Ups and Downs

Final Report

ECE1778: Creative Applications of Mobile Devices

December 14, 2016

Submitted to

Jonathan Rose

Submitted by

Fahad Almusharraf
Daniyal Liaqat
Naba Siddiqui

Word Count: 2492

Contents

1 Introduction

Mobile and Wearable devices contain numerous sensors that can provide objective data about users and their environment. Since data is collected passively, without user interaction, continuous and long term collection of this data is possible with minimal overhead to the user. Additionally, monitoring of this sensor data can provide insights into user's lives over time and allow detecting changes in the user's behaviours. These changes in user behaviour are particularly interesting for health monitoring purposes because they can indicate changes in the patient's health. For example, in patients with chronic lung disease, an increase in coughing frequency and decrease in physical activity may indicate a worsening of the disease. In patients with mood disorders such as Bipolar Disorder (BD) or Seasonal Affective Disorder (SAD), a decrease in sleep and interaction with other people may indicate a depressive episode. These types of objective measurements may provide a stronger basis for diagnosis than subjective self-reporting, which is often unreliable.

However, mobile and medical research is currently not at a point where we can accurately and confidently make diagnosis's from sensor data. Researchers first need to conduct studies to find and validate correlations between sensor data and health. These studies pose their own challenges. Firstly, it is not trivial to implement continuous sensing applications on Android. Studies can run for months so the application needs to be well tested, reliable and resilient to failures. Secondly, processing sensor data is also tricky. Sensor data is generally noisy and in some cases could be completely incorrect. So to make this data useful, it must be preprocessed. Finally, it is very difficult to derive a measure of lung or mental health from raw sensor data. Instead, sensor data is generally first used to detect activities (ex. walking, speaking, coughing, etc.) that have a more clear relation to the final measure.

To address these challenges, the goal of our project is to develop a configurable and extensible sensor data collection and processing platform based on Android and Android Wear that will allow medical researchers to conduct studies that make use of continuous sensing.

To focus our work, we chose BD as a motivating example. In the application we built, the sensors we used and activity detection algorithms we implemented are those that are likely to be used by researchers conducting a continuous sensing study on bipolar disorder. The application is named "Ups and Downs", referencing the manic (up) and depressive (down) states associated with BD.

2 Statement of Functionality

Ups and Downs consists of software running on three devices, (1) a smartwatch, (2) phone and (3) server.

2.1 Smartwatch Application

The main functionality of the smartwatch application is to collect sensor data. Since fully continuous recording would drain battery life very quickly, making the application inconvenient to users and resulting in incomplete data, we decided to use a duty cycling scheme

(a) App Launcher (b) Ups and Downs UI 1/2 (c) Ups and Downs UI 2/2

Figure 1: User Interface of the Smartwatch

where we record for 2 minutes and sleep for 8 minutes. This provides a battery life of over 14 hours and sensor data from throughout the users day.

While recording, we collect data from the step counter, heart rate sensor, accelerometer, gyroscope and microphone. We chose these sensors because (1) they will likely be useful in monitoring BD and (2) are some of the most commonly used sensors. However, adding other built-in sensors to the application is trivial.

Audio data collected on the smartwatch is saved as an MP3 file to make file storage and transfer manageable. Data from other sensors is stored in an JSON file. At night, the MP3 and JSON files collected throughout the day are transferred to the phone over Bluetooth.

The User Interface (UI) of the smartwatch is shown in Figure 1. The Android Wear launcher showing the Ups and Downs application is shown in Figure 1a. Figures 1b and 1c show the UI of the Ups and Downs smartwatch app. The app is designed to not require any interaction from the user, however, users may pause recording for 10 minutes by pressing the "Stop Recording" button. The "Delete" and "Send" buttons and other information displayed are mostly there for debugging purposes.

The only issue we had was with the step counter always reporting zero steps. We think this may have been as a result of an Android Wear update because it was working at some point and there was no change in our code.

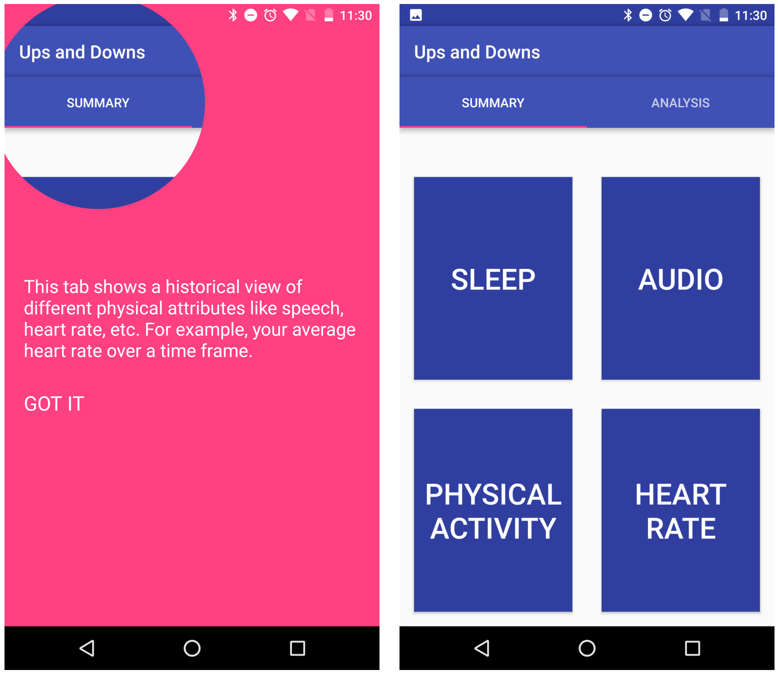
2.2 Phone Application

The Ups and Downs phone application has two roles, (1) act as in intermediary between the watch and server and (2) display results to the user.

As an intermediary, the phone application receives data from the watch. Currently data is manually transferred from the phone to the server.

The phone app shows a summary and analysis of data after its been processed by the server. Upon launching the app, the user is shown the screen displayed in Figure 3. Clicking one of the buttons shown in Figure 2a, leads to the summary screen shown in Figure 3.

The summary screen graphs activities over time and allows users to view all data, data



(a) Hint shown on first launch (b) Main Screen

Figure 2: Main screen of the phone app

for the last week and last day. Blue dots represent anomalies (values significantly higher or lower than the users baseline).

Another useful visualization is available to users in the analysis tab (Figure 4). It shows anomalies for each activity and whether the anomaly was due to a below or above baseline measurement. This screen is helpful is visualising co-occurring anomalies. For example, in Figure 4a there is a period where sleep, speech and physical activity are all above the baseline.

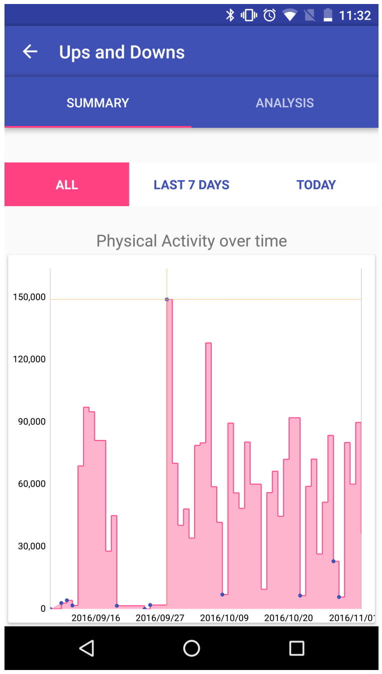
The phone application worked as expected. There are some minor issues with the graphing library we're using (ex. unshaded area under curve in Figure 3b). This is a bug in the graphing library and could be fixed either by submitting a patch or using an alternative library.

2.3 Server Application

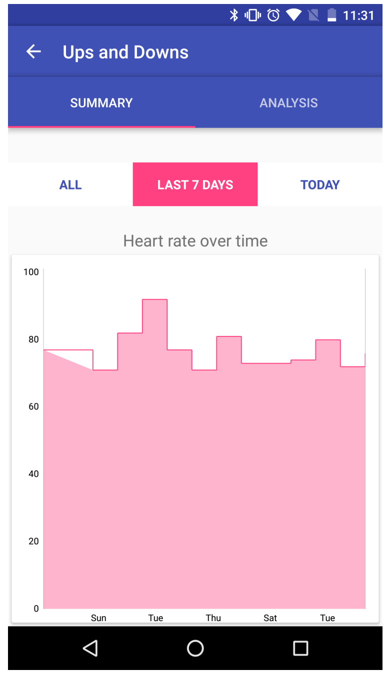
From a user's perspective, the functionality of the server is to process sensor data and generate data that can be graphed for the user. Details about how this is achieved are given in Section 3.1.

3 Overall Design

The design of our application can be explained by looking at how data flows through the system and at how we detect activities.

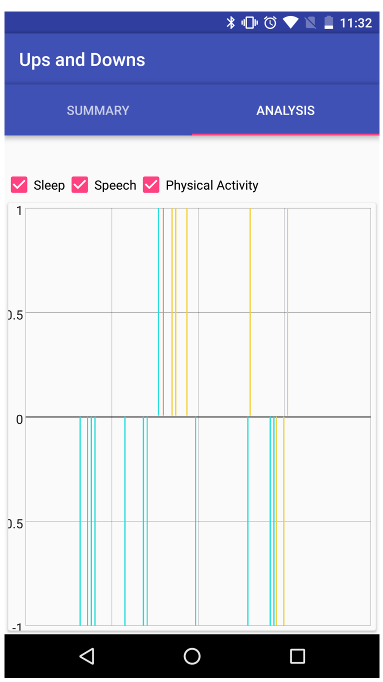


(a) Summary of all physical activity

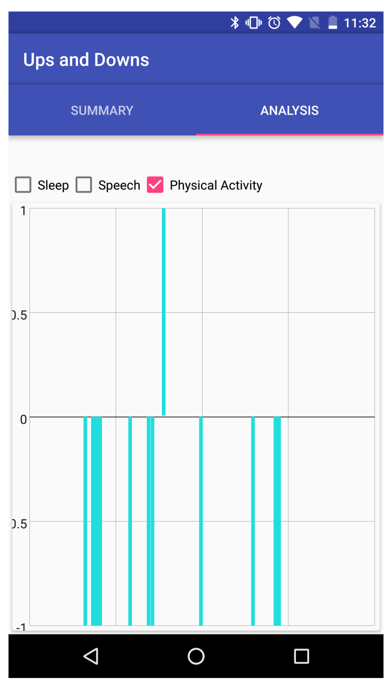


(b) Summary of last 7 days of heart rate

Figure 3: Summary screen in phone app



(a) Analysis of sleep, speech and physical activity



(b) Analysis of physical activity

Figure 4: Analysis screen in phone app

Figure 5: Ups and Downs dataflow

3.1 Data Flow

The watch and phone applications are built for Android and the data analysis framework is built with Python. The data flow between these components is shown in Figure 5.

A block diagram of our application is shown in Figure 6 and each component is described below.

3.1.1 Smartwatch

1. Collect sensor data: Collect data from sensors. A scheduler duty cycles recording and a custom file system manages files.

2. Send over Bluetooth: Send files to phone.

3.1.2 Smartphone

1. Receive over Bluetooth: Receive files from watch.

2. Store sensor data in files: Store sensor files in a custom file system.

3. Receive analysis files: Receive result files from the server.

4. Display sensor data summary: Parse result files and show graphs of of users activity.

5. Display sensor data anomalies: Show a visualization of anomalies.

3.1.3 Server

1. Pull files from phone: Receive files from the phone.

2. Pre-process and clean data: Remove data that is marked unreliable by Android and other cleaning schemes (ex. reject heart rate readings below 30 BPM or over 200 BPM).

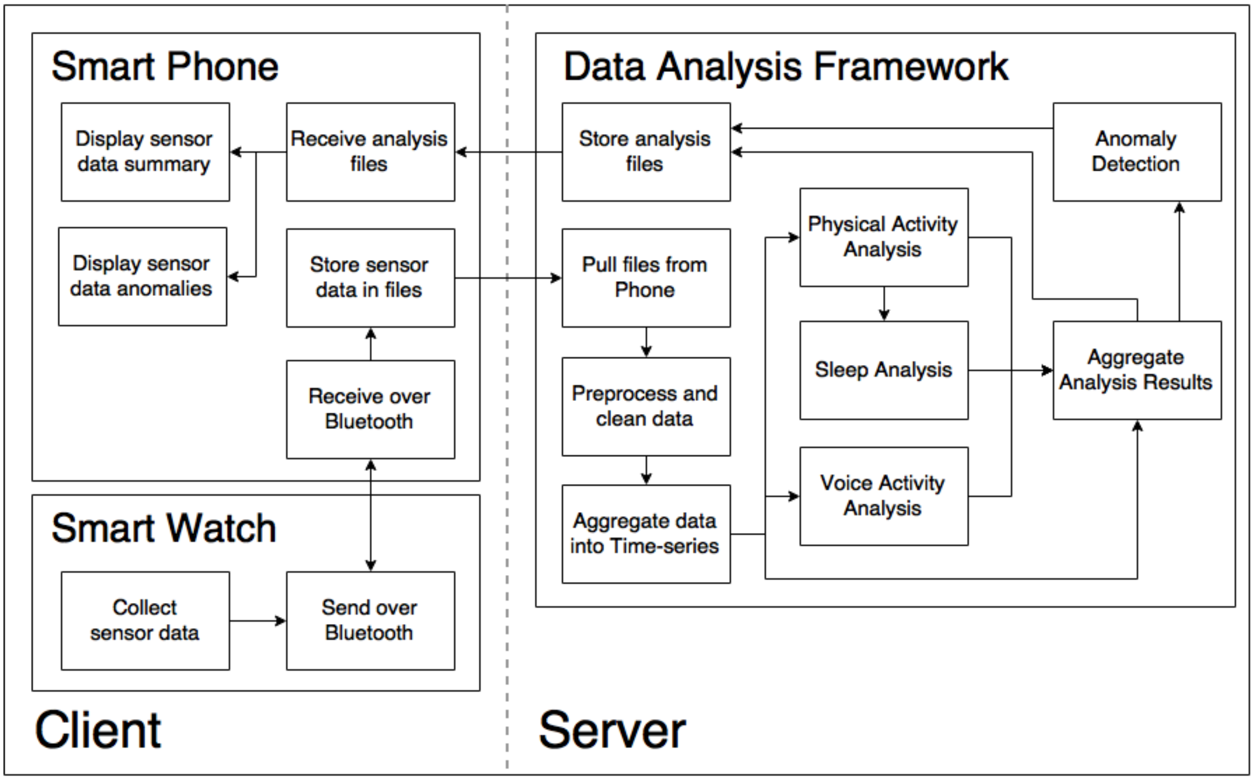


Figure 6: Block Diagram

3. Aggregate data into time-series: Use start and end timestamps along with sampling frequency to interpolate timestamp of each individual sample. This time series allows for resampling of data to look at different sized windows (ex. 1 minute windows for physical activity or 2 hour windows for sleep activity).

4. Physical Activity analysis: Analyze accelerometer data to detect significant motion as an estimate of physical activity.

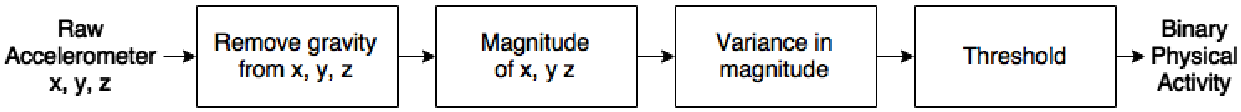
5. Sleep analysis: Use result of physical activity analysis to estimate time and duration of sleep.

6. Voice Activity Analysis: Determine how much speech there is in each audio file and how much of that speech is from the user.

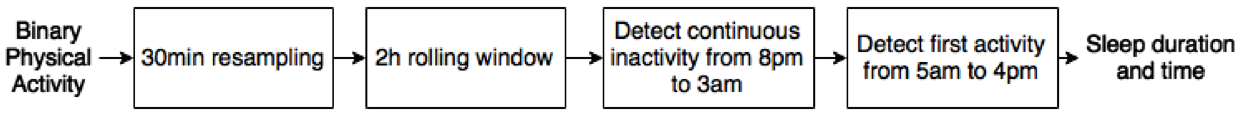
7. Aggregate analysis results: Aggregate the results of the activity detection into a time series for each activity.

8. Anomaly Detection: Establish a baseline and look for significant deviations from the baseline.

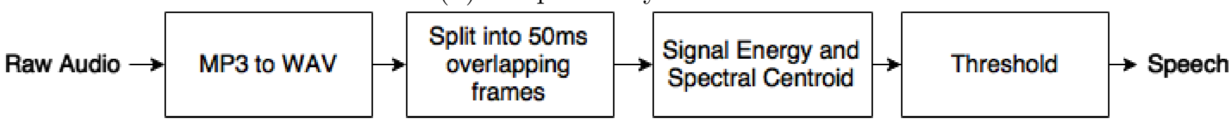
9. Store analysis files: Generate a result file that can be interpreted by the phone application



(a) Physical activity estimation



(b) Sleep activity estimation



(c) Voice activity estimation

Figure 7: Activity estimation

3.2 Activity Detection

These are the activities we implemented for Ups and Downs because they could be useful for detecting symptoms of BD. However, with the time-series representation of sensor data provided by our processing framework, it would not be too difficult to add other activities.

3.2.1 Physical Activity

The process to estimate physical activity is illustrated in Figure 7a. Physical activity estimation is just the magnitude of accelerations. This magnitude is calculated after removing the effect of gravity from the accelerometer raw sensor data. Had the step counter worked, it would have been useful as a physical activity estimation. Binary physical activity is generated to perform sleep analysis.

3.2.2 Sleep Activity

The sleep analysis component (Figure 7b) checks for inactivity in times where sleeping might occur. Sleep start time is assumed to be between 8 PM to 3 AM. The 30 minute resampling is done using a maximum strategy to determine if the user has shown a sign of activity at least once in the window. The 2 hour rolling window is used to look ahead and check if activity has occurred during the next two hours. Therefore, the algorithm checks if the user has slept at least 2 hours from 8 PM to 3 AM to determine the sleep start time. After that, the first sign of activity in the next morning is considered as the wake up time. The upper bound of wake up time is 4 PM. If no activity has been detected before 4 PM, the algorithm discards this day and does not provide a sleep amount for it.

3.2.3 Voice Activity

Voice activity estimation is shown in Figure 7c. Since audio was converted to MP3 on the watch for bandwidth conservation and MP3 is a compressed format, audio data must first be converted to an uncompressed format (WAV). Next, each audio file is examined in 50ms frames with 20ms overlap. For each frame, various features that can help determine speech are computed. If these features fall within a dynamically computed threshold, they are categorized as speech. This process is known as Voice Activity Detection. The features computed for each frame are also used to determine who is speaking, in a process called Speaker Diarization. These features provide a fingerprint of the users voice. By first creating a model of the users voice, we can check how similar each voice segment is to the users voice. This tells us how much the individual user was speaking and how much of that speech is from other people (we can also determine how many other people the user speaks to).

4 Reflection

Our team learnt that it is difficult to define mood, let alone measure it. The relationship between changes in measurable attributes and mood is not a simple translation. Additionally, the team learnt that continuous sensing can break from events like software updates, as happened in the case of step count. Lastly, the team realized that signal processing is tricky. Signal data is noisy, and even with the 'accuracy' score associated with each measurement, it is difficult to get reliable signals from sensors. Activity detection often relies on arbitrarily or experimentally chosen thresholds. Creating an annotated dataset before would make the process of choosing thresholds easier and give us more confidence in the accuracy of our activity detection.

If the team had to do the project again, we would first create annotated datasets before testing our application. This would involve listening to some of our audio data and annotating segments of speech and segments of speech where the user is speaking. We could also keep a manual sleep log and use that to validate our sleep estimation. One of the reasons we were unable to do this is because we hadn't decided exactly which activities we would look for. Deciding these activities earlier would have given us time to annotate data and validate our algorithms.

Additionally, if the team had to do the project again, we would define our project scope better and set expectations from all stakeholders. More importantly, the team will seek feedback from stakeholders more regularly and act upon it sooner than later.

5 Contributions

5.1 Fahad

Fahad worked on designing and developing the data analysis frameworks. He also did sensor data preprocessing, physical activity estimation, sleep analysis, and anomaly detection. A good amount of work has also been done to improve the efficiency of the data analysis framework. For example, using efficient data structures and concurrency to process 30GB of data consisting of more than 35 millions data instances and 30,000 files.

5.2 Daniyal

In his role as a specialist, Daniyal provided the knowledge required to develop an extensible and reusable data collection and processing framework. He also provided the background required to tailor the Ups and Downs app to BD. As a programmer, Daniyal worked on developing the Smartwatch application and communication protocol between the watch and phone. He also helped on the data analysis framework and did the voice activity detection and speaker diarization. Finally, Daniyal managed and administered the self-hosted server used by the group for data anlaysis.

5.3 Naba

Naba took the lead in working on the mobile application. She created all the various client components, as well as integrating them with the processed sensor data. Usability and user experience were at the core of the design philosophy behind the client facing application. To this effect, Android principles like ShowCase were employed to create a modern and sleek mobile experience.

6 Specialist's Context

Wearable sensing and mobile health monitoring are very hot topics currently. Although many people are using sensor data to predict all sorts of things, from COPD exacerbations to stress, to even students GPAs. However, these sorts of studies are out of reach for most medical researchers because of the technical investment required to collect and handle data. Our project is a step towards an extensible framework for collecting sensor data from Android smartwatches and phones and a data processing framework for processing that data. With more work, this platform could make it much easier and quicker for doctors to start research studies that make use of continuous mobile sensing.

7 Future Work

As future work, the team plans to set up phone to server communication to automatically transfer data files.

Secondly, we can improve the accuracy of our activity detection by using annotated data to set better thresholds. Annotated data would also let us provide hard numbers for the accuracy of our algorithms. Furthermore, annotated data would allow us to use supervised machine learning which could potentially further improve the accuracy of our activity detection.

Thirdly, the anomaly detection we used is a general implementation. However, since we are interested in specific health conditions, we can optimize the anomaly detection. For example, changes in certain lung diseases occur over the span of days, changes in BD occur over weeks and SAD over months. Using disease specific information, we can fine tune the time windows used by the anomaly detection.

Finally, the team would like to work with medical researchers who can help correlate the data collected by our platform with diseases and disorders. For example, Ups and Downs could be in a study looking at blood markers indicating depressive states of SAD and BD.

8 Release Info

Video	Yes
Report	Yes
Source Code	No