to the event to send out notifications. (Although this lag is much shorter than the session files since smaller files can be sent to the phone more directly.)

Lastly, the push notifications need to be tested. Without a full Apple developer account it is not possible to add the needed APN (Apple Push Notification) certificates to the database. Without the certificates, Apple devices will not accept a push notifications. Log files from the database show that the database is reacting to the changes in the two session flags (”isAlarmRunning” and ”isFalseAlarm” see section 2, detection methods) and running the correct programs to send out the notification. So once the certificates are available finalizing the companion app should require minimal work. Also, due to the notifications being dealt with solely by the server, observer apps for other operating systems can be designed without having to alter the main app in any way, removing restrictions on phone operating system for a user's followers.

While the SUDEPmonitor, in its current form, is ready for collecting data, its detection may be limited. It should be able to detect some indicators of SUDEP but some additional adjustments could vastly improve the detection potential of the app. After these last fixes, along with adding the required app icons, the SUDEPmonitor should be ready to be sent to the app store.



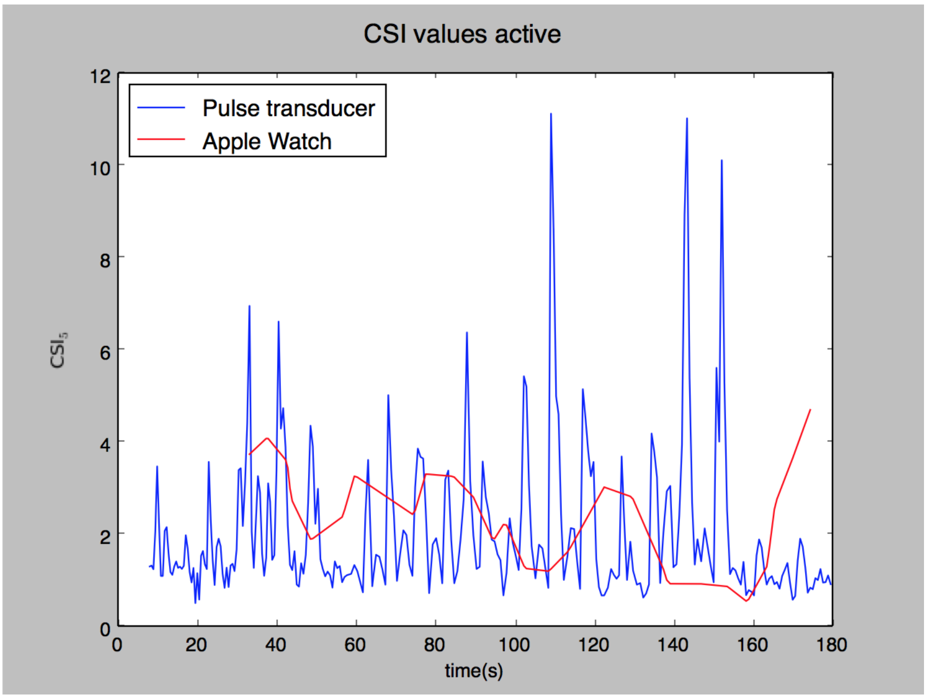


Figure 6.1. $\mathrm{CSI_n}$ values for the active test from figure 2.6 calculated with n = 10 (top) and n = 5 (bottom) points.

APPENDIX A

INSTALLING PYTHON MODULES

First pip needs to be installed if it is not already (it may be installed by default). Download get-pip.py from https://pip.pypa.io/en/stable/installing/. Follow the instructions on the website to complete the pip install for your OS. Now a single line in terminal can install the libraries:

```
$ pip install library_name
```

Note the name needs to be in quotation marks. If using a unix based system you may need to include sudo to the front of the command and type in the system password when prompted.

BIBLIOGRAPHY

[1] S. Beniczky, T. Polster, T. W. Kjaer, and H. Hjalgrim, "Detection of generalized tonic–clonic seizures by a wireless wrist accelerometer: a prospective, multicenter study," *Epilepsia*, vol. 54, no. 4, 2013.

[2] A. K. Golińska, "Poincaré plots in analysis of selected biomedical signals," *Studies in logic, grammar and rhetoric*, vol. 35, no. 1, pp. 117–127, 2013.

[3] A. G. Holst, B. G. Winkel, B. Risgaard, J. B. Nielsen, P. V. Rasmussen, S. Haunsø, A. Sabers, P. Uldall, and J. Tfelt-Hansen, "Epilepsy and risk of death and sudden unexpected death in the young: a nationwide study," *Epilepsia*, vol. 54, no. 9, pp. 1613–1620, 2013.

[4] J. R. Hughes, "A review of sudden unexpected death in epilepsy: prediction of patients at risk," *Epilepsy & Behavior*, vol. 14, no. 2, pp. 280–287, 2009.

[5] J. Jeppesen, S. Beniczky, P. Johansen, P. Sidenius, and A. Fuglsang-Frederiksen, "Using lorenz plot and cardiac sympathetic index of heart rate variability for detecting seizures for patients with epilepsy," pp. 4563–4566, 2014.

[6] S. M. Kay, *Fundamentals of statistical signal processing.* Prentice Hall PTR, 1998, vol. 2.

[7] S. D. Lhatoo, M. Nei, M. Raghavan, M. Sperling, B. Zonjy, N. Lacuey, and O. Devinsky, "Nonseizure sudep: Sudden unexpected death in epilepsy without preceding epileptic seizures," *Epilepsia*, vol. 57, no. 7, pp. 1161–1168, 2016.

[8] J. Piskorski and P. Guzik, "Geometry of the poincaré plot of rr intervals and its asymmetry in healthy adults," *Physiological measurement*, vol. 28, no. 3, p. 287, 2007.

[9] S. Shorvon and T. Tomson, "Sudden unexpected death in epilepsy," *The Lancet*, vol. 378, no. 9808, pp. 2028–2038, 2011.

[10] E. Tu, R. D. Bagnall, J. Duflou, and C. Semsarian, "Post-mortem review and genetic analysis of sudden unexpected death in epilepsy (sudep) cases," *Brain Pathology*, vol. 21, no. 2, pp. 201–208, 2011.