Smartphone Gait Inference

A Major Qualifying Project

Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree in Bachelor Science

in

Computer Science and

Mathematical Sciences

By

____

Zachary Arnold

____

Danielle LaRose

This report represents the work of WPI undergraduate students submitted to the faculty as evidence of completion of a degree requirement. WPI routinely publishes these reports on its website without editorial or peer review. For more information about the projects program at WPI, please see http://www.wpi.edu/academics/ugradstudies/project-learning.html

Date: April, 2015

Project Advisors:

____

Prof. Emmanuel Agu, Advisor

____

Prof. Joseph Petruccelli, Co-Advisor

Abstract

In this project, a smartphone application was developed to identify and display a user's level of alcohol intoxication based on their gait.

The application uses analysis of the smartphone's accelerometer readings to detect differences in the user's gait associated with different levels of alcohol ingestion. Exploratory analysis of the accelerometer signals identified features in both the time and frequency domain which were likely to be indicative of the user's level of intoxication.

Machine learning methods were employed using these features, including the Random Forest, the J48 Decision Tree, a Naive Bayesian Network, and a Support Vector Machine. They were each trained on a set of data from seven volunteers and compared for their performance on a separate validation set. Of the three methods, the random forest method proved to be the most accurate classifier, yielding a 56% success rate on the training set, and a 70% success rate on the validation set. Using these results, an Android smartphone application was created and distributed to users for testing.

Each copy of the application includes the previously described model and will be trained with the individual user's new data. Based on only a week's worth of use for some users, classification accuracy improved; however, it did get worse for others. The unweighted mean accuracy for the group of study participants was 57%. With increased use over time, it is expected that this result will improve.

Table of Contents

Table of Figures

Table of Tables

Table of Equations

1. Introduction

1.1. Historical Background of Substance Abuse

Substance abuse, also known as drug abuse, is a patterned use of a substance (or drug) in which the user consumes it in amounts or with methods that are harmful to themselves or others. Historically, substance abuse has been a serious societal issue that impacts those who engage in the activity and to their immediate surroundings. The earliest recorded substance capable of being abused was alcohol, in the form of beer, invented by the ancient Mesopotamians [1]. Since then, people have found many different ways of ingesting and abusing alcohol and other substances.

Substance abuse is on the rise around the world and has substantially increased in the past few decades [2]. When observing the wide range of drug-related causes of harm, alcohol is listed to be the most harmful drug or substance. This is with respect to all areas concerning physical damage, mortality, mental malfunctioning, loss of tangibles and relationships, family adversity, and economic cost [3].

There are several sobering facts when considering substance abuse in the form of alcohol. Excessive alcohol use is the third leading lifestyle-related cause of death for people in the United States each year [4]. Globally, alcohol misuse is the fifth leading risk factor for premature death and disability; among people between the ages of 15 and 49, it is the leading risk factor [1]. In 2012, 3.3 million deaths, or 5.9 percent of all global deaths (7.6 percent for men and 4 percent for women), were attributable to alcohol consumption [2]. Additionally, alcohol contributes to over 200 diseases and injury-related health conditions, including alcohol dependence, liver cirrhosis, cancers, and injuries. In 2012, alcohol accounted for 5.1 percent of Disability Adjusted Life Years (DALYs) worldwide [1].

1.2. Alcohol Abuse by College Students

According to Dr. Johann Westmaas of the State University of New York at Stony Brook, college students are typically a group with high levels of substance abuse, especially alcohol abuse [5]. He developed questionnaire as to the motives for drinking in college students and revealed four general categories of motives. First, alcohol can be used as a coping mechanism to "avoid negative affective states" [5]. Second, some people may use alcohol to conform to society and "obtain peer acceptance or approval" [5]. Next, alcohol can be used to enhance a positive situation or mood. Last, alcohol is used in many social situations in order to "satisfy affiliative needs" [5]. There are a variety of reasons for why college students may engage in drinking, but research has uncovered that with regards to alcohol abuse and American college students:

- 80% have consumed alcohol in the past year [6]
- 45% binge drank in the last month (5 or more drinks in one sitting) [6]
- 30% meet the criteria for alcohol abuse [6]
- 6% meet the criteria for alcohol dependence [6]

College students are the most susceptible population to making poor decisions while under the influence of alcohol, including the operation of heavy machinery [4]. Underage drinking is strongly associated with many health and social problems among youth including alcohol-impaired driving, physical fighting, poor school performance, sexual activity, and smoking [2]. Additionally, excessive alcohol consumption contributes to more than 4,300 deaths among underage youth (persons less than 21 years of age) in the United States each year [4].

These statistics reveal that there is a need for research with regards to substance abuse in at least two areas: 1) Identifying root causes of substance abuse and 2) Identifying behaviors that are destructive not just to the self, but also others after harmful substances are ingested. The present project focused on identifying whether or not an individual was impaired and to what extent they were impaired.

1.3. Effects of Alcohol on the Body

Alcohol intoxication has a significant effect on how the human body operates. There are two major organs in the human body which respond first to the presence of alcohol: the heart and the brain [7]. Approximately ten minutes after the initial alcohol consumption, the heart rate begins to increase in order to filter out the toxins from the bloodstream through the kidneys. After about twenty minutes, the alcohol is able to penetrate the blood-brain barrier causing noticeable impacts to cognitive and neuromotor functions. One of these functions is human gait [8]. Human gait is a coordinated effort by the brain and other muscles to produce mobility in an effort to go somewhere. It is the manner in which an individual walks [9]. Alcohol impairment significantly impacts this coordination and can dramatically impact the ability to walk, jog, or run.

1.4. Current Limitations in Detection

Currently, there are tools available which detect alcohol intoxication in humans beyond empirical observation of the effects of alcohol on the body. The standard quantified unit of measure with regards to alcohol is Blood Alcohol Concentration (BAC) or Breath Alcohol Concentration (BrAC), which is the amount of alcohol in the bloodstream or in a person’s breath, respectively. BAC is expressed as the weight of ethanol, measured in grams, in 100 milliliters of blood. BrAC is the weight of ethanol in 210 liters of breath [10]. When a person drinks alcohol, it can either be metabolized by the liver (goes into the blood) or released from the body through breath, urine, or sweat [11]. BAC and BrAC can be measured by breath, blood, or urine tests [10].

If everyone was responsible with their substance ingestion, then there would be no need for external intervention to limit unsafe or destructive behaviors of others under the influence. Unfortunately, there is currently a heavy reliance on law enforcement agencies and other designated authorities to uphold substance abuse laws and to protect the public's well-being by managing situations involving substance abuse. Since the resources that law enforcement have are limited, more widely available ways to identify problematic levels of substance abuse are needed.

1.5. Current Detection Devices

1.5.1. SCRAM Continuous Alcohol Monitoring

SCRAM Continuous Alcohol Monitoring (SCRAM CAM), Figure 1, is an alcohol detection device that is worn continuously around the ankle. It is mainly used for high-risk, DUI (Driving Under the Influence) alcohol offenders. This device samples the user's perspiration every 30 minutes in order to measure their BAC levels. Since the bracelet detects alcohol intake by analyzing the person's sweat, the detection of alcohol could be influenced by other environmental alcohol substances, such as cleaning products. However, the device is able to identify the difference between alcohol consumed by the user and environmental alcohol substances in order to reduce false positive alcohol consumption readings. By continuously monitoring the user's BAC, the offender is no longer able to consume alcohol around their regularly scheduled tests. The user's data is then sent to secure servers and can be accessed by the user at a later point. Federal, state, and local court officials are also able to review this data in order to view alcohol alerts and compliance reports. This information can be used in court trials as evidence of good or bad behavior by the user [12]. In the event that a user is found to have consumed alcohol while wearing the SCRAM CAM, they are subject to fines, jail time, probation, parole, loss of driving privileges, or mandatory rehabilitative measures [11].

Figure 1 - SCRAM CAM Device

1.5.2. Kisai Intoxicated LCD Watch

The Kisai Intoxicated LCD Watch by TokyoFlash Japan, Figure 2, is a breathalyzer watch to determine the user's intoxication levels. In addition to being a normal watch, it has a built-in breathalyzer on its side that the user can use at any point. They simply blow into the breathalyzer and the watch determines their BAC level. The screen displays a graph of the user's BAC in addition to the LED turning green if their BAC is between 0.00% and 0.40%, yellow if it is between 0.41% and 0.60%, and red if it is over 0.61%. The watch also includes a sobriety game. For the game, there is a fixed square on the bottom center of the screen and a line which moves from left to right on the screen. The overall point of the game is to align the moving line with the fixed square, a task that becomes more difficult if the user is intoxicated. The closer the user is able to line up the moving line with the fixed square, the less likely they are to be considered highly intoxicated [13].

Figure 2 - Kisai Watch with the Sobriety Game

1.6. Detection and Smartphones

One device that could potentially be used to solve the problem of over-intoxication notification in real-time is the smartphone. Since the explosion of the smartphone market in 1999, these devices have gained exceptional computing capability. With their technology, some have started to replace higher-end PCs. These devices also come preloaded with many sensors including an accelerometer, a gyroscope, pressure gauges, temperature and light sensors, a GPS, and several others. With all of these sensors available, the average smartphone is a vast, untapped data collection, analysis, and response tool. Computerized interventions for those trying to manage their substance abuse have been identified as being appealing and useful [3].

1.7. The Goal of this Project

The overarching goal of this study was to investigate the feasibility of a smartphone application that passively infers how much alcohol has been consumed by its owner in order to provide just-in-time notifications of alcohol intoxication. This study focused on investigating and quantifying the effects of alcohol on human gait. Additionally, it focused on determining how well a smartphone could detect and dynamically respond to these conditions. The average smartphone does not come with a built-in heart rate monitor, but many do come with accelerometers and gyroscopes, which can be used to determine the impact of alcohol on gait. Aside from direct BAC or BrAC testing, neuromotor testing including analysis of gait by direct observation is the most reliable way to determine intoxication in humans [8]. A device that could quantify, test, and analyze such observations as well as respond in real-time would be valuable in the world today. Additionally, on the market today, there is no interface that allows users to leverage their device in a way that monitors this type of activity in real-time [14].

The health sector is another area that could potentially benefit from a massively available substance abuse detection application. There are compelling reasons in both emergency medical care and in alcohol or substance abuse recovery care. This type of application could potentially provide emergency medical physicians the advantage of being able to have a near accurate record of a patient's consumption history over the past few hours, days, or even weeks. Additionally, in caring for recovering addicts, their behaviors that lead to intoxication can be identified and assistance could be provided to help patients in real time. Moreover, this application could help on the individual level. If a person is made aware of their potentially harmful alcohol habits, they may make better decisions in regards to alcohol.

The overall goal of this project was to create a smartphone application to accurately detect variances in a person's gait when they are consuming alcohol regardless of phone placement, orientation, gender, weight, and gait activity. To do so, we 1) performed exploratory analysis of the accelerometer signal and its properties, 2) created and extracted features of the signal in the time and frequency domains, 3) performed an experiment on the features of gait while varying different everyday factors, 4) performed a campus-wide study on the effect of alcohol on gait features, 5) created a machine learning model to predict the effect of alcohol on gait, 6) created an Android application which implemented the machine learning model, and 7) evaluated the application's effectiveness and the model's improvements over time.

2. Related Work

There are several areas of research that are relevant to this project, including neuroscience, computer science, and mathematical modeling. The studies, described below, have been identified and were utilized to formulate the current project. Many of the studies reference the Romberg Test and the Romberg Quotient. The Romberg Test is the comparison of postural sway with eyes open versus with eyes closed. The Romberg Quotient is the ratio of the value of postural sway when the eyes are open to its value when the eyes are closed from the Romberg Test [8].

Ando *et al.* [8] conducted a two day study in 2007 to determine the effects of alcohol ingestion on neuromotor functions, postural sway, hand tremors, and reaction time in thirteen healthy males. The tests of hand tremors and reaction time were done with both the right and left hands. Before beginning the experiment each day, the researchers measured these four different functions of each participant using the Romberg's test. The participants then blindly received either alcohol or juice from the researchers. Half of the participants received juice on the first day and alcohol on the second day, while the other half received alcohol on the first day and juice on the second day. After the juice or alcohol was consumed each day, the researchers again measured the four different functions of the participants at varying time intervals. They determined that sway area and transversal sway tended to increase after alcohol ingestion, while postural sway after juice ingestion tended to decrease. Reaction time after alcohol ingestion was found to be slower in the left hand, but the tremor intensity after alcohol ingestion tended to be smaller. The results of the Romberg quotients of postural sway after drinking alcohol were found to be inconclusive [8].

Nieschalk *et al.* [15] observed the effects of low or moderate amounts of alcohol on the body with respect to equilibrium. They used BrAC levels to measure alcohol ingestion. The BrACs of the participants were measured 30 minutes after the ingestion of alcohol. The researchers then used Romberg-test conditions to measure the stance and postural stability of the participants, including their body sway path and body sway area with eyes open then eyes closed. They determined that “sway area was the most sensitive parameter for detecting increased body sway after alcohol ingestion” [15].

Demura and Uchiyama [7] observed the gait of fifteen male adults at normal and controlled tempos before alcohol ingestion and at 10, 20, and 30 minutes after alcohol ingestion. Gait was measured using a gait analysis apparatus to record time and spatial information. They also recorded blood pressure, heart rate, and one-leg stance at the same time intervals as the gait to examine the physiological responses of the participants. In the preliminary tests, all measurements were recorded twice to assure accuracy. The researchers analyzed the gait cycle, stance phase, gait velocity, cadence, stride, one step width, and a few other variables. They discovered that there was a decline in static balance ability, stride length, gait velocity, and cadence around 20 minutes after ingestion [7].

Nishiguchi *et al.* [16] compared the accuracy of the built-in tri-axial accelerometer of a smartphone to a similar, standalone accelerometer that was securely fastened. The researchers recruited seventeen men and thirteen women to partake in their study. The researchers developed a gait analysis application to be installed on each participant’s smartphone in order to measure acceleration. They then had the participants walk for 20 meters at their preferred speeds while their trunk accelerations were measured with both the smartphone accelerometer and the mounted accelerometer. They observed “statistically significant and considerable correlations” between the smartphone accelerometer and the mounted accelerometer and concluded that the smartphone and gait analysis application have “the capacity to quantify gait parameters with a degree of accuracy that is comparable to that of the [standalone] tri-axial accelerometer” [16].

Lee and Cho [17] used the sensors in a mobile phone to accurately detect and learn a person's activities based on their gait. They used Global-Local Co-Training (GLCT) with both labeled and unlabeled data in order to train the Mixture-of-Experts (ME) model. The GLCT is a type of co-training in which a global model and local model are used together. ME models are "based on several local experts to solve smaller problems and a gating network to combine the solutions from separate experts," in which each expert is a statistical model for a piece of an overall problem. By using GLCT to train the ME model, the performance of the ME model is increased because errors due to unlabeled data are minimized. There were two Android phone datasets used for this project, consisting of acceleration, orientation, and magnetic fields. These datasets contained both labeled and unlabeled data. The different human activity labels included still, walk, run, vehicle, and subway. The datasets were then used to train the ME model with and without the help of GLCT. After running the algorithm ten times to record the average accuracy, they determined that GLCT could be used as a training tool for the ME model to increase its accuracy. In addition, they were able to correctly label data given by the datasets with a high degree of accuracy [17].

Kao *et al.* [18] created a phone-based system to identify anomalies in the gait of a person when walking under the influence of alcohol. Their system used the tri-axis accelerometer in the HTC Magic smartphone. It was designed to sense the alcohol intake of a person and then record the location and time. They had one female and two males participate in their study. They had the participants put the smartphone in their pocket and walk an initial 40 meters to get a baseline. The participants were then provided wine in 30 minute increments until they were at a BAC of over 0.05%. They then walked another 40 meters after alcohol ingestion. The researchers observed a noticeable difference in the gait of each participant after the alcohol ingestion, but established that further research should be done with a larger pool of participants using more than just the accelerometer for data collection [18]. The amount of alcohol consumed was also not inferred from gait anomalies, a direction we will investigate.

Wang *et al*. [3] created a smartphone application to work with a Bluetooth breathalyzer in order to record and track the alcohol consumption of patients who are recovering from alcohol dependence. They created this application to be used to track daily sobriety after the completion of an alcohol withdrawal treatment. The application also had artistic drawings depicting typical symptoms patients would have at each stage of the recovery process in order to educate patients and reduce relapse. The researchers conducted a four-week study on eleven recovering alcohol-dependent participants who had already completed their alcohol withdrawal treatment. They provided the participants with a Bluetooth breathalyzer and the SoberDiary application for their smartphone. The application consists of several main features including performing the breath alcohol test, reviewing personal progress, sharing recovery process, acquiring managing skills, and inputting current emotions. When the breathalyzer is used, it sends the user's BrAC to the smartphone and is displayed on the application. The data was also transferred to a backend server so that researchers and their psychologists could analyze the data at a later time. The application allows the users to record their progress in alcohol recovery. The more tests the user performs on themselves and the more they stay on track, the more rewards they receive from the application, including monetary coupons. In addition, the application allows users to view their ranking compared to other patients using the application. The application also displays the amount of money saved by the user from not purchasing alcohol. Overall, it was found that the application tended to reduce the craving for alcohol. Although the study produced promising results, some of the data was found insufficient because of when the breathalyzer tests were being performed. The participants needed to perform the breathalyzer test at least two times a day. This meant that the participants could take the test prior to drinking alcohol, in which case the data would not reflect that they had consumed alcohol [3].

Weaver *et al.* [14] reviewed the many different alcohol-related smartphone applications to determine their accuracy and relevance in measuring a person's BAC. The researchers used data from a prior field-based study, The Patron Offending and Intoxication in Night-Time Entertainment Districts (POINTED) [14]. POINTED was a massive project done in Australia from late 2011 to mid-2012 which surveyed almost 7000 people about their alcohol usage and recorded their BAC [19]. The researchers randomly selected the data from four POINTED participants. This information included gender, age, number of drinks consumed, the time period of drinking, and their respective BAC. The researchers then searched the Apple iTunes and Android Google Play stores for alcohol-related applications. They found about 500 applications, but only 98 were BAC-related applications. They found that a majority of the applications were for entertainment purposes and tended to encourage alcohol consumption, rather than for the anticipated health promotion purposes. For the remaining 98 applications, the researchers entered the required information into the applications (i.e. height, weight, gender, etc.) and recorded what the application determined to be each participant's BAC. They then compared the BAC values from the application to the actual BAC values recorded from the POINTED data. They discovered that the applications were highly unreliable and inaccurate. This was due to lack of information required, such as weight and gender, as well as different algorithms to determine BAC [14].

2.1. Alcohol Application Features

There are many existing applications that are similar to the one our study produced. Many of the applications contain an alcohol consumption tracker and drink recorder. These are used to record what type of alcoholic beverage was consumed by the user and when it was consumed. However, it could become bothersome to the user to have to continuously input the data for when alcohol was consumed. Additionally, the user's recollection of what and when they drank might not be completely accurate or truthful.

There are numerous applications that are used to calculate the BAC of the user. These applications are either stand-alone BAC calculators or are built-in to other applications. The user inputs the alcohol they have consumed and approximately when it was consumed, and they are provided with an estimate of their BAC. However, many of the BAC calculators are unreliable and the user might not input their alcohol consumption accurately. This could cause the user to believe that they are below the legal alcohol limit even if they are extremely intoxicated. Table 1 summarizes several such applications.

Name	Platform	Type	Description
AlcoDroid Alcohol Tracker [20]	Android	Alcohol consumption tracker, drink recorder, and blood alcohol content calculator	Provides a BAC estimate based on the logged drinks and notifies the user when they are sober or nearly sober.
SoberApp - Alcohol Calculator [20]	Android	Alcohol consumption tracker, drink recorder, and blood alcohol content calculator.	Provides a BAC estimate based on the logged drinks and determines when it is acceptable to drive again.
Blood Alcohol Calculator [20]	Android	Blood alcohol content calculator	Provides a BAC estimate either based on a predefined drink list or based on logged drinks and determines when it is acceptable to drive again.
My Daily Journal [20]	iPhone	Drink recorder	Record alcohol consumption, triggers for drinking, and feelings while drinking.
DrinkControl [20]	Andriod/ iPhone	Alcohol consumption tracker, drink recorder	Record alcohol consumption. Warns the user when they are drinking excessively.
DrunkGuard [21]	iPhone	Alcohol consumption tracker, blood alcohol content calculator	Record alcohol consumption. Locks certain applications or phone if it thinks the user drank too much. Tells them when they will be sober again. Calls a cab for the user if they are excessively drunk.

Table 1 - Table of Related Applications

3. Methodology

In order to accomplish the goal of being able to classify alcohol intoxication levels passively using gait patterns from an accelerometer, we first found it beneficial to understand certain mechanics of human gait analysis. This included understanding how the accelerometer signals abstracted the physical activity of walking. We then performed exploratory analysis on the accelerometer signal to search for patterns, properties, and features in both the time and frequency domains. Once features of the signal were selected and extracted, we performed an experiment to assess the variation in feature responses when varying certain everyday walking conditions. Next, we performed a multiple user study to collect data in order to determine the possibility of a machine learner accurately classifying reported intoxication levels from extracted features of the accelerometer signal. This machine learning model was then used in building the mobile application and would make improvements to classification over time. Once the application was built, we then distributed a usability survey to users who volunteered in order to gather insight on the application's effectiveness and improvements to onboard classification.

3.1. Gait Analysis Techniques

Gait analysis, by definition, is the method used to asses a person's gait and determine any abnormalities in it [22]. For the purpose of this research, our focus of gait analysis was on smartphone accelerometer responses to changes in gait due to alcohol consumed. Human locomotion produces an electromagnetic pulse in the tri-axial accelerometer sensors of the smartphone. These accelerometer sensors turn the pulses into a signal which represents the gravitational pull on each sensor at any given moment in time [23]. When many of these readings occur consecutively in time, a "signal" is created. This signal can then be processed and analyzed using computational and mathematical methods. Every smartphone hardware design is different; however, the typical accelerometer signal has a sampling rate of 200Hz and values between -19.82 to 19.82 Newtons (N), or about +/- 2G's, for the three separate planes (x, y, z) of the phone as shown in Figure 3 [23].

Figure 3 - Accelerometer Orientation in a Smartphone

We performed analysis in three separate areas using MATLAB R2014b running an Intel PC with a Core i7 processor (1.9 Ghz) and 8Gb of RAM. These areas were 1) common numerical estimation techniques, 2) analyzing properties of the signal in the time domain as well as 3) the frequency domain.

3.1.1. Discovering Signal Properties Using Numerical Methods

During our initial analysis of a signal, it was best to discover some of its properties through visualization. Numerical methods allow for further discovery by providing estimation techniques that reveal more properties of the signal, such as inferring the signal's periodicity. Figure 4 is an example of common human gait data collected using a smartphone accelerometer, which we then transferred to MATLAB for further analysis. The action being performed, visualized in Figure 4, is a simple walk in a straight-line down a hallway. For continuity purposes, throughout the remainder of the signal analysis sections, when "example data" is referenced, it will refer to Figure 4:

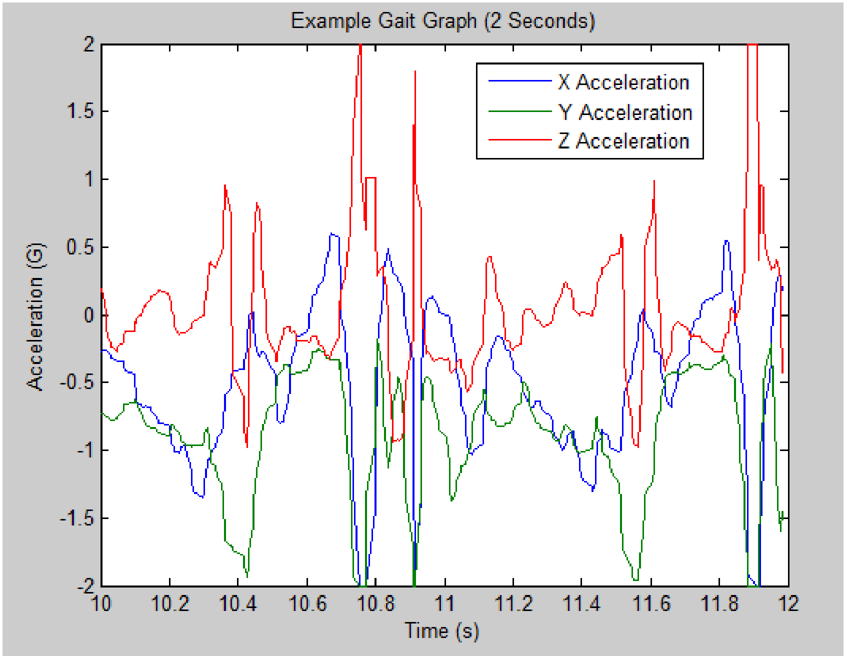


Figure 4 - Example Gait Data

From Figure 4, the x, y, and z axis accelerations intuitively seem to be correlated with regards to the motion being performed, but are not equal. Since the phone could be in several different orientations while someone is walking, such as in a bag or a pocket, we needed to normalize the outputs of the accelerometer. Therefore, we analyzed the gravity-corrected magnitude of the accelerometer sensors. The accelerometer sensors are recorded to the phone in vector form, so at time t, a tuple is stored: $a(t) = \{x_t, y_t, z_t\}$

For some amount of time while data is being recorded, the gravity-corrected magnitude of the vector $a(t)$ is calculated using Equation 1, assuming $a(t)$ has n equally spaced readings.

$$b(t) = \sqrt{{x_t}^2 + {y_t}^2 + {z_t}^2} - \frac{\sum_{i=1}^{n} \sqrt{{x_i}^2 + {y_i}^2 + {z_i}^2}}{n}$$

Equation 1 – Gravity Corrected Magnitude Formula

This allowed for a normalized data set from the accelerometer regardless of phone orientation. Another observation we made from the diagram was that the accelerometers in smartphones are highly sensitive to change, causing the data to be noisy. Signal noise describes unwanted or unknown modifications to a signal in any of the following ways: capturing, storing, transmission, processing, and conversion [24]. In order to smooth the inputs in the time domain for MATLAB and the later algorithm to help classify gait patterns, we filtered the signal with a simple moving average using Equation 2.

$$c(t) = \frac{\sum_{i=0}^{m} b(t-i)}{m}, where\ m = 5$$

Equation 2 - Simple Moving Average

After smoothing the signal, we used MATLAB's curve fitting toolbox to further analyze and discover rudimentary properties about the signal. Of the available fitting functions, Fourier estimation provided an accurate method to approximate the example signal from above with little error. The Fourier model in MATLAB for a polynomial in discrete time can be expressed in its general form as Equation 3.

$$f(t) = a_0 + \sum_{j=1}^{k} [a_j \cos(j\omega t) + b_j \sin(j\omega t)]$$

Equation 3 - Fourier Model Equation [25]

Where:

$\omega =$ The fundamental frequency of the data in Hertz (Hz), which is calculated by MATLAB using the Fast Fourier Transform (FFT) of the signal and a technique called the Harmonic Product Spectrum [26].

$n =$ The number of terms in the series. $(1 \leq k \leq 8)$

$a_0 =$ Models a constant (intercept) term in the data and is associated with the $i = 0$ cosine term.

Using MATLAB's trust-region, non-linear, least squares method with options DiffMinChange = 1.0e-8, DiffMaxChange = 0.1, MaxFunEvals = 600, and MaxIter = 400 to solve for all a_j and b_j, we found the most accurate estimation (in terms of the smallest sum of squared error) on the example data to be found in Figure 5.

```
General model Fourier8:
   f(x) =
         a0 + a1*cos(x*w) + b1*sin(x*w) +
         a2*cos(2*x*w) + b2*sin(2*x*w) + a3*cos(3*x*w) + b3*sin(3*x*w) +
         a4*cos(4*x*w) + b4*sin(4*x*w) + a5*cos(5*x*w) + b5*sin(5*x*w) +
         a6*cos(6*x*w) + b6*sin(6*x*w) + a7*cos(7*x*w) + b7*sin(7*x*w) +
         a8*cos(8*x*w) + b8*sin(8*x*w)
Coefficients (with 95% confidence bounds):
    a0 =    -0.4819  (-0.4955, -0.4684)
    a1 =    -0.1988  (-0.2184, -0.1793)
    b1 =     0.1366  (0.1167, 0.1566)
    a2 =    -0.1945  (-0.2146, -0.1744)
    b2 =     0.1009  (0.07845, 0.1233)
    a3 =    -0.0737  (-0.0938, -0.0536)
    b3 =   -0.06354  (-0.08375, -0.04334)
    a4 =     0.0485  (0.01335, 0.08365)
    b4 =     0.2436  (0.2235, 0.2638)
    a5 =     -0.243  (-0.2624, -0.2236)
    b5 =   -0.01869  (-0.06, 0.02262)
    a6 =    0.08634  (0.03996, 0.1327)
    b6 =    -0.2388  (-0.2638, -0.2137)
    a7 =    -0.1086  (-0.158, -0.0591)
    b7 =     0.2184  (0.1884, 0.2484)
    a8 =     0.1033  (0.04985, 0.1567)
    b8 =    -0.2081  (-0.2394, -0.1769)
    w =      5.494  (5.492, 5.496)

Goodness of fit:
  SSE: 193.3
  R-square: 0.687
  Adjusted R-square: 0.6844
  RMSE: 0.3104
```

Figure 5 - MATLAB Fourier Estimation Results on Approximation of Example Data

This model was analyzed with respect to the example data in terms of SSE, RMSE, and R-squared value. The Sum of Squared Error (SSE), Equation 4, is a measure of the discrepancy between the data and an estimation model [27]. The SSE for this model was 193.3 and the Root Mean Squared Error (RMSE) of this estimation is 0.3104.

$$SSE = \sum_{t=1}^{n} (b(t) - f(t))^2$$

Equation 4 - Sum of Squares due to Error

Where:

$b(t) =$ the data of the model at time t

$f(t) =$ the fitted value from the model at time t

Another useful statistic to evaluate the “goodness of fit” for this model is the R-squared value. This statistic measures how successful the fit is in explaining the variation of the data. The R-square can be calculated by taking the ratio of the sum of squares of the regression with the total sum of squares. Put another way, R-square is the square of the correlation between the response values and the fitted response values. The sum of squares regression is calculated using Equation 5 [28]. The value for this model’s R-square is 0.687. This means that the model accounts for 68.7% of the variance in the data.

$$SSR = \sum_{t=1}^{n} (\hat{y}_t - \bar{y})^2$$

Equation 5 - Sum of Squares of the Regression

$\hat{y}_t =$ the fitted value from the model at time t.

$\bar{y} =$ the mean value of the response data

Similarly, the total sum of squares is calculated using Equation 6.

$$SST = \sum_{t=1}^{n} (y_t - \bar{y})^2$$

Equation 6 - Total Sum of Squares

Where:

$y_t =$ the data of the model at time t.

$\bar{y} =$ the mean value of the response data

The resulting formula for R-square can be seen in Equation 7.

$$R - square = \frac{SSR}{SST}$$

Equation 7 - R-Square

A diagram of the example data with its Fourier estimation can be seen in Figure 6.

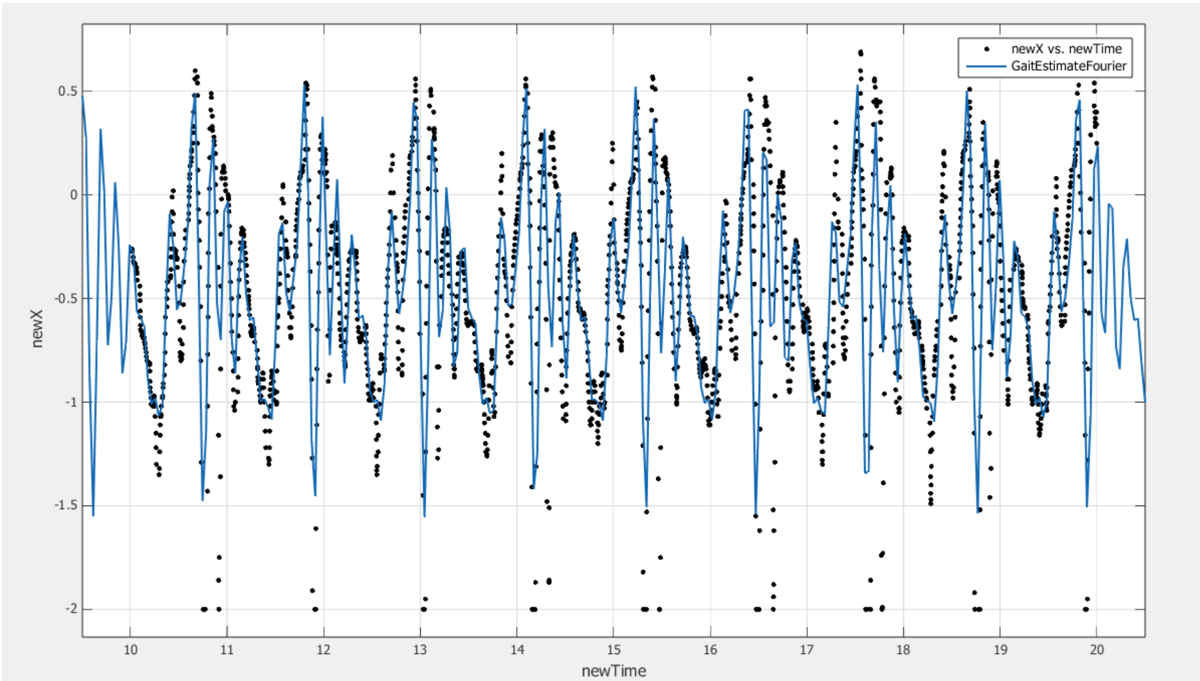


Figure 6 - Fourier Estimation of Gate Data MATLAB

Figure 6 indicates that the model (blue line) approximates the data reasonably well. However, there are some values above and below the curve for which the model does not account. While no model is perfect, this one illustrates well the properties of the signal. Performing a Fourier estimation, like the one in the figure above, also provided us with one key insight that helped signal analysis feature extraction: the data from the gravity-corrected magnitude of the signal was periodic in nature. Intuitively, this makes sense because walking is a repetitive motion. Hence, the model selected in the diagram and resulting equation above is periodic. This periodicity allowed us to calculate a few important features when signal analysis was performed in the time and frequency domains.

Numerical analysis of the example data provided us with a few key insights. It revealed the periodicity of walking and the range of fundamental frequencies at which walking occurs, which is approximately between 1 and 5 Hz. The fundamental frequency of the example data was 5.542 however, which is outside of this range. This could be the result of error in the model's calculation or estimation of the fundamental frequency. Additionally, the model could be including noise in the calculation for the fundamental frequency. For example, a phone jostling in the pocket of a user will shake more often that each individual step and would add error to the model fitting. This showed that this noise needed to be filtered out of the signal which eventually allowed us to extract features in the time and frequency domain.

3.1.2. Time Domain Feature Discovery and Extraction

A time-series is a collection of data points recorded at uniform, measured intervals in time [29]. Specifically in the context of this research, the data were the recorded (x, y, z) values of the smartphone's accelerometer with respect to time. Much of the analyses that occurred in the time-domain involved extracting the properties of the signal as a "feature of gait." A gait feature, in this context, is a property of the signal representing gait which can be observed or calculated using the signal from the phone's raw accelerometer data [30]. These features are all based not on analyzing the entirety of the signal, but rather a window of about 10 seconds, which equates to around 2000 observations, at a time. The time domain features we used were: 1) the number of steps taken in a window, 2) the average step length for that window, 3) the average time between steps, 4) the average velocity, 5) the cadence, as well as the 6) skewness and 7) kurtosis of each signal. We chose many of these features because of their usage in similar projects [9].

In alcohol research, it is well known that gait is impaired with the presence of alcohol. One specific impact of impairment is the number of steps taken in a given interval of time [7]. The number of steps taken in an interval can be calculated by finding the number of local maxima of the gravity corrected magnitude of the accelerometer signals which exceed one standard deviation from the mean of the signal [31]. This was done using MATLAB's findpeaks algorithm. The 15 local maxima marked by triangles in Figure 7 correspond to the 15 steps calculated by this algorithm.

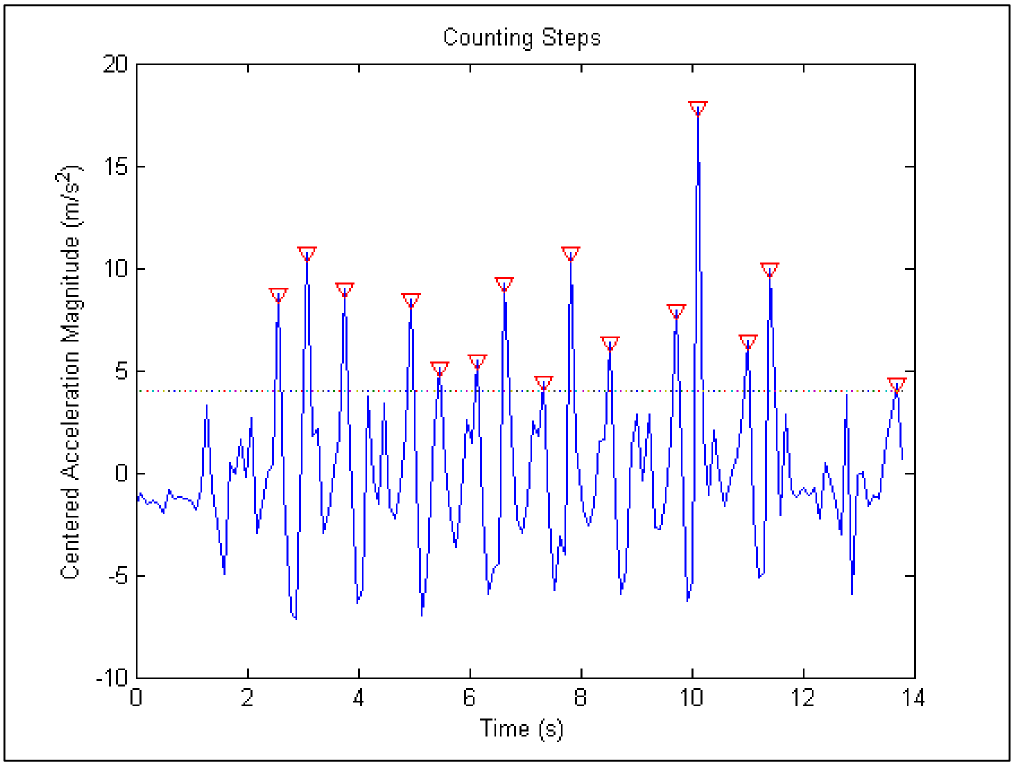


Figure 7 - Example Data with Number of Steps Highlighted

Similarly to the number of steps being affected, the average step length and step time also undergo a change with the presence of alcohol. According to the findings of Kao *et al*., as the amount of alcohol present in an individual increases, so will the difference between sober gait stretch and step time and intoxicated gait stretch and step time [18]. In the context of this research, gait stretch is equivalent to the average step length. A snippet of the MATLAB implementation for average step length and average step time can be seen in Figure 8 and Figure 9 respectively with the full implementation in Appendix V.

```
function dasl = diffAvgStepLength(s,t,l,len)
...
for i=2:s
    timeOfStep = [timeOfStep t(l(i))];
    stepLength = v*(timeOfStep(i) - timeOfStep(i-1));
    stepLengthArray = [stepLengthArray stepLength];

end

diffAvg = asl - stepLengthArray(1);

for j=2:s
    difference = asl - stepLengthArray(j);
    diffAvg = [diffAvg difference];
end

dasl = sum(diffAvg);

end
```

Figure 8 – MATLAB Code snippet of Difference from Average Step Length

```
function dasl = diffAvgStepLength(s,t,l,len)
...
for i=2:s
    timeOfStep = [timeOfStep t(l(i))];
    stepLength = v*(timeOfStep(i) - timeOfStep(i-1));
    stepLengthArray = [stepLengthArray stepLength];

end

diffAvg = asl - stepLengthArray(1);

for j=2:s
    difference = asl - stepLengthArray(j);
    diffAvg = [diffAvg difference];
end

dasl = sum(diffAvg);

end
```

Figure 9 – MATLAB Code snippet of Difference from Average Step Time

Figure 10 is a visualization on what each of these features looks like in a signal.

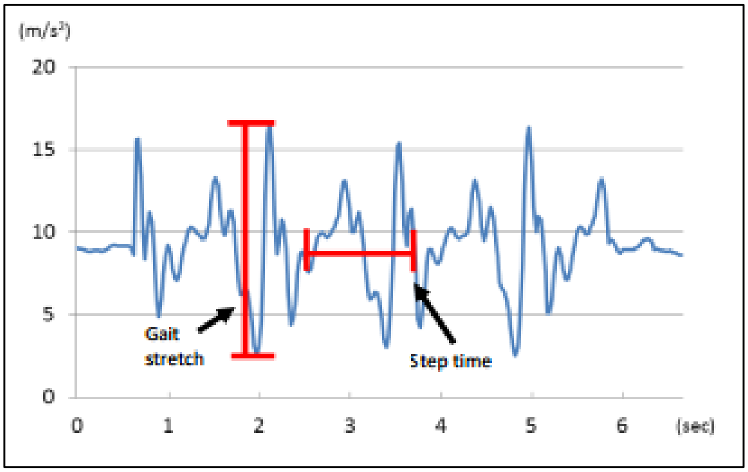


Figure 10 - Example Data with Gait Stretch and Step Time Highlighted [18]

The last of the features that are abstractions of reality represented through a signal are gait velocity and gait cadence [9]. Gait velocity is a ratio of the total distance covered divided by the total number of steps. Similarly, gait cadence is a ratio of the number of steps taken divided by the amount of time taken. Both of these features were used in phone-based gait verification projects; however, research suggests that velocity and cadence are also affected by alcohol [7].

Additionally, we added the skewness and kurtosis of the marginal distribution of the signal to help further characterize the data for the classifier. Skewness is a measure of symmetry, or more precisely, the lack of symmetry in a dataset [32]. A data set from the accelerometer is symmetric if it looks the same to the left and right of the center point. Similarly, kurtosis is a measure of whether the data are peaked or flat relative to a normal distribution [32]. That is, a data set with high kurtosis has a distinct peak near the mean, declines rather rapidly, and has a heavy tail. Table 2 lists the time domain features and their definitions for reference.

Time Domain Feature	Colloquial Definition
Number of Steps [9]	The number of steps taken in a given time interval
Average Step Length [18]	Average in the distance covered by each step
Average Step Time [18]	Average in the time covered by each step
Gait Velocity [9]	Ratio of the total distance covered by the total time
Cadence [9]	Ratio of the total number of steps by the total time
Skewness [9]	Asymmetry of the signal distribution
Kurtosis [9]	"Peakedness" of the distribution and the heaviness of its tail

Table 2 - Time Domain Gait Features

Analysis of the signal in the time domain allowed us to extract relevant features for the classifiers that were dependent on time. Another area of interesting analysis for the signal was examining it in the frequency domain.

3.1.3. Frequency Domain Analysis

The frequency domain is popular for signal analysis. Most common frequency analyses occur on sound or other high frequency vibrations; however, accelerometer data is also just a signal which can be processed using frequency domain techniques. The features extracted from the frequency domain also help to classify the presence of alcohol in gait. In order to transform the time series data into the frequency domain, we used the discrete Fourier transform (DFT), as seen in Equation 8.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi k \frac{n}{N}}, k = 0, \ldots, N-1$$

Equation 8 - The Discrete Fourier Transform

One helpful tool to provide insight on the presence of alcohol in gait was the one-sided power spectral density (PSD) of a signal. The power spectrum was also generated using the discrete Fourier transform. A periodogram is one way to estimate the power spectral density of the signal and is calculated from the DFT. Unlike the simple DFT, it describes how the variance of data in the time domain is distributed over the frequency components into which the signal can be decomposed [33]. Often times in signal analysis, there is a considerable amount of noise in the periodogram of the PSD. To overcome this, several techniques of overlapping periodograms and averaging them together has allowed for smoother PSD's to be produced. We calculated the one-sided PSD of the example data using the Welch's overlapped segment averaging estimator algorithm in MATLAB and is plotted in Figure 11.

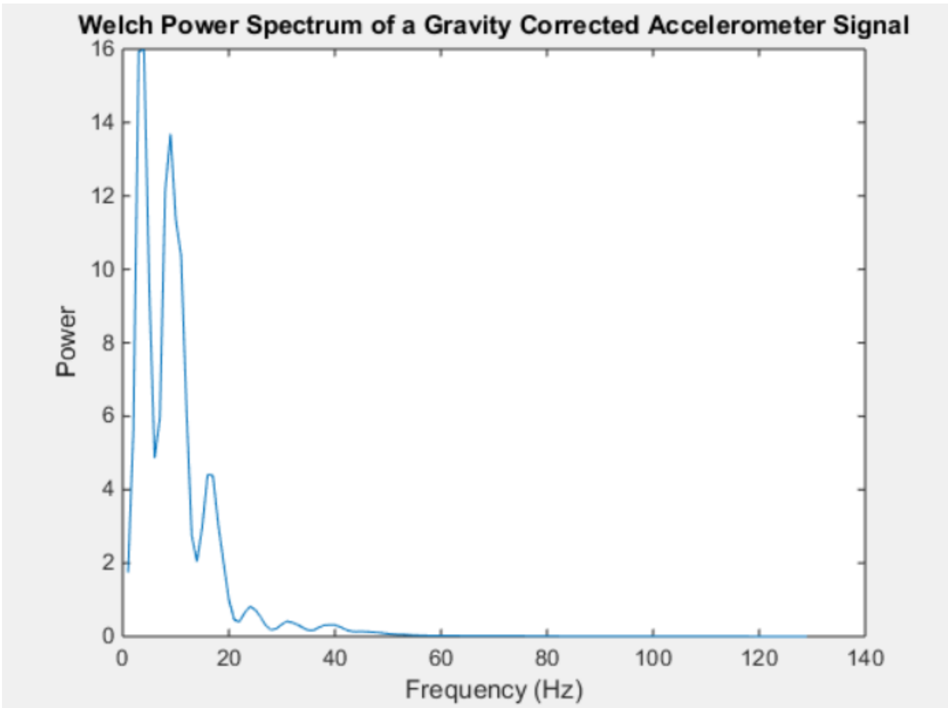


Figure 11 - PSD of Example Data Using Welch Algorithm

The Welch's overlapped segment averaging estimator algorithm for calculating the PSD divides the signal into the longest possible sections to obtain as close to but not exceed 8 segments with 50% overlap. Each section is windowed with a Hamming window. The modified periodograms are averaged to obtain the PSD estimate. If the length of x cannot be divided exactly into an integer number of sections with 50% overlap, x is truncated accordingly [34]. From the PSD, we can gather where the energy of the signal is distributed in relation to its frequency. Not surprisingly, the highest peak can be found at the fundamental frequency of the signal. When the activity of walking is being performed, the fundamental frequency of the signal is typically between 1-5Hz [9].

The frequency domain features of the signal that we extracted for gait classification were 1) the average power of the signal, 2) the ratio of high to low energy peaks of the PSD, 3) the signal to noise ratio of the signal, and 4) the total harmonic distortion of the signal. We selected these based on their effectiveness in previous research involving passive gait verification [18] [9].

The average power of a signal is the mean of the total power underneath the curve of the PSD estimate for a signal [35]. The technique to calculate average power is a rectangle approximation of the integral of the signal's power spectral density (PSD).

Another frequency domain feature shown to help in gait classification is the ratio the energy in high frequency peaks to the energy in low frequency peaks in the power spectral density estimate [9]. Intuitively though, the ratio of energy in high to low frequency peaks helps to determine the distribution of energy in the frequency of a signal. A value of 1 means that there is an equal amount of energy in the high and low frequency peaks. The peaks are discovered by looking at the PSD estimate of the signal. The algorithm will only consider an even number of peaks, so it calculates $d = floor(\frac{n}{2})$ where n is the number of peaks discovered. The area under the peaks (or energy) greater than d are then divided by the area under the peaks less than d. A snippet of the MATLAB implementation for peak ratio can be seen in Figure 12 with the full implementation in Appendix V.

```
function r = ratio(x,y,z)
...
high = [];
low = [];

for i=1:numSP
    if(sp(i) < (mean(sp)))
        low = [low sp(i)];
    elseif (sp(i) > (mean(sp)))
        high = [high sp(i)];
    end
end

r = sum(low)./sum(high);

end
```

Figure 12 – MATLAB Code Snippet of Peak Ratio

Signal to Noise ratio (SNR) is a value typically calculated in decibels relative to the carrier (dBc) of a real-valued input signal. The SNR is typically used in radio broadcast transmission to determine how much of the signal is distorted by noise in the airwaves. For an accelerometer, noise could be generated by any outside force on the phone which is not directly attributable to walking. For example, if the sidewalk the user is on is being excavated using a jackhammer. Mathematically, the SNR is determined using the periodogram of the signal. The periodogram was smoothed using a Kaiser window with $\beta = 38$. The Kaiser window is a windowing technique that is commonly used in digital signal processing, especially when smoothing periodograms. By default, the result generated using MATLAB considers the energy contained within the peaks of the first six harmonics (including the fundamental frequency) to be the signal. Thus, the power outside of these harmonics is considered to be the noise of the signal for the ratio value [36].

Another feature that we selected to help determine other properties of the signal in the frequency domain was the Total Harmonic Distortion (THD). Typically, the THD is used in audio analysis to determine the amount of distortion the signal undergoes when being played through another source, such as a stereo speaker. Using the accelerometer, this value will help us to determine the extent to which the fundamental frequency of walking is distorted by outside influences that affect the readings of the accelerometer, such as hand movements and phone repositioning. The THD is typically calculated in dBc of a real-valued signal. It is determined from the fundamental frequency and the first five harmonics using a periodogram of the input signal from the phone [37]. The measurement is most commonly defined as the ratio of the RMS amplitude of a set of higher harmonic frequencies to the RMS amplitude of the first harmonic, or fundamental, frequency [38]. Assuming V_1 is energy contained within the peak of the PSD at the fundamental frequency and V_i are the energy contained within the harmonics, THD can be calculated in general form using Equation 9.

$$THD_F = \frac{\sqrt{V_2^2 + V_3^2 + V_4^2 + V_5^2 + V_6^2}}{V_1}$$

Equation 9 - Total Harmonic Distortion

A lower THD, along with the SNR, indicates that a clean signal is being analyzed and that there is a small amount of contamination in the signal.

Table 3 lists the frequency domain features and their definitions.

Frequency Domain Feature	Colloquial Definition
Average Power [9]	The variance per unit time
Ratio of Spectral Peaks [9]	Ratio of the energies of low and high frequency bands
SNR	The ratio of the power of whole signal to that of its computed noise
THD	The distortion of the whole signal when compared to its harmonics

Table 3 - Frequency Domain Gait Features

Ultimately, all of the time and frequency domain features we generated from time-series data were used to aid in the classification of alcohol in gait using machine learning techniques.

3.2. Experiment to Determine Everyday Factors Which Influence Gait Features

We performed an experiment to determine some everyday factors which influence gait features. We did so under the assumption that the only activity being performed was normal, rhythmic, human walking. Other activities, such as running, jogging, jumping, climbing stairs, etc., were not considered for the experiment.

3.2.1. Design

There were different covariates and experimental factors used for the purpose of this experiment. Since covariates are subject-specific, they cannot be altered during the experiment and cannot be considered experimental factors. The experimental conditions, or factors, that we tested that have been shown in research to impact gait are shown in Table 4 and are presented in Figure 13 [39]:

Factor and Representation	Levels	Description
Surface Incline **(A)**	Incline	approximately 5% grade
	Decline	approximately -5% grade
Body Placement of Phone **(B)**	Front of body	Front pocket
	Back of body	Back pocket
Material **(C)**	Loose	Gym shorts, sweatpants
	Tight	Jeans, dress pants
Shoe Type **(D)**	Hard	Dress shoes, high heels, hard tennis shoes
	Soft	Slippers, Dr. Scholls inserts, soft tennis shoes
In Attaché **(ABCD)**	Yes	Phone in purse or briefcase
	No	Phone not in purse or briefcase

Table 4 - Table of Factors Tested

We were the two participants for this experiment and are shown in Table 5:

Participant	Gender	Weight
Subject 1	Female	155
Subject 2	Male	280

Table 5 - Table of Participants in Experiment

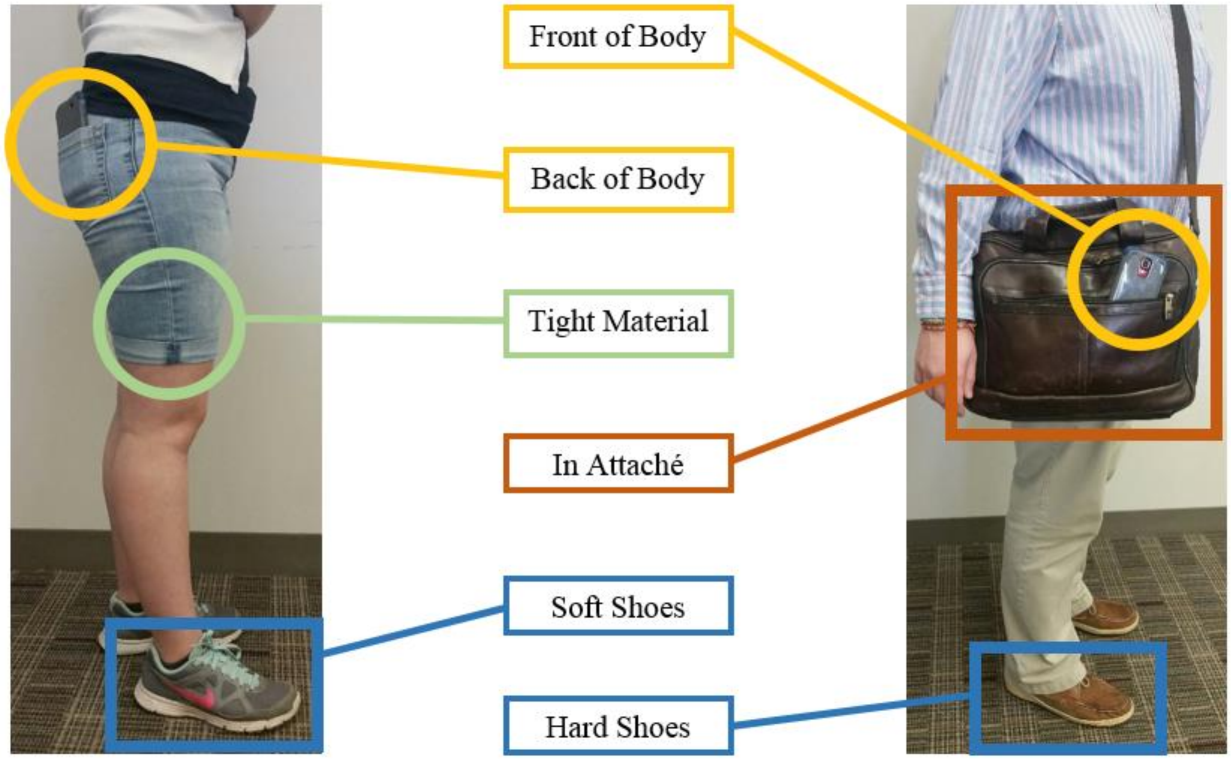


Figure 13 - Experimental Factor Expression Images

A brute force testing of the effects of all factors individually would lead to an extremely large number of trial runs. Therefore, we chose a factorial experimental design instead. Using a Resolution V one-half replication of a 2^5 design, we confounded the interaction of factors ABCD with E to reduce the trial runs by half and only considered $2^{(5-1)} = 2^4$ trial runs for the fractional factorial design. We repeated these sixteen runs three times each in a randomized order. This repetition was done in order to gather enough data to make a determination on measurement error's impact on the results. Additionally, these repetitions allowed for more precise estimates and greater power for tests. We performed these forty-eight runs using an indoor ramp of approximately 5% incline to assess the varying factor's impact on gait. For each trial run, we collected accelerometer data in the x, y, and z directions along with a demarcation of time relative to the beginning of the experiment for each reading. These readings were wirelessly streamed from the smartphones using a web server connected to MATLAB on a host computer. Each trial run consisted of the experimental factors being varied from Table 4 and the subject walking in a straight line for 15 seconds while data was being gathered.

3.2.2. Results and Analysis

Once all the data were gathered, we analyzed it individually per subject. We made this decision due to the difference of expressions for some of the experimental factors by each subject. For example, subject 1 used an in attaché of a small purse and hard shoes of high heels while subject 2 used an in attaché of a large briefcase and hard shoes of work boots. We then input the accelerometer data and corresponding times of the different trial runs into MATLAB for calculations of features from Table 2 and Table 3 (Time domain and Frequency domain respectively). Analysis consisted of a simple linear regression conducted on each feature generated. In addition to the experimental factors A, B, C, D, and E, all other possible interactions of the factors had to be considered (AB, AC, AD, BCD, BC, BD, ACD, CD, ABD, ABC).

Once that was complete, a regression model for each feature took the form of Equation 10.

$$gaitFeature \sim I\ (intercept) + ABCD + A + B + C + D + AB + AC + AD + BCD + BC + BD + ACD + CD + ABD + ABC$$

Equation 10 - Gait Feature Regression Model Equation

Below is one example of regression that is fully calculated:

This case considered the female with the feature number of steps, or numSteps, as response. This numSteps feature is a calculation of the signal peaks above one standard deviation away from the mean of the gravity corrected magnitude of the signal from Equation 1.

The estimated coefficients and test results are shown in Table 6.

Factors	Estimate	Test Statistic	p-value
(Intercept)	29.708	18.904	6.3086e-19
E	-2.0417	-1.2992	0.20316
A	0.20833	0.13257	0.89536
B	-1.4167	-0.90147	0.37407
C	0.66667	0.42422	0.67424
D	5.4583	3.4733	0.0014968
AB	-0.33333	-0.21211	0.83337
AC	1.25	0.79542	0.43223
AD	-0.70833	-0.45074	0.65522
AE	-0.875	-0.55679	0.58154
BC	-0.45833	-0.29165	0.77243
BD	1.3333	0.84845	0.40249
BE	1.1667	0.74239	0.46327
CD	1.5833	1.0075	0.32124
CE	-3.0833	-1.962	0.058506
DE	0.20833	0.13257	0.89536

Table 6 - numSteps Calculations

Number of observations: 48
Error degrees of freedom: 32
Root Mean Squared Error: 10.9
p-value = 0.172

The estimate column is a calculation which determines about how much impact a factor or interaction of factors had on the response (or feature). For example, in the above table, factor "D" causes the numSteps response to vary on average by about 5.5 steps. This estimate along with the test statistic is used in calculating the p-value for each factor or interaction.

In testing for statistical significance, p-values < 0.05 for each factor or interaction of factors were considered statistically significant. Based on the p-values in the above example, the only factor to have an impact on this particular model for this particular feature of gait was D, hard versus soft shoes.

Table 7 denotes all factors for all gait features determined to be statistically significant at the p < 0.05 level (denoted by *) and p < 0.01 level (denoted by **).

Feature	Factor	Estimate	SE	tStat	p-value
Number of Steps – Female	D	5.4583	1.5715	3.4733	0.0014968**
Number of Steps – Male	E	-4.3542	1.5715	-2.7707	0.0092379**
Step Length – Female	D	23.372	6.3103	3.7036	0.00079897**
	CE	-14.445	6.3106	-2.2891	0.028821*
Step Length – Male	E	-19.237	6.399	-3.0063	0.0051122**
Step Time – Female	D	16.642	5.3606	3.1046	0.0039698**
	AE	-13.177	5.3606	-2.4582	0.019561*
	CE	-14.346	5.3606	-2.6762	0.011641*
Step Time – Male	E	-16.16	7.6893	-2.1017	0.043545*
Average Power – Female	E	-2697.5	205.3	-13.139	1.9367e-14**
	A	-584.76	205.3	-2.8483	0.0076189**
	B	518.39	205.3	2.525	0.016722*
	C	-635.4	205.3	-3.095	0.0040698**
	D	422.7	205.3	2.0589	0.047717*
	AD	726.08	205.3	3.5367	0.0012609**
	AE	885.92	205.3	4.3153	0.00014334**
	BC	-657.49	205.3	-3.2026	0.0030756**
	BE	-492.74	205.3	-2.4001	0.022382*
	CE	910.96	205.3	4.4372	0.0001011**
Average Power – Male	E	-4256.3	473.49	-8.9894	2.8735e-10**
	A	-2320.2	473.49	-4.9003	2.6545e-05**
	B	3542	473.49	7.4806	1.6255e-08**
	C	-1579.2	473.49	-3.3352	0.0021673**
	BE	-2014.5	473.49	-4.2546	0.00017042**
Kurtosis – Female	E	-0.51935	0.16236	-3.1988	0.0031062**
Kurtosis – Male	AD	-0.44098	0.18768	-2.3496	0.025129*
	CE	0.44098	0.18768	2.3496	0.025129*
Skewness – Female	E	-0.24938	0.049629	-5.0248	1.8489e-05**
	AB	0.10551	0.049629	2.126	0.041311*
Skewness – Male	AD	-0.15223	0.059695	-2.5502	0.015754*
	CE	0.24593	0.059695	4.1198	0.00024986**

Table 7 - Table of Statistically Significant Experimental Factors from Regression

In summary, the three features in the time-domain that produced statistically significant results under experimental conditions were the number of steps taken, the average step length, and the average step time. The most impactful experimental factor for the female subject was the hard versus soft soled shoes (factor D). In each regression which returned results of a p-value less than 0.05, the factor that was involved was D. There were a few results where D and other factors (A, B, C) were interacting with each other; however, due to the Resolution V design confounding interactions of factors at the third level, the interaction of these other factors with D can be considered insignificant. By the hierarchy of significance, we are able to assume that the simplest explanation, D, is correct and had an impact on gait. Similarly, the most impactful experimental factor for the male subject was whether the phone was in attaché or not (factor E). Also, by the hierarchy of significance, we are able to assume that the factor ABCD, or E, was the simplest explanation on the impact of gait. Possibly the more exciting result here is that there was not an overwhelming response difference when factors were varied.

3.3. Study to Determine Impact of Alcohol on Human Population Gait

Once we understood the effects of walking conditions with respect to the feature set, we conducted a study to gather data of sober gait and intoxicated gait. The study sought to discover whether or not the presence of alcohol had a distinct, quantifiable effect on human gait, and more importantly, whether or not a smartphone could detect and analyze that effect. Although there exists previous research that is similar to the purpose of this study, there are differences in the approach and overall purpose of this project compared to the others. We leveraged the research of Kao *et al*. [18] in the sense of using the built-in accelerometer of the smartphone to determine walking. However, unlike their research, we did not provided any alcohol to the participants in this study. Additionally, they did not aim to infer the amount of alcohol consumed based on gait.

3.3.1. Design

Once the study was approved by the WPI Institutional Review Board, we sent emails to WPI students and faculty to inform them about an opportunity to participate in this study (See Appendix I). Our friends and families were also recruited via word of mouth and emails. Those recruited were informed that they must have a smartphone to be considered for the study, since the application being used was only for smartphones. Those who expressed interest in the study received and email with a short description of the study and were asked to meet with us to sign a consent form in order to complete the questionnaire (See Appendix II). Once the form was signed, they were then emailed a link to an online survey which included a short study overview and an inclusion/exclusion questionnaire (See Appendix III). The first page consisted of a general overview of the study, a description of the questions being asked, and that a limited number of respondents would be asked to participate in the study. The AUDIT questionnaire was used in the survey to determine each respondent's eligibility to participate. To be eligible, the respondent must have been at least 21, owned an Android smartphone, and have received a score of 8 or less out of 40 on the questionnaire. If they were determined to be eligible, they were then invited to meet with us in order to read and sign the study consent form (See Appendix IV). At that time, they were given the opportunity to have their questions answered or concerns addressed. Once they had signed the letter of consent, we downloaded the data collection and recording application to their smartphone. The participants were asked to continuously run the application over a two week period with an option to continue for an additional two weeks depending on the amount and quality of data collected. They were then asked to continue their daily routine as usual, without changing it due to participation in the study.

Participants were given the ability to opt-out of the study at any point during it. If they chose to do so, they were given the option to either allow their data, up to that point, to remain in the study or have their data permanently deleted and removed from the study. Participants were also randomly assigned a number for the duration of the study. We did not know the identity of the participants and were not able to associate their data to them. We only knew the randomized numbers and the data associated with those numbers.

3.3.1.1. *Data Gathering Application for the Study*

In order to collect data remotely from users, we needed to create an application that could be installed on smartphones which reported data to the us. In order to satisfy requirements of the WPI Institutional Review Board, the application had to be such that no participant could access his/her data on the phone, that that data was stored securely wherever it was transmitted to, and that it would not require participants to engage in a way that would reveal their identity to anyone but the us.

In order to accomplish this task, we leveraged a third party library for the Android operating system called Funf. Funf is an open-sensing framework which allows the data of a smartphone's sensors to be recorded on some interval of time and automatically transferred to a remote location for analysis [40]. Using this framework, we created a lightweight application that could be installed on any Android smartphone. The application shown in Figure 14 consisted of one screen which was initiated on startup:

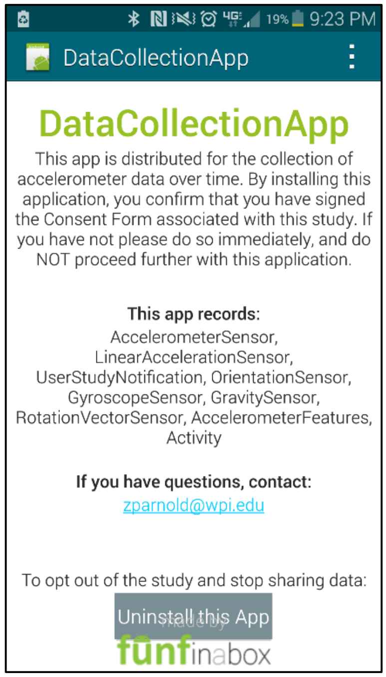


Figure 14 - Opening Screen of Data Collection App

This screen reiterated the participant's ability to opt out and destroy all of their data at any time. After this screen had been viewed there was no longer any interaction with the user interface. In the background, data was sampled from the different phone sensors and stored in 10Mb database files. Once per hour, these files were uploaded to a secure server. Upon receipt of the data, the server processed the data and filtered out unusable data. Unusable data was considered to be when the participant was not actually walking with the phone (i.e. stationary).

We determined that there were several different ways that participants could have recorded their alcohol consumption throughout their participation in the study. We discussed having participants record their alcohol intake in real-time, on a daily basis, or on a weekly basis. There were also options of having the participants record their intake through an online survey or a paper survey. The research of Del Boca and Darkes [41] on self-reporting influenced the overall decision for which response methods to use. They determined that daily estimation tended to allow for more accurate information and that computer/internet assisted reporting allowed the respondents to respond in a more truthful manner [41]. Therefore, we decided that the participants would record their alcohol consumption twice daily using the data collection application that had been installed on their smartphone. Approximately every twelve hours, the application asked the user to record any amount and the type of alcohol consumed on the recording application. The participants were not provided alcohol nor in any way asked or encouraged to ingest any alcohol, but they were encouraged to record on the application if any alcohol was consumed.

3.3.2. Results and Analysis

We then analyzed the data collected to determine if intoxication levels could be detected in the gait of the smartphone's user. All data remained secure and subjects were only identifiable by randomly assigned identifiers.

The first step in analysis was to filter out unwanted data. Much of the data was recorded when the smartphone was stationary due to the low probability of the study application recording at the moment a participant was walking. After filtering the data, each participant's accelerometer data was matched to their consumption data. In order to determine whether or not alcohol consumption affected gait, the data were manually tagged on the next day corresponding to consumption data gathered from the survey. Since participants were asked to report both what they drank and approximately when they drank it, recordings from the phone that were gathered at or around that time were labeled with an updated drink count.

Each segment of data of about 10 seconds had features extracted using the previously mentioned methods. Below are plots of a randomly selected sober gait (Figure 15) compared to gait associated with an estimated BAC of 0.117, or > 6 drinks (Figure 16).

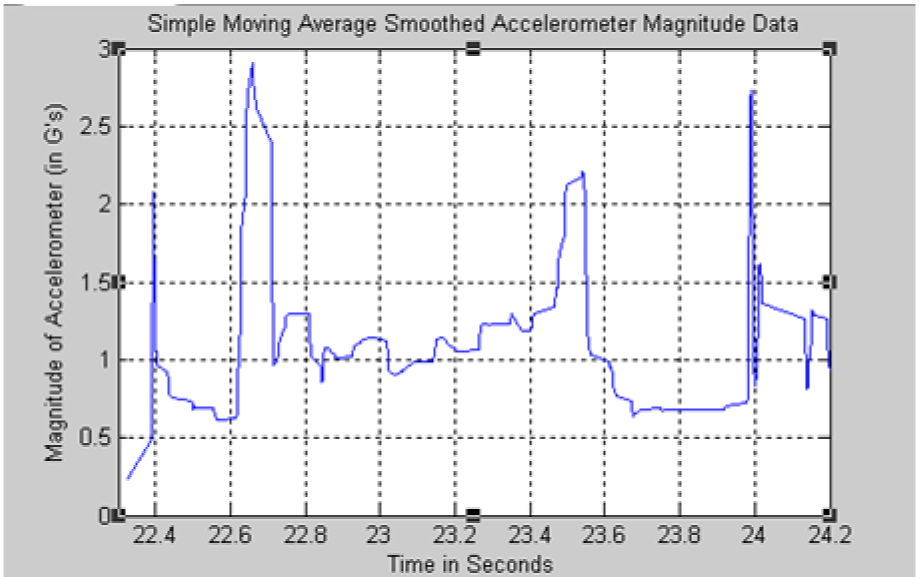


Figure 15 - MATLAB Plot of Sober Gait in Time Domain

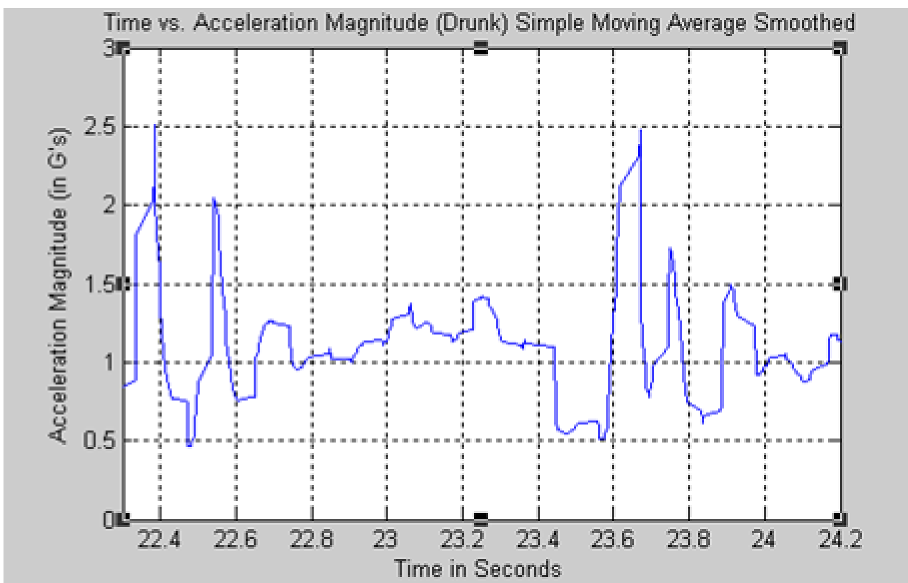


Figure 16- MATLAB Plot of Intoxicated Gait in Time Domain

Intuitively, we can see a difference in sober and intoxicated gait in the signal response of the accelerometer. This can be seen in the more sharply defined "steps" in sober gait vs. intoxicated gait as well as the amount of additional noise between steps present in intoxicated gait. However, it was still unknown whether or not a machine learner could accurately classify the amount of alcohol present given the feature set. Table 8 details the features that were extracted from the signals plotted in Figures 15 and 16.

	Sober	Intoxicated
numSteps	12	12
cadence	1.1638	1.3327
skewness	1.6739	0.81458
kurtosis	6.1112	3.6834
gaitVelocity	0.096984	0.11106
stepLength	-1.9231	-1.9231
ratio	0.47392	0.79152
stepTime	3.6547	6.9889
avgPower	32307	13379
SNR	-2.9788	-5.1409
THD	-2.0745	-14.41
numDrinks	0	12

Table 8 - Sober vs Intoxicated Feature Comparison

The number of steps and step length for this recording window (10 sec.) are the same. Interestingly though, that the cadence and velocity were quicker when this particular study candidate was walking. There is a noticeable difference in all of the features. The SNR measurements suggest that intoxicated gait produces a less noisy signal, and the THD measurements suggest that there less distortion in the signal as the number of drinks increase. For each feature, Table 9 contains each gait feature response for varying levels of intoxication.

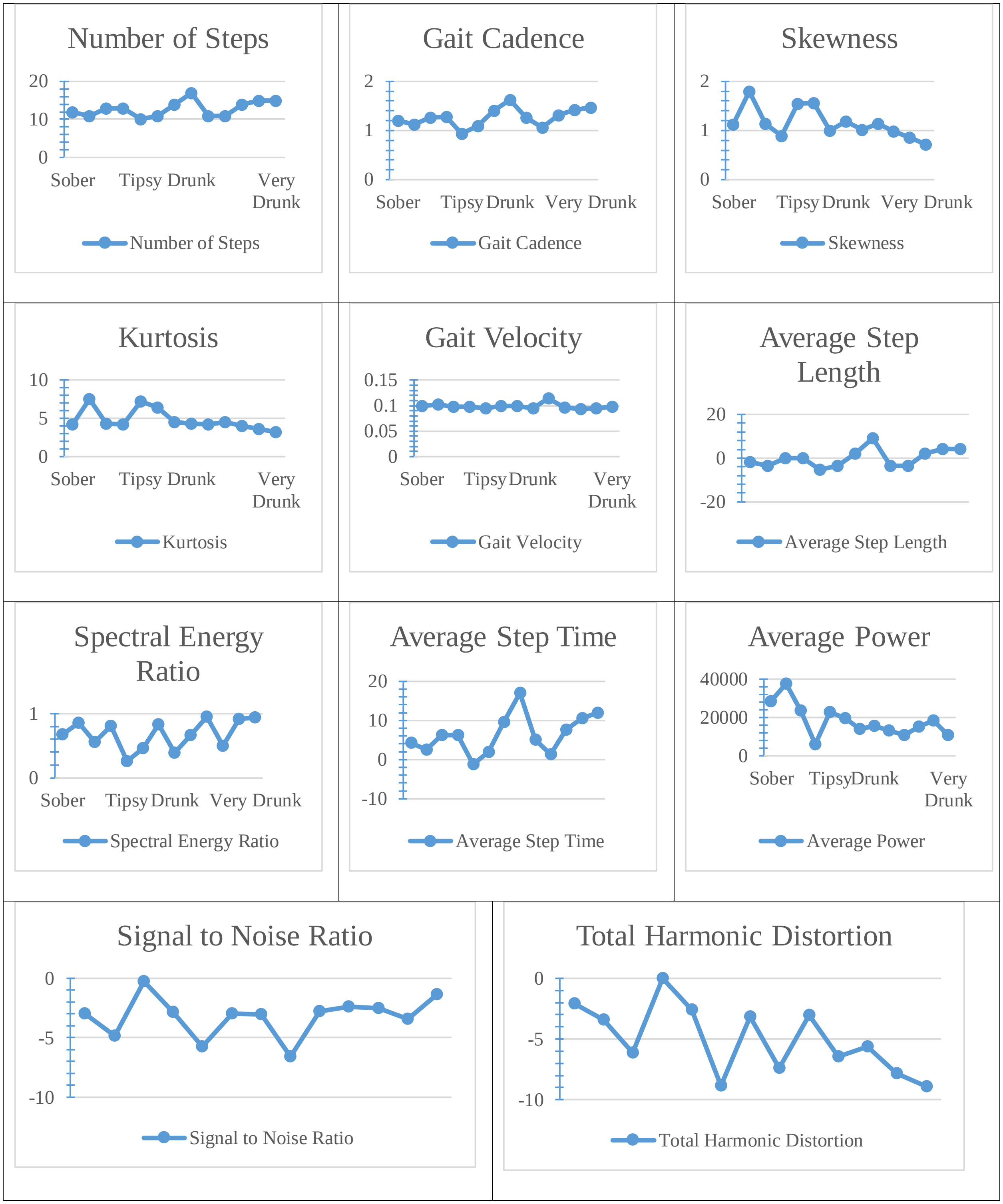


Table 9 - Features for Varying Levels of Intoxication

In order to understand classification, we first needed to have an understanding of machine learning. According to Ron Kohavi, a professor of practice at Stanford University on Machine Learning:

> In Knowledge Discovery, machine learning is most commonly used to mean the application of induction algorithms, which is one step in the knowledge discovery process. This is similar to the definition of empirical learning or inductive learning in Readings in Machine Learning by Shavlik and Dietterich. Note that in their definition, training examples are "externally supplied," whereas here they are assumed to be supplied by a previous stage of the knowledge discovery process. Machine Learning is the field of scientific study that concentrates on induction algorithms and on other algorithms that can be said to "learn" [42].

Waikato Environment for Knowledge Analysis (Weka) is a tool that is well known in the computer science community for machine learning and was leveraged for this project to help classify the samples of data. Of the 209 samples that were analyzed, the minimum number of drinks was 0 and the maximum number reported was 12. The mean of this data set was 4.643 drinks, and the standard deviation was 3.896 drinks. Because of the mean and standard deviation of the data set, we used a discretization method on the data in order to improve the accuracy of the model. Samples were placed into one of three "bins" for classification. These bins were approximately 0-2 drinks, 3-6 drinks, and above 6 drinks consumed. Figure 17 is a diagram of the Weka preprocessor with instances placed into bins based on number of drinks.

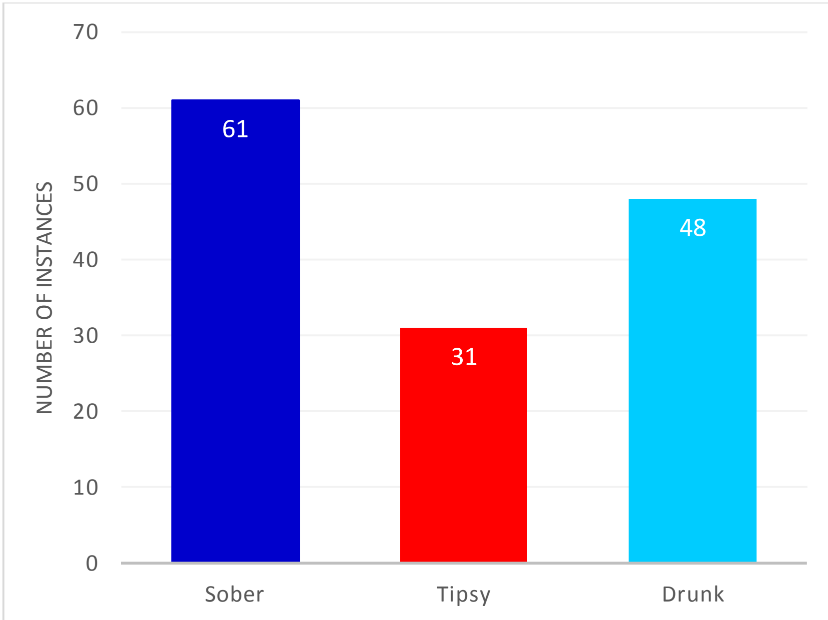


Figure 17 - WEKA Preprocessing Output

The blue color equates to the approximately 0-2 drink bin, the red is the 3-6 drink bin, and the cyan color is the more than 6 drinks bin. The bins contain 61, 31, and 48 pre-labeled examples respectively.

We compared several classifiers (machine learners) inside of Weka to determine ultimately which one would be loaded onto the phone for onboard classification. These were the Naïve Bayes Net, the J48 Decision Tree, the Support Vector Machine, and the Random Forest.

Before discussing the classifiers themselves, the methods used to compare them will first be introduced. In each classifier output, Weka also computes and displays many useful statistics on how the classifier performed on example and test data. For each test run, five-fold cross validation was used to ensure that the model was not over-fitting or memorizing the data [43]. K-fold cross validation runs the classifier K times, but each time 1/K's worth of the dataset is withheld and used to test (or validate) the model [43]. By doing this, the model can be tested to ensure that it is actually predicting and not memorizing data.

In order to determine model accuracy, there were three descriptive statistics used: the percent of correctly classified instances, the weighted F-score, and the weighted receiver operating characteristic (ROC) Area. These statistics can all be related using a confusion matrix. A confusion matrix is a specific table layout that allows visualization of the performance of a supervised machine learning algorithm [43]. An illustrative diagram, Figure 18, helps to understand how a confusion matrix is organized.

		Nature	
		+	-
Prediction	+	True positive	False positive (Type I error)
	-	False negative (Type II error)	True negative

Figure 18 - Confusion Matrix Diagram

Figure 18 is specific to a binary classifier. That is, a classifier which is choosing between two classes. The entries on the diagonal from top left to bottom right are the correctly classified instances. The entries on the other diagonal are the instances classified incorrectly as the other class.

Equation 11 displays the percentage of correctly classified instances is a measure of the number of correctly classified instances over the total number of instances.

$$\%Correct = \frac{\#\ Correctly\ Classified\ Instances}{\#\ of\ Total\ Instances} * 100$$

Equation 11 - Percentage of Correctly Classified Instances

This measure was useful for a baseline understanding of how well the model performed on the test (cross-validated) data when trained; therefore, we selected it as a baseline comparison statistic. In terms of a confusion matrix, this is the number of entries on the true diagonal divided by the sum of all of the entries of the matrix.

The weighted F-score is a weighted average of the F-scores of each class, where the weights are based on the number of instances in each class. (A class here refers to the predictor's output). The individual F-score is a measure of precision and recall for each class. Precision and recall relate the number of true positives, false positives, and false negatives. Precision can be seen in Equation 12:

$$Precision = \frac{\#\ of\ True\ Positives}{(\#\ True\ Positives + \#\ False\ Positives)}$$

Equation 12 - Precision of a Model

Recall can also be calculated as the top left entry in the confusion matrix divided by the top row of the confusion matrix, as shown in Equation 13.

$$Recall = \frac{\#\ of\ True\ Positives}{(\#\ True\ Positives + \#\ False\ Negatives)}$$

Equation 13 - Recall of a Model

Like recall, precision is also calculated from the confusion matrix of the classifier. This is the top left entry divided by the left column of the confusion matrix. These two measures are combined to create the F-score as in Equation 14.

$$F\ Score = 2 * \frac{Precision * Recall}{(Precision + Recall)}$$

Equation 14 - Individual F-Score

We selected the F-score because of its usefulness to represent model accuracy. The F-score helps relate the precision and recall of a model. The weighted F-score is a value between 0 and 1. The closer to 1 that the value of the F-score is, the more useful the model is for prediction.

The ROC Area, or area under the receiver operating characteristic curve, is a plot of the true positive rate vs. the false positive rate when the classifier makes decisions on instances [43]. The area under the curve is a value from 0.5 to 1. An AROC value close to 0.5 denotes that the true and false positive rate are the same and that the classifier is essentially guessing between classes. A value closer to 1 denotes that the classifier is performing accurately. The curve is generated by plotting the true positive rate of the model on the false positive rate. The true positive rate is calculated from the confusion matrix by dividing the top left entry by the sum of the entries in the left column. Similarly the false positive rate is calculated by dividing the bottom right entry of the matrix by the sum of the entries in the right column. An example of an ROC curve plot is seen in Figure 19.

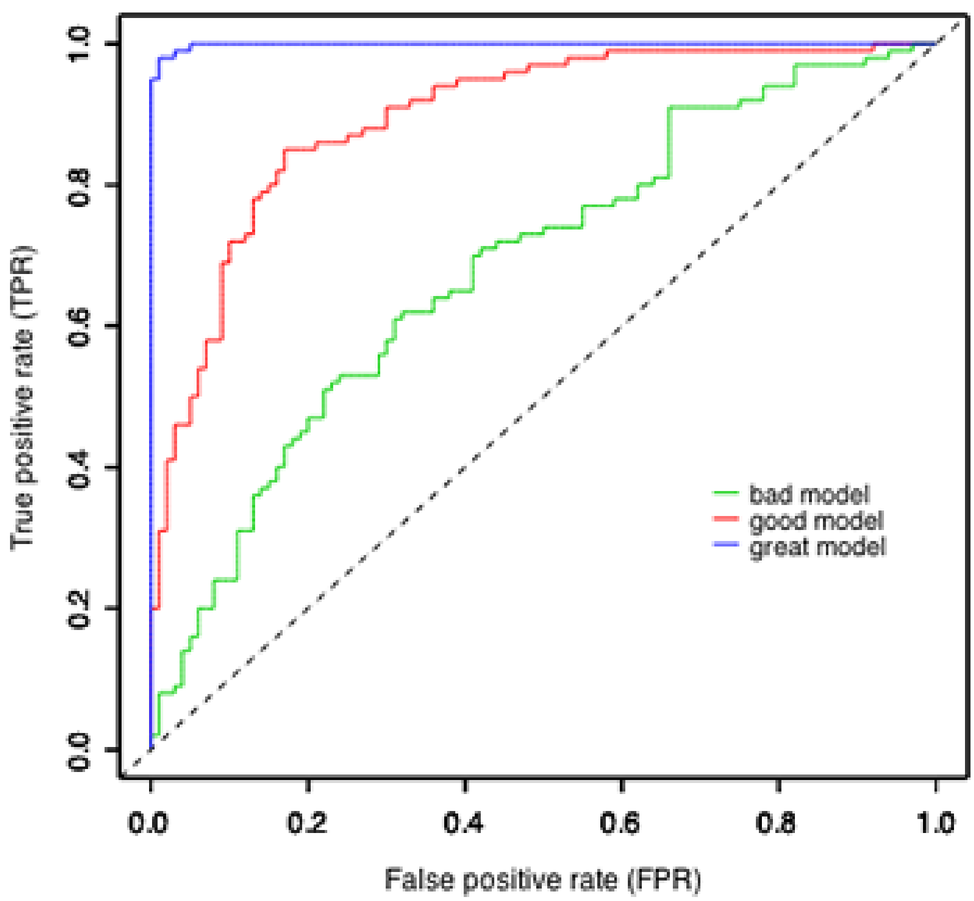


Figure 19 - Example ROC Curve

We chose this statistic to help validate the F-score and model accuracy statistics. The AROC value will add to the understanding of the “goodness” of a model on the data. With these statistics in mind, it is then possible to decide between different classifiers.

Initially in exploring ideal classifiers and their options, it is good to perform a Random Forest technique to determine about how “learnable” the data are using a decision tree. A Random Forest is a collection of trees each considering a random number of features, a random first feature, and a random depth to find the best possible classification [43]. The random forest on the data collected in the study produced the most accurate classification of all of the investigated methods. It had an accuracy of 56%, an F-score of 0.629, and an AROC of 0.658. Next, we analyzed other classifiers which could be selected for the model.

The first was a Naïve Bayesian Network. Naïve Bayes is a decision tree model that is based on conditional dependence, with the assumption that none of the features is dependent on the others (naïve) [43]. A Bayes Net is generated from the conditional probability table (CPT) of the model, which is also called the full joint probability distribution. The full joint probability distribution is often called the probability model as well. There are several different rules for constructing a classifier from the probability model. Weka, in its Naïve Bayes package, uses the “Maximum A Posteriori” or MAP rule. MAP is similar to the Fisher method of Maximum Likelihood (ML); but unlike ML, the MAP rule uses an optimization objective which uses a prior probability distribution with the one it is trying to maximize [43]. Sometimes we have *a priori* information about the physical process whose parameters we want to estimate. For each model regeneration, the *a priori* information is the previous model. Bayes' theorem, Equation 15, shows the way for incorporating prior information in the estimation process:

$$P(\theta|x) = \frac{P(x|\theta)P(\theta)}{P(x)}$$

Equation 15 - Bayes Theorem [44]

Where $P(\theta|x)$ is the posterior probability of class θ given predictor data (or features) $\boldsymbol{x}$**.** $P(\theta)$ is the probability of a class. For our data, this is 1/3. $P(x)$ is the prior probability of the predictor data. $P(x|\theta)$ is the likelihood of the predictor data given the class assignment. In laymen's terms this is, "what is the probability of a class given this set of observations."

The term on the left hand side of the equation is called the *posterior*. On the right hand side, the numerator is the product of the likelihood term and the prior term. The denominator serves as a normalization term so that the *posterior* probability density equation integrates to 1. Thus, Bayesian inference produces the *Maximum A Posteriori* (MAP) estimate in Equation 16.

$$\underset{\theta}{argmax}\ P(x) = \underset{\theta}{argmax}\ P(x|\theta)P(\theta)$$

Equation 16 - MAP Rule for Naive Bayesian Networks [44]

In theory, this helps to simulate the probability that there is conditional dependence of some subset of the features of the probability distribution on each other. We selected the Naïve Bayes Net for its versatility in being able to make inferences on the data independently of causal relationships. This is especially true in limited understanding, more exploratory data analysis.

However, the Bayes net performed worst of the four classifiers considered. Despite its simplicity, it was unable to draw useful inferences from the data about whether or not the presence of alcohol existed in gait. It had 42.1429% accuracy, a weighted F-score of 0.393, and a ROC area of 0.564. From these statistics, the model is essentially flipping a coin on whether or not an instance belongs to a class.

Naïve Bayes	A ($<$ 3 Drinks)	B ($3 \le$ Drinks ≤ 6)	C ($>$ 6 Drinks)
A ($<$ 3 Drinks)	67	18	21
B ($3 \le$ Drinks ≤ 6)	6	2	5
C ($>$ 6 Drinks)	34	25	31

Table 10 - Naive Bayes Classifier Confusion Matrix

As the confusion matrix in Table 10 indicates, this classifier does not perform well at classifying the class B. in the middle. That is, both the sober and heavily intoxicated instances were guessed with more frequency than the middle category.

We then considered the Support Vector Machine (SVM) classifier. In Weka, this is implemented by using John Platt's Sequential Minimal Optimization (SMO) algorithm. SVM's work by constructing a maximum margin separator, which is a decision boundary with the largest possible distance between example points. This separator helps the SVM create something called a "linear separating hyperplane." This is a line that passes through the space between maximally separated pre-classified instances [43]. Figure 20 is a visual of this definition.

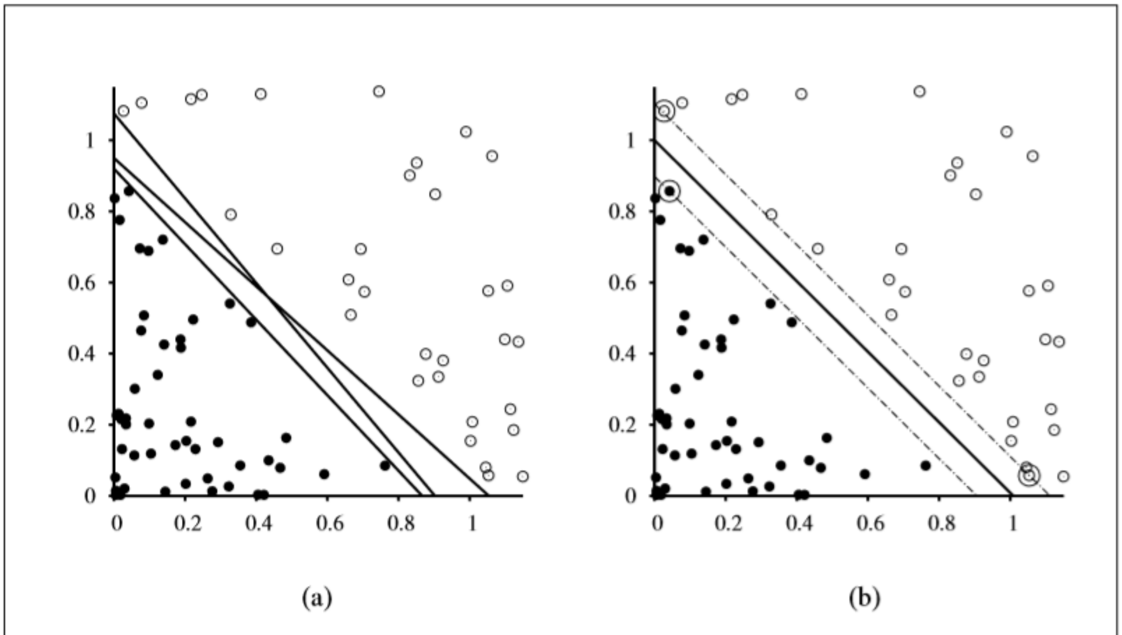


Figure 18.30 Support vector machine classification: (a) Two classes of points (black and white circles) and three candidate linear separators. (b) The maximum margin separator (heavy line), is at the midpoint of the **margin** (area between dashed lines). The **support vectors** (points with large circles) are the examples closest to the separator.

Figure 20 - Support Vector Machine Illustration [43]

Algebraically the sequential minimal optimization algorithm is a quadratic programming problem whose solution is found by solving for the vector α in Equation 17.

$$\underset{\alpha}{argmax} \sum_j \alpha_j - \frac{1}{2} \sum_{j,k} \alpha_j \alpha_k y_j y_k \left(x_j * x_k \right)$$

Equation 17 – Sequential Minimal Optimization Equation

Which is subject to the constraints of $\alpha_j \geq 0$, and $\sum_j \alpha_j y_j = 0$.

Typically SVM's perform better in binary classification when not much is known about the problem domain [43]. However, this data are not linearly separable. To overcome this, a technique called "kerneling" is used to create the maximal separating hyperplane.

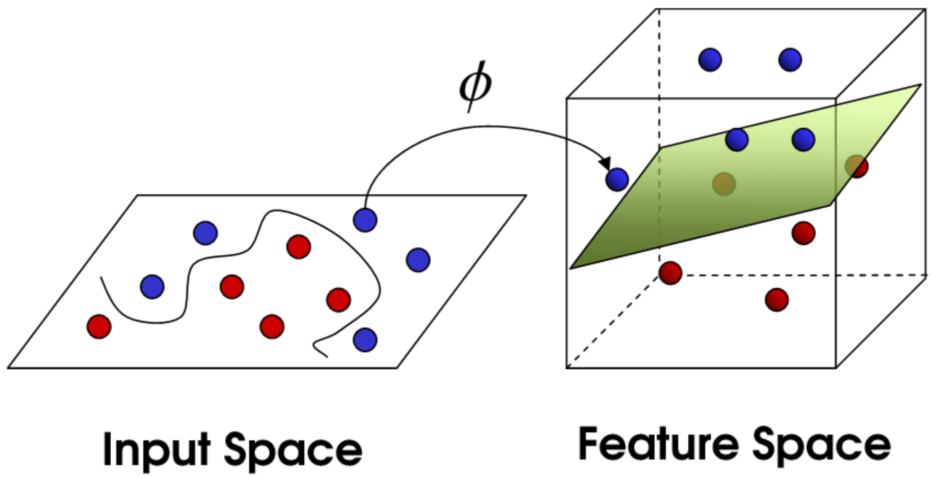


Figure 21 - Example of Kerneling Visual

Kerneling is achieved through the use of a similarity function that helps to separate instances of data in the input space and transform them into a linearly separable set in the feature space. By doing this, a maximal hyperplane distance can be calculated for non-linearly separable data. To handle multiple classes, the SVM in Weka uses a technique called "one-against-one." That is, for each class (sober, tipsy, and drunk) an SVM is trained for each pair of classes. For the dataset that it was trained on in the study, the SVM performed only slightly better than the Naïve Bayesian network. It had an accuracy of 47.1429%, with a weighted F-score of 0.427, and a ROC Area of 0.562. In essence, like the Naïve Bayes classifier, this model is only doing slightly better than chance. The confusion matrix of this classifier is shown in Table 11. This confusion matrix indicates that when the model is unsure, it will select class A (denoted "Sober") almost always.

SVM	**A ($<$ 3 Drinks)**	**B (3 $\leq$ Drinks $\leq$ 6)**	**C ($>$ 6 Drinks)**
A ($<$ 3 Drinks)	95	42	55
B (3 $\leq$ Drinks $\leq$ 6)	0	0	0
C ($>$ 6 Drinks)	12	3	2

Table 11 - SVM Confusion Matrix

The final classifier we considered was the J48 decision tree. The J48 algorithm is actually an open source implementation of the proprietary C4.5 classification algorithm [45]. The algorithm builds a decision tree from the training data that it is presented using the concept of information entropy. Information entropy is a quantitative measure of the amount of information contained in a single message (or single classified instance) [43]. Entropy is calculated by Equation 18.

$$H(V) = \sum_k P(v_k) \log_2 \frac{1}{P(v_k)}$$

Equation 18 – Entropy Calculation

Where:

v = is a random variable with possible values v_k, each with probability $P(v_k)$.

For each feature of gait, the decision tree builder considers the feature that maximizes the information gain at that step. Information gain is the difference between the entropy of the tree at level $i + 1$ and i. The decision tree seeks to minimize this value (thereby maximizing information gain) [43].

For the training data, the J48 tree performed second best of all of the classifiers. With an accuracy of 53.5714%, an F-score of 0.510, and an ROC Area of 0.646, this model is doing slightly better than random chance. The hope is that if this algorithm is implemented on the smartphone's application, the accuracy of classification improves. None of these methods were particularly effective due in part to the limited availability of training instances.

J48 Decision Tree	**A ($<$ 3 Drinks)**	**B ($3 \leq$ Drinks ≤ 6)**	**C ($>$ 6 Drinks)**
A ($<$ 3 Drinks)	73	32	13
B ($3 \leq$ Drinks ≤ 6)	24	5	11
C ($>$ 6 Drinks)	10	8	33

Table 12 - J48 Classifier Confusion Matrix

The confusion matrix in Table 12 indicates that, despite this being the most accurate method, the middle category (between 3 and 6 drinks) is not an often guessed category, and is not correct when it is. This would suggest that our classifier needs more data to be able to resolve the difference between categories better.

A summary table of compared classifiers is reproduced in Table 12.

Classifier	Accuracy %	Weighted F-Score	ROC Area
Naïve Bayes	42.1429	0.393	0.564
J48 Decision Tree	53.5714	0.510	0.646
SVM	47.1429	0.427	0.562
Random Forest	56.0000	0.629	0.658

Table 13 - Comparison of Statistical Classifiers for Data

As previously stated, the Random Forest decision tree performed best of all of the compared algorithms and will be the machine learner of choice to be implemented in the final application. After choosing this classifier, we looked to see whether techniques such pruning, bagging, and boosting would improve performance. These were experimented with in order to find an even more favorable result. This process is known as "ensembling." Bagging is a technique whereby the training data are sampled uniformly with replacement and trained. The output is the average of all classifications by the decision tree model being bagged. This technique is used to help make improvements to unstable dataset classification [43].

From all of the ensembling techniques considered, a bagged Random Forest of 10,000 trees created the best model. It had an accuracy of 56.00%, an F-score of 0.629, and AROC of 0.658. Its training data confusion matrix is shown in Table 14. This model was loaded onto the phone for use in the algorithm that will classify gait.

Ensemble Classifier	A (0-2 Drinks)	B (3-6 Drinks)	C (> 6 Drinks)
A (0-2 Drinks)	85	34	26
B (3-6 Drinks)	7	5	5
C (> 6 Drinks)	15	6	26

Table 14 – Random Forest Confusion Matrix

Since there is an overwhelming majority of data which is considered "sober," it makes sense that the classifier was able to accurately classify sober gait more than heavily intoxicated gait. It seems also that each classifier struggled to classify the 3-6 drink range. This could be due to both the lack of data for this bin and the fact that the feature response values are not well separated at the 3 and 6 drink mark.

After the classifier was selected, we verified its accuracy on a validation dataset previously unseen to the classifier. We used a sample of 30 instances to test the pre-trained model. It had an accuracy of 70.00%, an F-score of 0.786, and AROC of 0.825. Its validation data confusion matrix is shown in Table 15. This indicates the model is performing better than the cross-validated Random Forest.

Validation Set	A (0-2 Drinks)	B (3-6 Drinks)	C (> 6 Drinks)
A (0-2 Drinks)	1	0	9
B (3-6 Drinks)	0	6	0
C (> 6 Drinks)	0	0	14

Table 15 - Validation Set Confusion Matrix

The model performs well on predicting class C and appears to have a reduced number of errors on predicting class B when compared to initial training confusion matrix. This could be due to the different proportion of instances in each bin between the training and validation sets. The hope is that it will continue to improve its accuracy when more data on the phone is available.

3.4. Algorithm Design for Onboard Mobile Application

Ultimately, the goal of this research was to design and implement an application which leveraged the knowledge discovered above. It must be a lightweight, accurate, on-board gait classification engine. This section details how the knowledge gathered above was implemented into a smartphone application which can classify gait in real-time.

The vast majority of this algorithm was implemented as a "background service" in the application. A "background service" in Android is a type of application component that can perform long-running operations in the background and does not provide a user interface

[46]. Here, the background service waits continuously to receive updates from the Activity Recognition API from Android. The Activity Recognition API provides a wrapper for the `DetectedActivity` Android class. The `DetectedActivity` method `getType()` returns an integer representing the activity the phone has determined is occurring as well as a confidence level. A confidence level of 75% or better indicates that the result being returned is accurate [46].

Once walking has been detected, the raw accelerometer data is fed into the background service's main thread in 5 second increments. If walking terminates, or the confidence level falls below 75% during that 5 second interval, the data are discarded and the service waits for the next walking event. This was done by leveraging the GaitLib library's method `GaitAnalysis()` [47]. A snippet of this implementation can be seen in Figure 22.

```
public int onStartCommand(Intent intent, int flags, int startid) {
        ...
        //initialize service
        gaitLoggerServiceIntent = new Intent(this, GaitLoggerService.class);

        //check to see if the GaitLoggerService is already running
        if (GaitLoggerServiceStillRunning()) {
                setStateToLogging();
        }
        else {
                setStateToNotLogging();
        }

        startService(gaitLoggerServiceIntent);

        featureExtractionService = new Intent(AccelerometerService.this,
                                                FeatureExtractionService.class);
        featureExtractionService.putExtra(SELECTED_DATA_FOLDER, dataFolderName);
        featureExtractionService.putExtra(PARAM_WINDOW_SIZE, 2000);
        featureExtractionService.putExtra(PARAM_SAMPLING_INTERVAL, 1000);
        featureExtractionService.putExtra(PARAM_MIN_DATA_POINTS, 30);
        startService(featureExtractionService);

        GaitAnalysis mGaitAnalysis = new GaitAnalysis();
        mGaitAnalysis.startGaitAnalysis();

        return START_STICKY;
}
```

Figure 22 – Android Code snippet of AccelerometerStartService

A maximum of three 5 second samples are gathered per hour to prevent overcrowding of data and to manage storage space on the device. Once a 5 second sample is captured and checked for continuity between readings, it is fed into the feature generator.

The feature generator is a JAR file exported from MATLAB which contains all of the functions written to generate features from the previous experiment and study. The JAR is called with raw accelerometer data and returns the list of features calculated from them in order. If the JAR does not return values for some of the features, the data are discarded and the service waits for a new dataset. Once the features are generated from a set, the accelerometer readings are discarded, and the generated features are saved as a new row in a database table locally on the phone.

These features are labeled on the following day by the user using the in-app survey which included when they began drinking, when they finished drinking, and how many drinks they had. The data are labeled with the first sample inside the window being the baseline "0" drink mark. The number of drinks are spaced out over the interval of time spent drinking and samples are labeled accordingly. In addition to labeling the window during which drinking occurred, the application also labels the period after drinking occurs based on the average rate of alcohol metabolization of 1 drink per hour. After this labeling process occurs, the model is retrained using 10-fold cross validation on the entire data set to date.

A visual representation of the flow is as follows in Figure 23:

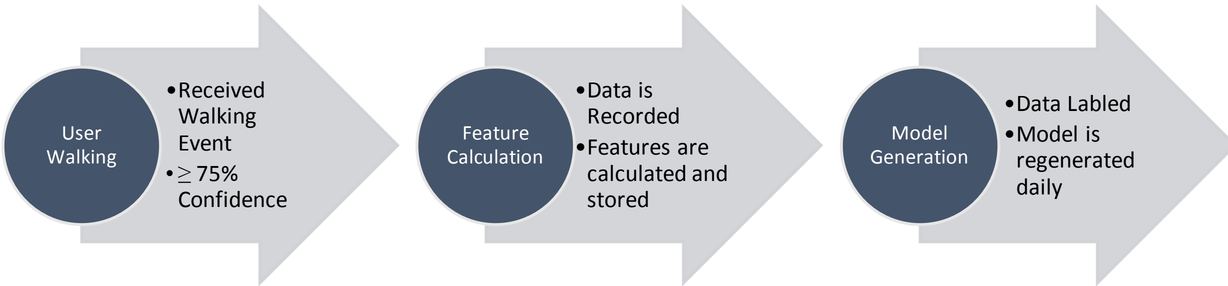


Figure 23 - Algorithm Model Flow

Once this training occurs, this model is loaded back into the application to be used to make inferences for the next 24 hours when it is updated again. In pseudocode this will look like:

```
program SmartphoneGaitInference:
//function gathers data
function gatherData(accelerometer)
	do while time < 5 seconds
		store accelerometerSignal
	end
	extractFeatures(accelerometerSignal)
end

function extractFeatures(signal)
	//Time domain
	getNumSteps(signal)
	getAvgStpLen(signal)
	getAvgStpTime(signal)
	getGaitVelocity(signal)
	getCadence(signal)
	getSkewness(signal)
	getKurtosis(signal)

	//Frequency domain
	freqSignal = fastFourierTransform(signal)
	getAvgPower(freqSignal)
	getEnergyRatio(freqSignal)
	getSigNoiseRatio(freqSignal)
	getTotalHarmonicDistortion(freqSignal)

	//store everything locally
	storeExtractedFeatures()
end

function classifyGait(storedFeatures)
	//use machine learner to learn a new model on
	//5000 random trees
	performRandomForest(storedFeatures,5000)
	storeModel()
end

function makeInference(accelerometer)
	signal = gatherData(accelerometer)
	model = getLatestModel()
	classification  = model.getInference(signal)
end

function updateModel(surveyData)
	for each getSamplesForLast24Hours() do
		if sample.time is contained in surveyData
			sample.drinkNum = surveyData.drinkNum
		end
```

3.5. Application Implementation

The overall goal of this project was to create an Android smartphone application to determine a user's intoxication level based on their gait. The application should be able to record the user's personal information including name, gender, weight, and age as well as determine whether the user is sober, slightly intoxicated, or heavily intoxicated.

3.5.1. Design

On the initial launch of the application, the user is brought to the startup screen. Here, the user can input their full name, gender, weight, and birthday. When the user clicks the Continue button, their information is saved and they are brought to the main screen. The user will never be brought back to the setup screen unless they reinstall the application.

On the main screen, the user can view the application's approximation of their intoxication level. Additionally, they may click the Profile button to view their profile. On the profile screen, the user can view their personal information. They then have the option of clicking the Back button or the Edit button. The Back button returns the user to the main screen. The Edit button brings the user to the edit profile screen. Here, the user can update their personal information including their name, gender, weight, and birthday. They then have the option of clicking the Cancel button or the Save button. The Cancel button returns the user to the profile screen without making any changes to their personal information. The Save button saves any changes that they have made to their information and returns the user to the profile screen with the updated information.

There is also a survey screen. Every day at noon, a notification is sent to the user to record their alcohol consumption from the previous twenty-four hours. When the user clicks on the notification, they are brought to the survey screen. Here, the user inputs the number of drinks they had, the time they began drinking, and the time they finished drinking. When they click the Submit button, they are then brought back to the main screen. Figure 24 shows the flow of screens for the application.

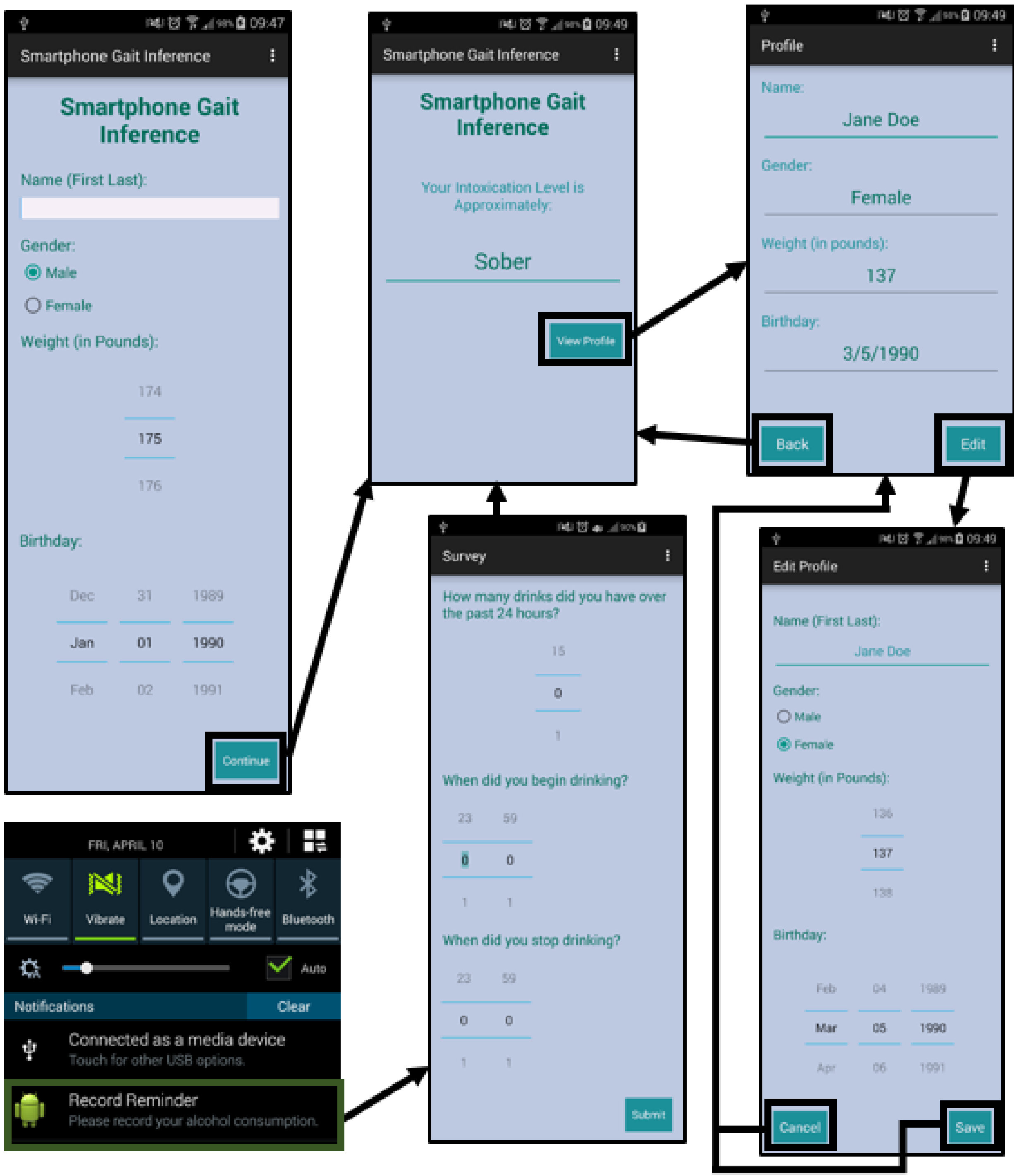


Figure 24 - Workflow of Screens

3.5.2. Application Architecture

Before implementation of the application could begin, we had to choose the platform for development. This could have been Apple iOS, Google Android, or both. However, based on the 2014 fourth quarter statistics, Android had approximately 77% of the market share worldwide versus the approximate 20% market share of iOS [48]. Additionally, development for Android is open source. Therefore, we chose Android as the platform for development. The Interactive Development Environment (IDE) then had to be selected. Eclipse or Android Studio could have been used. This application was created using Android Studio because it is the official IDE used for Android application development [49]. In order to manage code changes, we used Github as a central shared repository.

The core component of an Android application is an activity. An activity is a component of the application where the user can interact using a provided user interface, or screen. An activity exits in one of three states: Resumed, Paused, or Stopped. The functions used by the system to achieve these states can be seen in Figure 19. In order for this application to maintain the states, it primarily uses the `onCreate()`, `onResume()`, and `onPause()` methods. When an application is initially launched, it calls the `onCreate()` method which loads the user interface for the given screen. `onResume()` is called when the activity begins interacting with the user. `onPause()` is called when the system is about to resume a previous activity. Here, data changes are saved and items consuming CPU are stopped [50].

In general, there is one main activity that is loaded on startup of the application. When an application consists of multiple screens that the user can interact with, there are usually multiple activities which can create or start each other. When a new activity is started, the previous one is stopped. This causes only one activity to be visible to the user at a given time. However, the system records previous activities allowing the user to return to the previous activity at any time by simply pressing the back button. When this happens, the current activity is destroyed [50]. Figure 25 displays the lifecycle of an activity.

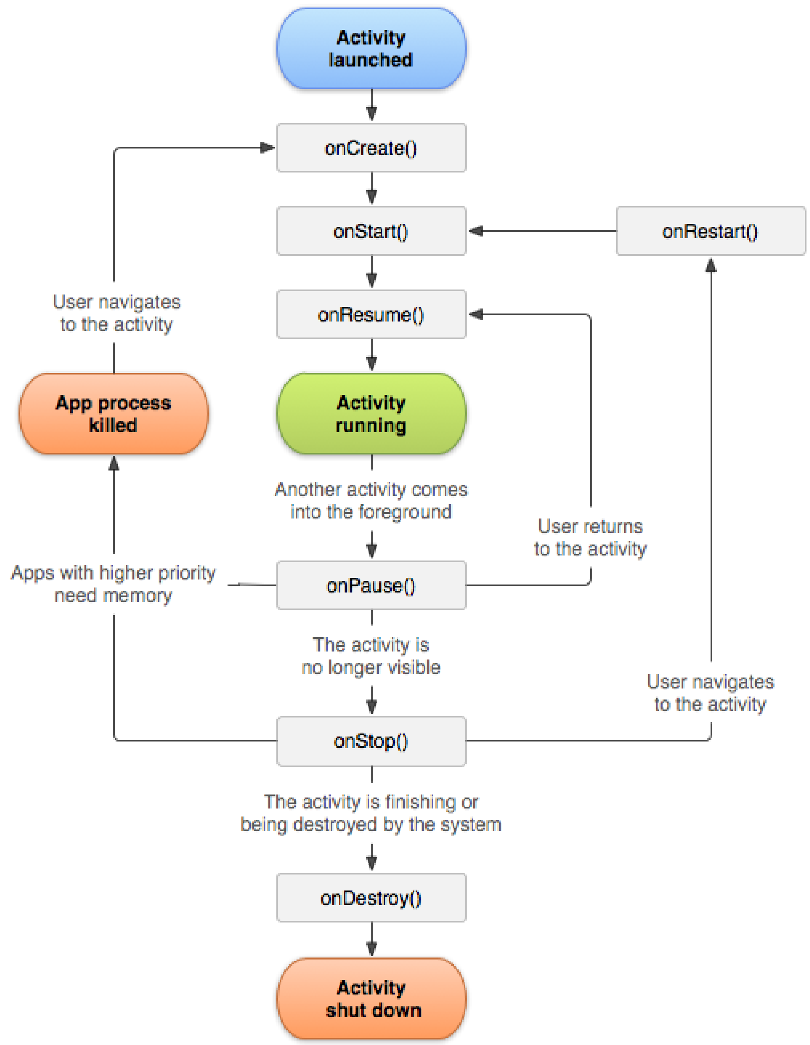


Figure 25 - Activity Lifecycle

The screens of the application were created using XML. Extensible Markup Language (XML) is a language used to encode text documents that is both human and machine readable [51]. A snippet of the XML and a corresponding application screen used for this project can be seen in Figure 26.

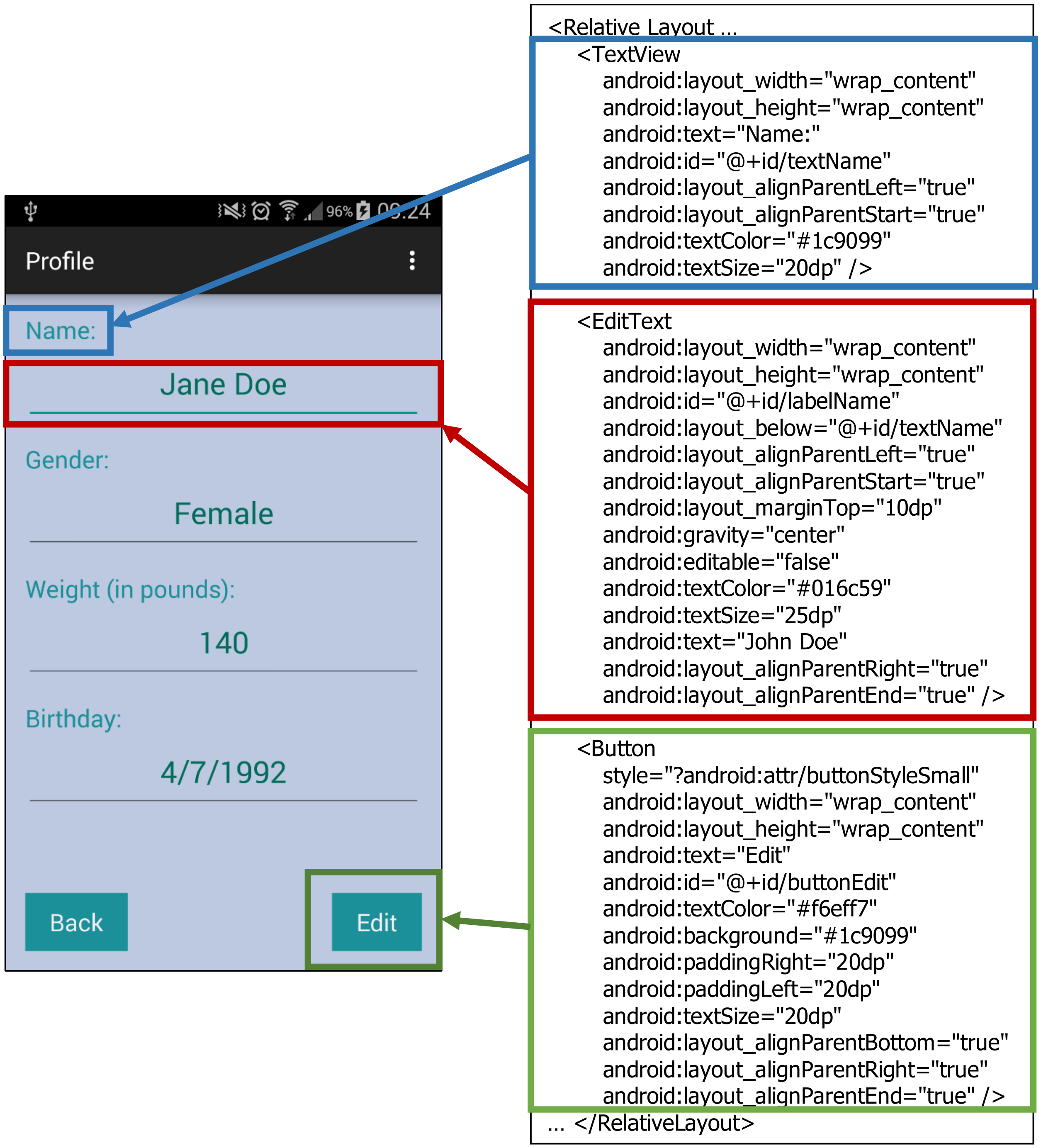


Figure 26 - Profile Screen with XML

While XML was used to create the user interface of the application, Java was used to code the functionality of the application. There were several different classes with different responsibilities which are outlined below.

The `MainScreen` class was responsible for starting the application on the startup screen on initial load, initializing the daily alarm and notification to take the survey, and setting the user's approximate intoxication level.

The `Startup` class was responsible for obtaining personal information about the user. This included the user's full name, gender, weight (in pounds), and birthday. When the user clicked the Continue button, all information was stored locally on the phone using `SharedPreferences` and a `PreferenceManager`.

The `Profile` class and the `EditProfile` class were responsible for displaying the personal information entered by the user. To display the information, the app retrieved the information from the `SharedPreferences`. Additionally, the `EditProfile` class gave the user the ability to edit and update their information. A snippet of this implementation to update the information can be seen in Figure 27.

```
name = labelName.getText().toString();
weight = numberWeight.getValue();
month = dateBirthday.getMonth();
day = dateBirthday.getDayOfMonth();
year = dateBirthday.getYear();

rGroup = (RadioGroup) findViewById(R.id.radioGender);
int selected = rGroup.getCheckedRadioButtonId();
RadioButton button = (RadioButton) findViewById(selected);
gender = (String) button.getText();

//put information into memory
mSharedPreferences = getSharedPreferences(PROFILE, MODE_PRIVATE);
SharedPreferences.Editor editor = mSharedPreferences.edit();

editor.putString(NAME, name);
editor.putString(GENDER, gender);
editor.putInt(WEIGHT, weight);
editor.putInt(BIRTHDAY, day);
editor.putInt(BIRTHMONTH, month);
editor.putInt(BIRTHYEAR, year);
editor.commit();
```

Figure 27 – Android Code snippet from EditSurvey to retrieve user input information and update the system

The `Survey` class was responsible for obtaining information on how many drinks the user had and over what time period. This information was again saved using the `SharedPreferences`. There was also a `SurveyAlertService` class which was responsible for sending the notification to the user's phone informing them to take the daily survey. This was implemented using a `TimerTask`. A snippet of this implementation can be seen in Figure 28.

```
int NOTIFY_RATE = 86400000;
calendar.set(Calendar.HOUR_OF_DAY, 12);
calendar.set(Calendar.MINUTE, 0);

timer.scheduleAtFixedRate(new TimerTask() {

        @Override
        public void run() {
          NotificationCompat.Builder notificationBuilder = new
                                   NotificationCompat.Builder(getApplicationContext())
                .setSmallIcon(R.drawable.ic_launcher)
                .setContentTitle("Record Reminder")
                .setContentText("Please record your alcohol consumption.");

          Intent resultIntent = new Intent(getApplicationContext(), Survey.class);

          TaskStackBuilder stackBuilder =
                      TaskStackBuilder.create(getApplicationContext());
          stackBuilder.addParentStack(MainScreen.class);
          stackBuilder.addNextIntent(resultIntent);

         PendingIntent resultPendingIntent = stackBuilder.getPendingIntent(0,
                                         PendingIntent.FLAG_UPDATE_CURRENT);

          notificationBuilder.setContentIntent(resultPendingIntent);

          NotificationManager notificationManager = (NotificationManager)
                                 getSystemService(Context.NOTIFICATION_SERVICE);

          Notification notification = notificationBuilder.build();
          notification.flags = Notification.FLAG_AUTO_CANCEL;

          notificationManager.notify(1, notification);

        }
}, calendar.getTime(), NOTIFY_RATE);
```

Figure 28 – Android Code snippet of SurveyAlertService to display alcohol consumption recording notification

3.5.3. Usability Study

After the application was completed, we wanted to determine how well the application functioned and inferred intoxication levels. We also wanted to determine how well the application worked for various users as well as gather the user impressions. To do so, we created a short usability study. This study consisted of distributing the application for volunteers to install on their phone for a few days and having them complete a short usability survey. The volunteers were students of the WPI computer science department as well as students of a WPI sorority. The only restriction for participation was that the volunteer must have been over 21 years of age. The survey consisted of a brief description of the intended future use of the application once it has been fully implemented and is consumer ready. It asked if they thought the goal of this future application was meaningful and useful to society as well as if there were any improvements they thought would be beneficial for the future application. Additionally, it asked about generic functionality of the application. This included if the application correctly transition between screens, if the users were notified daily at noon to complete the alcohol consumption survey, and if the application ever crashed. The full survey can be found in Appendix VI.

4. Results

The goal of this research was to produce a smartphone application for the Android operating system that is able to classify the presence of alcohol in human gait using only the phone's hardware accelerometers and a machine learning technique. In terms of this definition, the research was successful. An application was produced which was created using the methods described in the previous chapters.

4.1. Usability Study Results

The vision of the application was to help users monitor their alcohol intake using gait patterns and the accelerometer to make inferences. That user could then possibly make corrections to their drinking patterns if they deem it necessary. There were five males and two females that responded to the usability study. Approximately 86% of respondents found the vision being useful when the application is complete. One of the users responded that they "found it to be very insightful." Another found it to be "functional and easy to use." All of users said they would recommend this application to a friend but only 57% would use the application themselves. When users were asked to rate the classifier's accuracy on their intoxication levels on a scale of 1-10 (with 10 being 100% accurate), the mean value was 8.86 with a standard deviation of 1.86. Table 16 includes responses to primary questions and the number of individuals who responded yes.

Question	**Number that Responded 'Yes'**
Do you see the envisioned app being useful when complete?	6 (86%)
Would you recommend the envisioned app to others when it is complete?	7 (100%)
Would you actually used this envisioned application?	4 (57%)
Did you like the current application?	5 (71%)
Do you like the appearance of the application?	6 (86%)

Table 16 - Table of Responses

Next, the users were asked a series of questions to ensure that the application was relatively bug-free. The full report of this study can be found in Appendix VII. Revisions to the application were made before the final submission to account for some of the usability errors discovered during the study. They were negligible and will not be expressed in detail in this section.

There were some responses on possible improvements for the application. One user suggested there be a persistent notification that tells the user that the application is running and their intoxication level. Another user suggested to not display the start/stop drinking times when they select zero for the number of drinks.

4.2. Classification Improvement over Time

One of the most useful properties of supervised machine learning is the fact that classifiers tend to get more accurate when given more data. We speculated that after the initial loading of the model at application install time, classification would become more accurate as more data are available to train and test the model. The lower bound of classification is the 57% accuracy which comes with the application on installation. After only a few days' worth of use, the range of descriptive statistics retrieved from users in the usability study are expressed in Table 17.

User	Accuracy %	F-Score	ROC Area
User 1	58.435	0.597	0.612
User 2	52.784	0.513	0.538
User 3	62.947	0.640	0.631
User 4	67.249	0.651	0.683
User 5	73.854	0.754	0.749
User 6	33.333	0.410	0.551

Table 17 - Report of Usability Study Descriptive Statistics on Classifier Improvements

We also noticed that classification did improve for some but got worse for others. This result was interesting since we had initially thought that the lower bound was 57% for classification. This reduction in accuracy could be due to abnormal walking conditions or accidental triggers of the recording mechanism. For example, shaking the phone rapidly in a rhythmic motion for 5 seconds engages the step detector which signals our application to begin recording. Over time, we have seen some slight improvements and expect to see greater accuracy once more data is acquired.

5. Analysis

The application on the phone produced an average of 56% accuracy in classification. This result is impressive given that there is a large set of factors that influence gait along with alcohol. The result is not without flaw however. There are many ways in which this classification is not able to handle all edge cases in interacting with alcohol. For starters, the basic assumption which occurs during the survey phase, and subsequent model training, is that the reported drinking occurs in one sitting. For example, if in the last twenty-four hours a user has had four drinks and they were spaced out evenly during that time span, even though there was plenty of time between drinks for the body's liver and kidneys to filter the alcohol out of the bloodstream, the model will assert that the user is becoming progressively more drunk over this period.

In considering interactions with alcohol, there are a large number of possibilities which could explain how a model might be trained with mislabeled data. There is the complication of how each individual processes alcohol, whether or not food was consumed prior to or during drinking, and the type of alcohol consumed.

Specific to the classifiers, we noted that each technique was better at classifying the extremes (the 0-2 drink bin, and the > 6 drink bin) than classifying the 3-6 drink bin. We also noted that it was interesting that the validation set of data accuracy percentage (70%) was better than the overall model accuracy percentage (56 %). Over time, it seems that this result improved for some users of the application, and got worse for others. This could be due to gait conditions specific to the user, a mistake in the feature calculation, or an error in data collection. We noted that not all classifications improved over time, but after only a few days, most showed promise of improvements.

Additionally, a user’s walk may change over time due to any number of circumstances including weather, phone placement, ground conditions, personal injury, etc. The hope right now is to be able to gather enough data to cover the majority of cases and circumstances involving alcohol consumption and other situations impacting gait which occur in real life in order to accurately classify the unique effect of alcohol on gait.

6. Conclusion

Ultimately, there was some level of success gained in producing a smartphone application capable of classifying the presence of alcohol in gait in real-time. However, this research only began to address the entire scope of the problem. The following section details ways in which this application and research could be furthered for an even better result.

6.1. Future Work

Although this application successfully implemented basic intoxication level prediction, it could always be improved. First, the application could pull information from other sensors of the phone, not just the accelerometer data. This could include gyroscope, GPS, Bluetooth, and other movement sensors. This new information from the movement sensors could be used to generate more features and classifiers, potentially increasing the accuracy of the model. Using this updated model, the application might be able to more accurately determine the user's intoxication level between single drinks, rather than groups of drinks.

Additionally, cloud computing and storage could increase the effectiveness of the application and the model. The global model can be stored in the cloud while the more personal model stays local on the application. As the users label their data every day, the data is also uploaded to the global model in order to help general classification. Therefore, when a new user downloads the application, the model will be more accurate than the original. Likewise, personal classification can also be moved to the cloud. This could reduce the amount of CPU and storage necessary on the smartphone to calculate the user's intoxication level.

The application could also use a slight user interface enhancement. Currently, it is very simple and repetitive from screen to screen. More diversity amongst screens could increase usability for the user. A new screen for displaying a log of the user's drinking activities could also be implemented. Additionally, removing the need for a daily survey would benefit the user. They could just have the application running on their smartphone and not be bothered daily to report on their alcohol consumption from the previous day. However, this could only be done once the model is more accurate. Once the accuracy of the intoxication level prediction is increased, the application could be able to predict the user's BAC.

Furthermore, this application could eventually be integrated into the healthcare system. This application could be used for recovering alcoholics who are in therapy. The user would install the application onto their smartphone and continue about their daily life. The application could record GPS data to determine where the user is when they are drinking, the time of day the user is drinking, and Bluetooth data to determine if the user is around other people when they are drinking. This information could then be stored on the cloud to be used at a later date by their therapist. They could then review the data and determine if there are any correlations between the time of day, location, or people the user is with when they are drinking. The therapist could then make more accurate plans on how to assist the recovering alcoholic.

Moreover, social networking could be incorporated into the application. It could be used to find other people near the user who are using the application. This would enable the user to create a virtual support network. The users would be able to track and discuss their progress in recovering from alcohol abuse with each other and help keep each other free from alcohol.

References

[1] Coalition against Drug Abuse, "History of Drug Abuse," 2014. [Online]. Available: http://drugabuse.com/library/history-of-drug-abuse/. [Accessed 14 October 2014].

[2] World Health Organization, "Global status report on alcohol and health," World Health Organization, 2014. [Online]. Available: http://www.who.int/substance_abuse/publications/global_alcohol_report/msb_gsr_2014_1.pdf?ua=1. [Accessed 22 September 2014].

[3] K.-C. Wang, Y.-H. Hsieh, C.-H. Yen, C.-W. You, M.-C. Huang, C.-H. Lee, S.-Y. Lau, H.-L. Kao, H.-H. Chu and M.-S. Chen, "SoberDiary: A Phone-based Support System for Assisting Recovery from Alcohol Dependence," ACM, Seattle, WA, 2013.

[4] J. Rehm, T. K. Greenfield and J. D. Rogers, "Average Volume of Alcohol Consumption, Patterns of Drinking, and All-Cause Mortality: Results from the US National Alcohol Survey," *American Journal of Epidemiology,* vol. 153, no. 1, pp. 64-71, 2001.

[5] J. Westmaas, S. Moeller and P. B. Woicik, "Validation of a Measure of College Students' Intoxicated Behaviors: Associations With Alcohol Outcome Expectancies, Drinking Motives, and Personality," *Journal of American College Health,* vol. 55, no. 4, pp. 227-237, 2007.

[6] D. K. L. Seigers and K. B. Carey, "Screening and Brief Interventions for Alcohol Use in College Health Centers: A Review," *Journal of American College Health,* vol. 59, no. 3, pp. 151-158, 2010.

[7] S. Demura and M. Uchiyama, "Influence of moderate alcohol ingestion on gait," *Sport Sci Health,* no. 4, pp. 21-26, 2008.

[8] S. Ando, T. Iwata, H. Ishikawa, M. Dakeishi and K. Murata, "Effects of acute alcohol ingestion on neuromotor functions," *NeuroToxicology,* vol. 29, pp. 735-739, 2008.

[9] H. Lu, J. Huang, T. Saha and L. Nachman, "Unobtrusive Gait Verification for Mobile Phones," in *ISWC,* Seattle, WA, USA, 2014.

[10] Centers for Disease Control and Prevention, "Alcohol and Public Health," Centers for Disease Control and Prevention, 19 August 2014. [Online]. Available: http://www.cdc.gov/alcohol/fact-sheets/alcohol-use.htm. [Accessed 22 September 2014].

[11] M. Izzi, "SCRAM Bracelet Laws," LegalMatch, 3 September 2014. [Online]. Available: http://www.legalmatch.com/law-library/article/scram-bracelet-laws.html. [Accessed 16 12 2014].

[12] SCRAM Systems, "SCRAM Continuous Alcohol Monitoring," Alcohol Monitoring Systems, Inc., 2014. [Online]. Available: http://www.scramsystems.com/index/scram/continuous-alcohol-monitoring. [Accessed 09 December 2014].

[13] Tokyoflash Japan, "Kisai Intoxicated LCD Watch," Tokyoflash Japan, 2014. [Online]. Available: http://www.tokyoflash.com/en/watches/kisai/intoxicated/. [Accessed 09 December 2014].

[14] E. R. Weaver, D. R. Horyniak, R. Jenkinson, P. Dietze and M. S. Lim, ""Let's get Wasted!" and Other Apps: Characteristics, Acceptability, and Use of Alcohol-Related Smartphone Applications," *JMIR MHEALTH AND UHEALTH,* vol. 1, no. 1, pp. 1-11, 2013.

[15] M. Nieschalk, C. Ortmann, A. West, F. Schmäl, W. Stoll and G. Fechner, "Effects of Alcohol on body-sway patterns in human subjects," *International Journal of Legal Medicine,* vol. 112, pp. 253-60, 1999.

[16] S. Nishiguchi, M. Yamada, K. Nagai, S. Mori, Y. Kajiwara, T. Sonoda, K. Yoshimura, H. Yoshitomi, H. Ito, K. Okamoto, T. Ito, S. Muto, T. Ishihara and T. Aoyama, "Reliability and Validity of Gait Analysis by Android-Based Smartphone," *Telemedicine and e-Health,* vol. 18, no. 4, pp. 292-296, 2012.

[17] Y.-S. Lee and S.-B. Cho, "Activity recognition with android phone using mixture-of-experts co-trained with labeled and unlabeled data," *Neurocomputing,* no. 126, pp. 106-115, 2013.

[18] H.-L. Kao, B.-J. Ho, A. C. Lin and H.-H. Chu, "Phone-based Gait Analysis to Detect Alcohol Usage," ACM, Pittsburgh, USA, 2012.

[19] P. Miller, "NDLERF," October 2013. [Online]. Available: http://www.ndlerf.gov.au/publications/monographs/monograph-46. [Accessed 12 November 2014].

[20] K. R. Hirsch, "The 19 Best Alcoholism iPhone & Android Apps of 2014," Healthline Networks, 21 May 2014. [Online]. Available: http://www.healthline.com/health-slideshow/top-alcoholism-iphone-android-apps#1. [Accessed 4 November 2014].

[21] N. Pell, "The 9 Best Alcohol-Related Apps," Made Man, 30 December 2011. [Online]. Available: http://www.mademan.com/the-9-best-alcohol-related-apps/. [Accessed 4 November 2014].

[22] "Sports Injury Clinic," 2014. [Online]. Available: http://www.sportsinjuryclinic.net/sports-specific/running-injuries/gait-analysis. [Accessed 20 March 2014].

[23] Google, "Motion Sensors," Google, [Online]. Available: http://developer.android.com/guide/topics/sensors/sensors_motion.html. [Accessed 3 December 2014].

[24] V. Tuzlukov, Signal Processing Noise, Electrical Engineering and Applied Signal Processing Series, CRC Press, 2010.

[25] Mathworks, "Fourier Series," [Online]. Available: http://www.mathworks.com/help/curvefit/fourier.html. [Accessed October 2014].

[26] G. Middleton, "Pitch Detection Algorithms," Rice University, December 17 2003. [Online]. Available: http://cnx.org/contents/8b900091-908f-42ad-b93d-806415434b46@2/Pitch_Detection_Algorithms. [Accessed 16 April 2015].

[27] N. R. Draper and H. Smith, Applied regression analysis, New York: Wiley, 1998.

[28] MathWorks, "Evaluating Goodness of Fit," MathWorks, [Online]. Available: http://www.mathworks.com/help/curvefit/evaluating-goodness-of-fit.html. [Accessed 18 November 2014].

[29] R. H. Shumway and D. S. Stoffer, Time Series Analysis and Its Applications, New York City: Springer, 2011.

[30] E. Sejdić, K. A. Lowry, . J. Bellanca, M. S. Redfern and J. S. Brach, "A comprehensive assessment of gait accelerometry signals in time, frequency and time-frequency domains.," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 22, no. 3, pp. 603-612, 2014.

[31] Mathworks, "Counting Steps by Capturing Acceleration Data from Your Android™ Device," Mathworks, 1 March 2015. [Online]. Available: http://www.mathworks.com/help/supportpkg/mobilesensor/examples/counting-steps-by-capturing-acceleration-data-from-your-android-device.html?prodcode=ML. [Accessed 22 March 2015].

[32] NIST/SEMATECH, "Measures of Skewness and Kurtosis: NIST/SEMATECH e-Handbook of Statistical Methods," 1 April 2012. [Online]. Available: http://www.itl.nist.gov/div898/handbook/eda/section3/eda35b.htm. [Accessed 22 March 2015].

[33] E. W. Weisstein, "Power Spectrum," MathWorld--A Wolfram Web Resource, 2 March 2015. [Online]. Available: http://mathworld.wolfram.com/PowerSpectrum.html. [Accessed 23 March 2015].

[34] Mathworks, "pwelch - Welch's power spectral density estimate," Mathworks, 1 March 2015. [Online]. Available: http://www.mathworks.com/help/signal/ref/pwelch.html. [Accessed 23 March 2015].

[35] E. W. Weisstein, "Average Power," MathWorld -- A Wolfram Web Resource, 2 March 2015. [Online]. Available: http://mathworld.wolfram.com/AveragePower.html. [Accessed 23 March 2015].

[36] Mathworks, "Signal To Noise Ratio," Mathworks, 1 March 2015. [Online]. Available: http://www.mathworks.com/help/signal/ref/snr.html. [Accessed 23 March 2015].

[37] Mathworks, "Total Harmonic Distortion," Mathworks, 1 March 2015. [Online]. Available: http://www.mathworks.com/help/signal/ref/thd.html. [Accessed 23 March 2015].

[38] I. V. Blagouchine, "2478IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 59, NO. 9, SEPTEMBER 2011Analytic Method for the Computation of theTotal Harmonic Distortion by theCauchy Method of Residues," *IEEE TRANSACTIONS ON COMMUNICATION,* vol. 59, no. 9, pp. 2478-2491, 2011.

[39] T. Kitabayashi, S. Demura, M. Noda and T. Yamada, "Gender Differences in Body Sway Factors of Center of Foot Pressure in a Static Upright Posture and under the Influence of Alcohol Intake," *Journal of Physiological Anthropology and Applied Human Science,* pp. 111-118, 2004.

[40] funf, "Funf Open Sensing Framework," Behavio, [Online]. Available: http://www.funf.org/. [Accessed 24 March 2015].

[41] F. K. Del Boca and J. Darkes, "The validity of self-reports of alcohol consumption: state of the science and challenges for research," in *Self-Report and Biochemical Markers of Alcohol Consumption,* Bethesda, MD, 2002.

[42] K. Ron, "Glossary of Terms," Stanford University, 1998. [Online]. Available: http://ai.stanford.edu/~ronnyk/glossary.html. [Accessed 29 October 2014].

[43] S. J. Russell and P. Norvig, Artificial Intelligence: A Modern Approach, Upper Saddle River: Prentice Hall/Pearson, 2023.

[44] S. P. Awate, "Maximum-a-Posteriori (MAP) Estimation," University of Utah, 21 February 2007. [Online]. Available: https://www.cs.utah.edu/~suyash/Dissertation_html/node8.html. [Accessed 25 March 2015].

[45] J. R. Quinlan, C4.5: Programs for Machine Learning, Ann Arbor: Morgan Kaufmann Publishers, 1993.

[46] Google, "Android Developers Reference," 19 March 2015. [Online]. Available: http://developer.android.com/reference/packages.html. [Accessed 31 March 2015].

[47] M. M. A. Wu, O. S. Schneider, I. Karuei, L. A. Leong and K. E. MacLean, "Introducing GaitLib: A Library for Real-time Gait Analysis in Smartphones," 2014. [Online]. Available: https://circle.ubc.ca/bitstream/handle/2429/46848/gaitlib-paper.pdf?sequence=1. [Accessed 11 November 2014].

[48] IDC, "Smartphone OS Market Share, Q4 2014," IDC, 2015. [Online]. Available: http://www.idc.com/prodserv/smartphone-os-market-share.jsp. [Accessed 30 March 2015].

[49] "Android Studio Overview," Android, [Online]. Available: http://developer.android.com/tools/studio/index.html. [Accessed 23 March 2015].

[50] Google, "Activity," [Online]. Available: http://developer.android.com/reference/android/app/Activity.html. [Accessed 30 March 2015].

[51] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)," 7 February 2013. [Online]. Available: http://www.w3.org/TR/xml/. [Accessed 30 March 2015].

Appendices

I. Subject Recruiting Email

Subject: Smartphones and Alcohol

Body:

Do you drink alcohol, and are you of legal drinking age?

Are you willing to participate in a survey regarding drinking habits?

Do you own a smartphone?

Would you be curious to see if your smartphone could accurately detect how much you've had to drink by how you walk?

Some of our survey participants will be invited to a second study on alcohol consumption and its effect on human gait. We hope to model this relationship using smartphones as a data gathering and analysis device. We plan to use this research to improve just-in-time responses of the phone to its environment, especially when alcohol is present. The results of this study will possibly aid in the development of an application that will be able to passively monitor alcohol consumption.

If you are interested in joining this study or would like more information, please contact either Danielle (dmlarose@wpi.edu) or Zach (zparnold@wpi.edu).

As a note: alcohol will NOT be provided to participants for the purpose of this study.

Thanks and take care!

-Zach Arnold and Danielle LaRose (on behalf of the Smartphone Gait Inference team.)

II. Survey Consent Form

The University of
Science and Technology.
And Life.

Informed Consent Agreement for Participation in a Research Study

Investigators: Zachary Arnold, Danielle LaRose
Contact Information: smartphonegait@wpi.edu
Title of Research Study: Smartphones and Alcohol Consumption Survey

Introduction:
You are being asked to participate in a research study. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks or discomfort that you may experience as a result of your participation. This form presents information about the study so that you may make a fully informed decision regarding your participation.

Purpose of the study:
In this survey we are gathering information on subjects' smartphone use and alcohol consumption. Some of the participants in this survey will be invited to participate in a second study to measure the effect of alcohol on walking gait using a smartphone. No alcohol will be provided to participants in either study, however.

Procedures to be followed:
You will be asked to complete a short survey on your smartphone use and alcohol consumption.

Risks to study participants:
There is a small risk that sensitive information that you disclose on the survey could be disclosed to others. However, at any time you may choose to have your data removed from the survey database and it will be permanently destroyed.

Benefits to research participants and others:
There are no direct benefits to the participants of the study. Some participants will be invited to participate in a second study about alcohol consumption and gait, using a smartphone app. If this second study is successful, the overall results of this experiment will provide more accurate information to future users on their intoxication levels without the hassle of reporting their activities.

Record keeping and confidentiality:
Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data that identifies you by name. Any publication or presentation of the data will not identify you.

Compensation or treatment in the event of injury:
Since there are no direct risks associated with this research, there will be no compensation provided in the event of injury.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact:

WPI Institutional Review Board Chair:
Professor Kent Rissmiller, Tel. 508-831-5019, Email:
kjr@wpi.edu WPI's University Compliance Officer:
Michael J. Curley, Tel. 508-831-6919, Email:
mjcurley@wpi.edu Primary Investigator:
Professor Emmanuel Agu, Email:
emmanuel@cs.wpi.edu Co-Investigator:
Professor Joseph Petruccelli, Email:
jdp@wpi.edu Student Investigators:
Zachary Arnold, Email:
zparnold@wpi.edu Danielle LaRose,
Email: dmlarose@wpi.edu

Your participation in this research is voluntary. Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. If you choose to do so, you will be given the option to erase all previous data and to have it not used in the study. There will not be any repercussions from the university, including grades or academic standing. The project investigators retain the right to cancel, postpone, or extend the experimental procedures at any time they see fit.

By signing below, you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

______________________________ Date: ______________________

Study Participant Signature

APPROVED
WPI IRB 1
1/21/15 to1/20/16

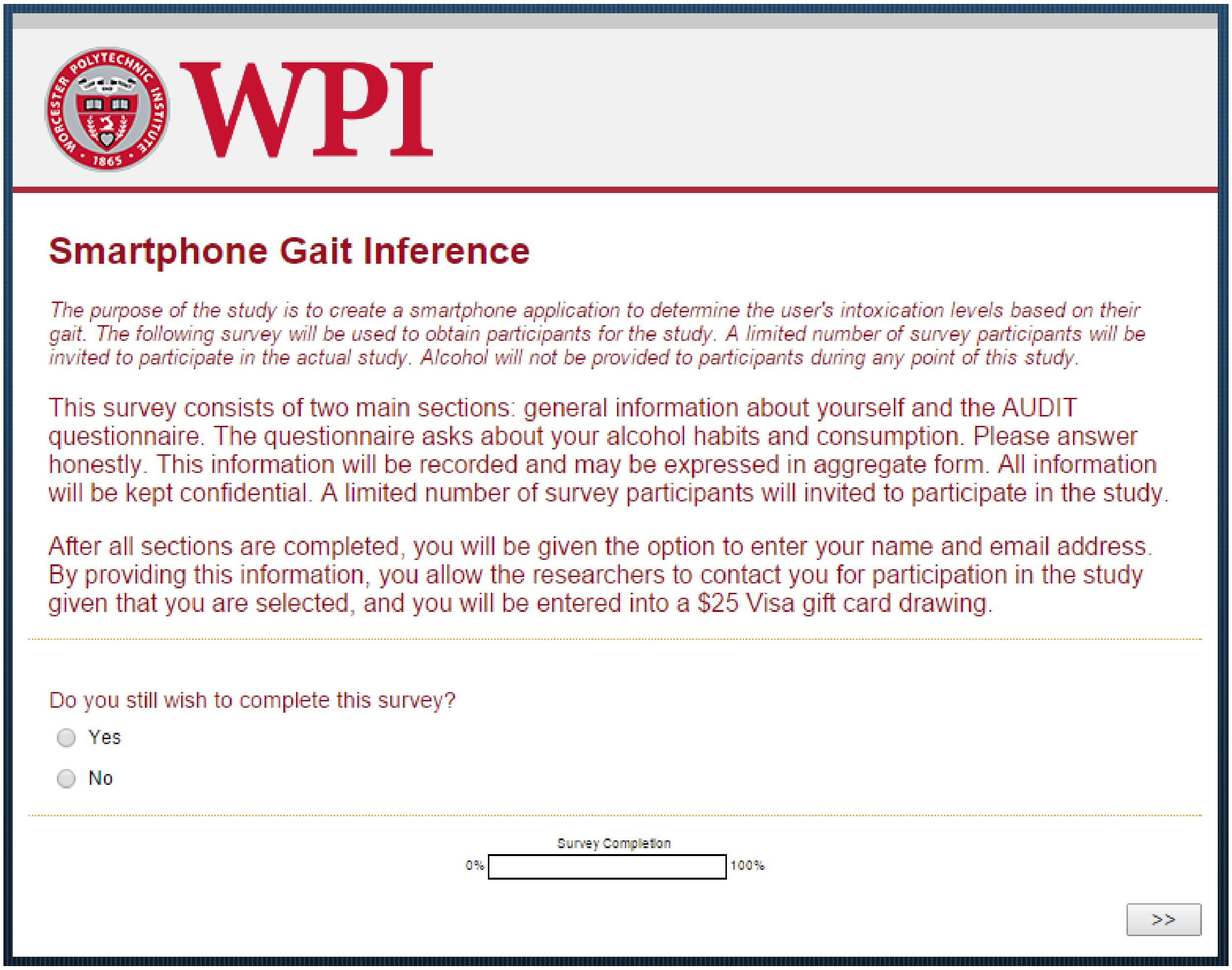
____

Study Participant Name (Please print)

______________________________ Date: ______________________

Signature of Person who explained this study

III. Subject Screening Pre-Questionnaire



WPI

Smartphone Gait Inference

The purpose of the study is to create a smartphone application to determine the user's intoxication levels based on their gait. The following survey will be used to obtain participants for the study. A limited number of survey participants will be invited to participate in the actual study. Alcohol will not be provided to participants during any point of this study.

This survey consists of two main sections: general information about yourself and the AUDIT questionnaire. The questionnaire asks about your alcohol habits and consumption. Please answer honestly. This information will be recorded and may be expressed in aggregate form. All information will be kept confidential. A limited number of survey participants will invited to participate in the study.

After all sections are completed, you will be given the option to enter your name and email address. By providing this information, you allow the researchers to contact you for participation in the study given that you are selected, and you will be entered into a $25 Visa gift card drawing.

Do you still wish to complete this survey?

- Yes
- No

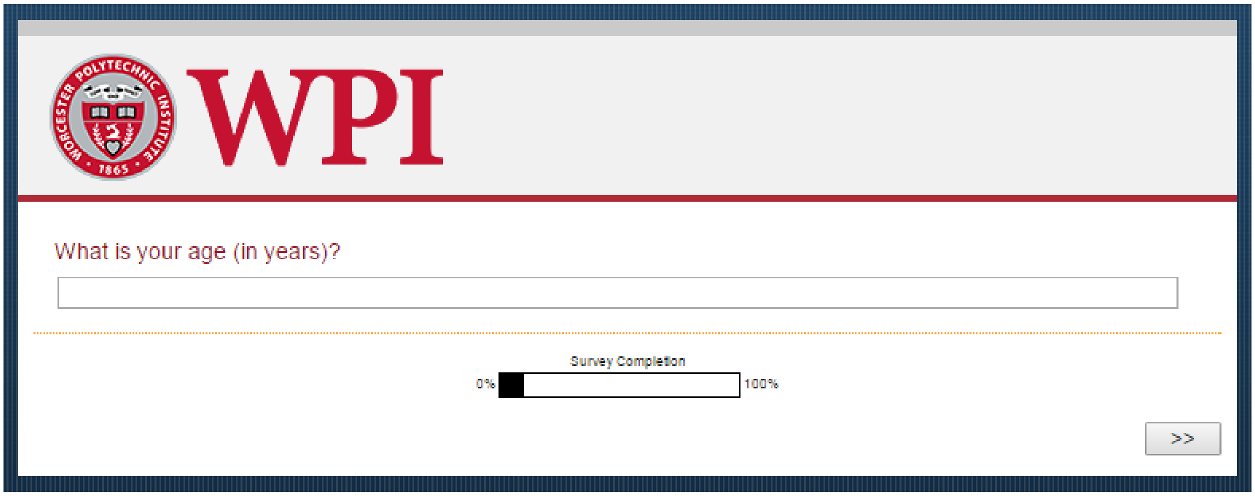
Survey Completion

0% 100%

>>

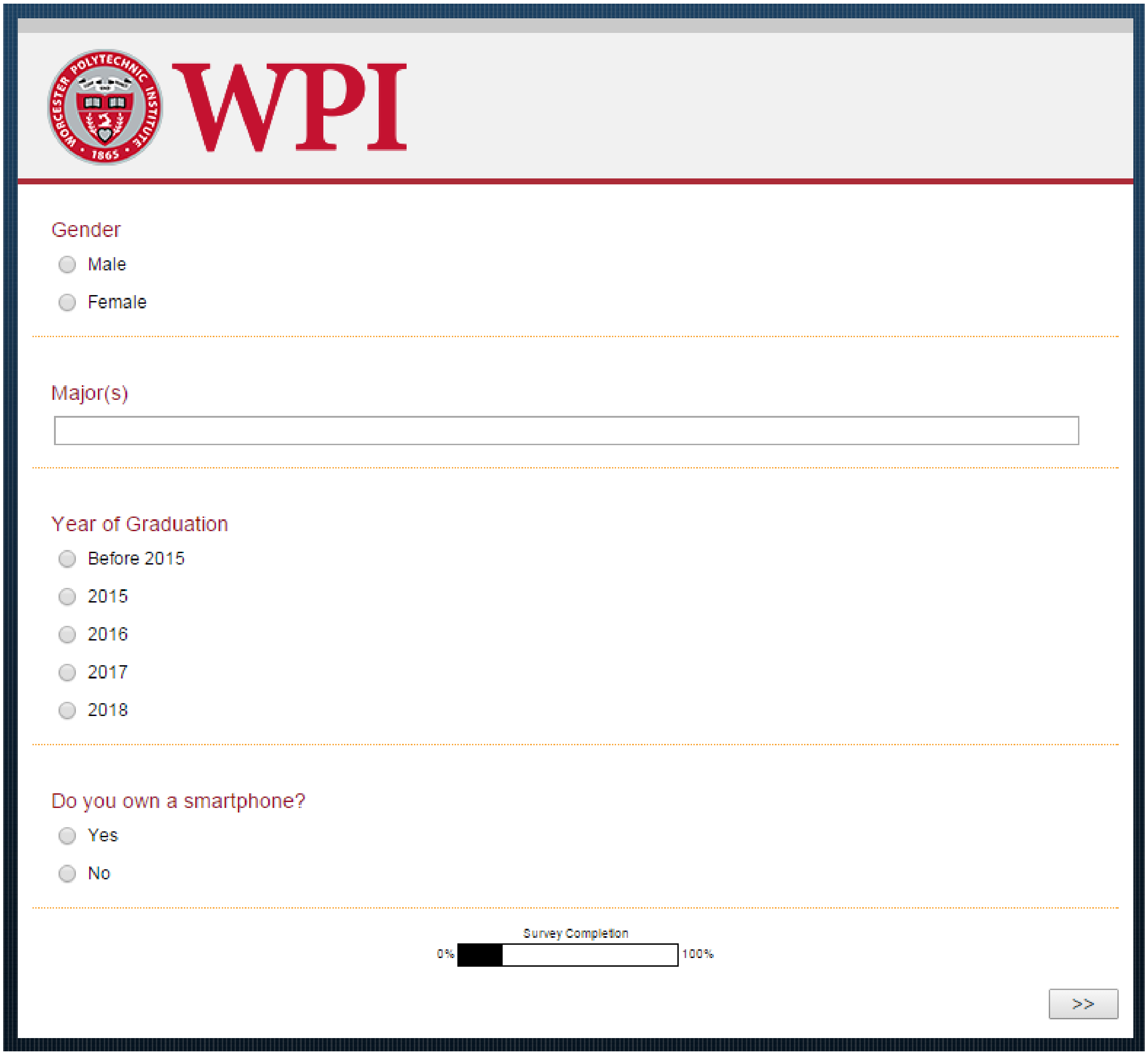
If the answer is:

- "No", then go to the Submit Survey page
- "Yes", then continue



If the response is:

- Less than “21”, then go to Gift Card page
- Greater than or equal to “21”, then continue


WPI

Gender

Male

Female

Major(s)

Year of Graduation

Before 2015

2015

2016

2017

2018

Do you own a smartphone?

Yes

No

Survey Completion

0% 100%

>>

If "Do you own a smartphone?" is:

- "No", then go to Gift Card page
- "Yes", then continue

WPI

Which smartphone do you own?

- Apple
- Android

How comfortable are you using your smartphone?

- Very Comfortable
- Somewhat Comfortable
- Neither Comfortable nor Uncomfortable
- Somewhat Uncomfortable
- Very Uncomfortable

How comfortable are you using computers or computing technology?

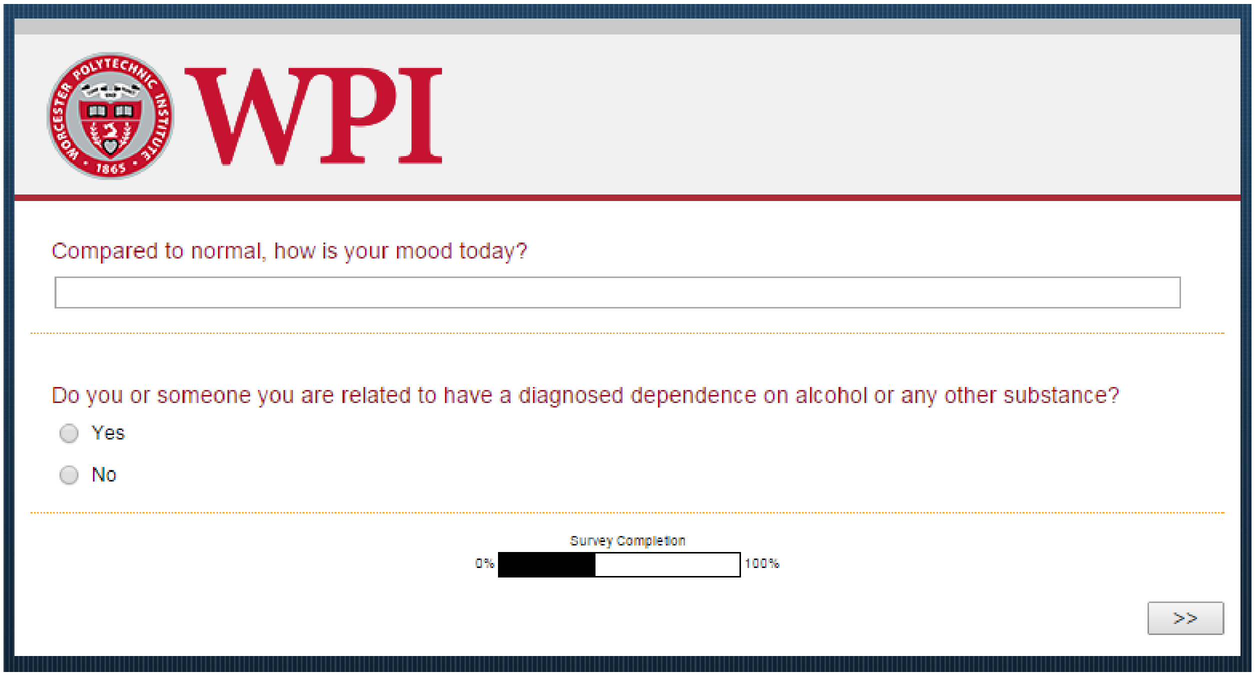
- Very Comfortable
- Somewhat Comfortable
- Neither Comfortable nor Uncomfortable
- Somewhat Uncomfortable
- Very Uncomfortable

Survey Completion

0% 100%

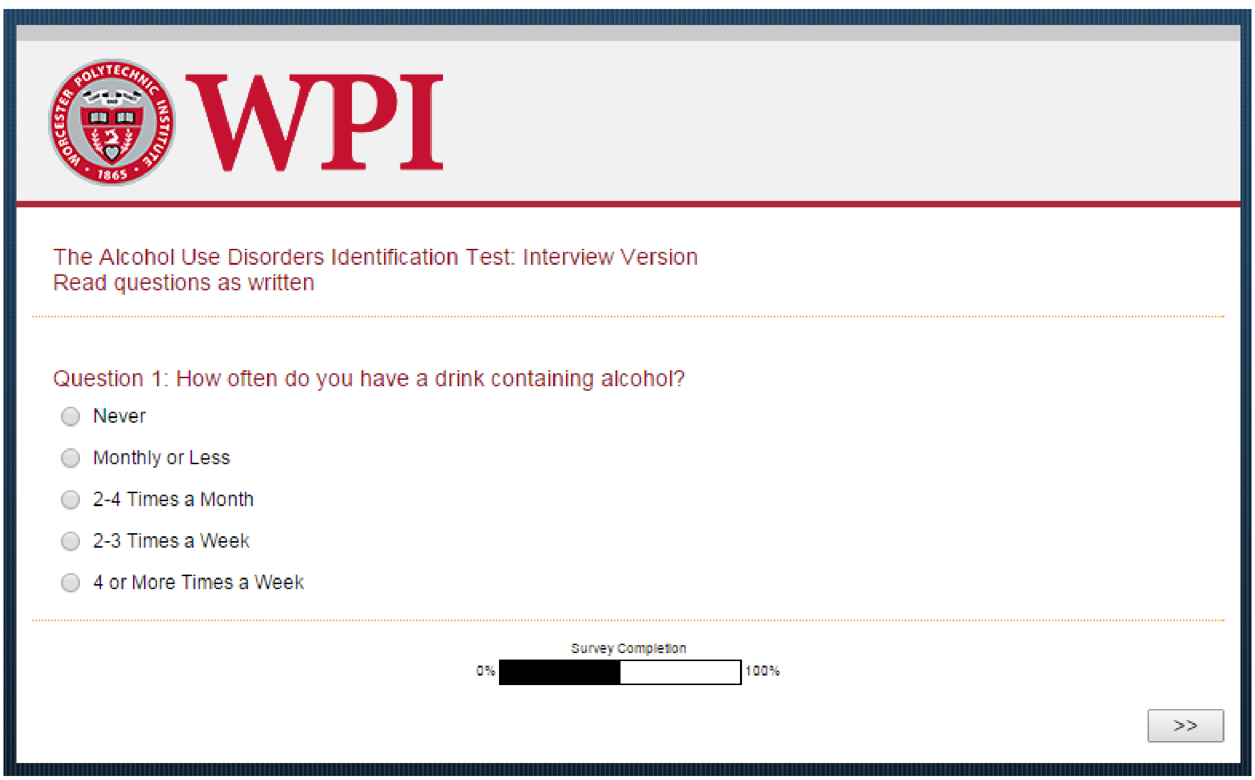
>>

Continue



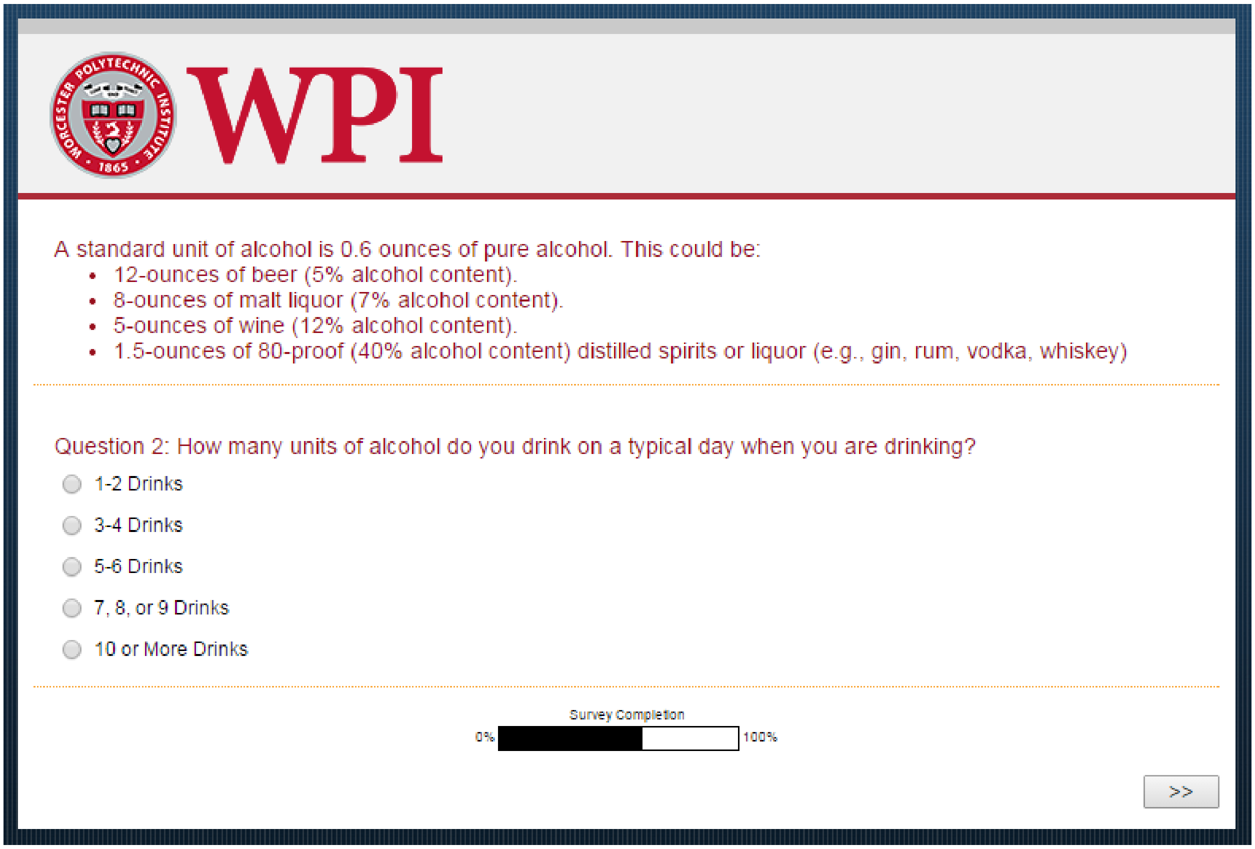
If the answer to “Do you or someone you are related to…” is:

- “Yes”, then go to Gift Card page
- “No”, then continue



If the answer to Question 1 is:

- “Never”, then go to Question 9 and 10 page
- Another response, then continue



If the answer to:

- Question 2 is "1-2 Drinks" and Question 3 is "Never", then go to Question 9 and 10 page
- Another response, then continue

Question 4: How often during the last year have you found that you were not able to stop drinking once you had started?

- Never
- Less than Monthly
- Monthly
- Weekly
- Daily or Almost Daily

Question 5: How often during the last year have you failed to do what was normally expected from you because of drinking?

- Never
- Less than Monthly
- Monthly
- Weekly
- Daily or Almost Daily

Question 6: How often during the last year have you needed an alcoholic drink in the morning to get yourself going after a heavy drinking session?

- Never
- Less than Monthly
- Monthly
- Weekly
- Daily or Almost Daily

Question 7: How often during the last year have you had a feeling of guilt or remorse after drinking?

- Never
- Less than Monthly
- Monthly
- Weekly
- Daily or Almost Daily

Question 8: How often during the last year have you been unable to remember what happened the night before because you had been drinking?

- Never
- Less than Monthly
- Monthly
- Weekly
- Daily or Almost Daily

Question 9: Have you or someone else been injured as a result of your drinking?

- No, Never
- Yes, But Not in the Last Year
- Yes, During the Last Year

Question 10: Has a relative or friend or a doctor or another health worker been concerned about your drinking or suggested you cut down?

- No, Never
- Yes, But Not in the Last Year
- Yes, During the Last Year

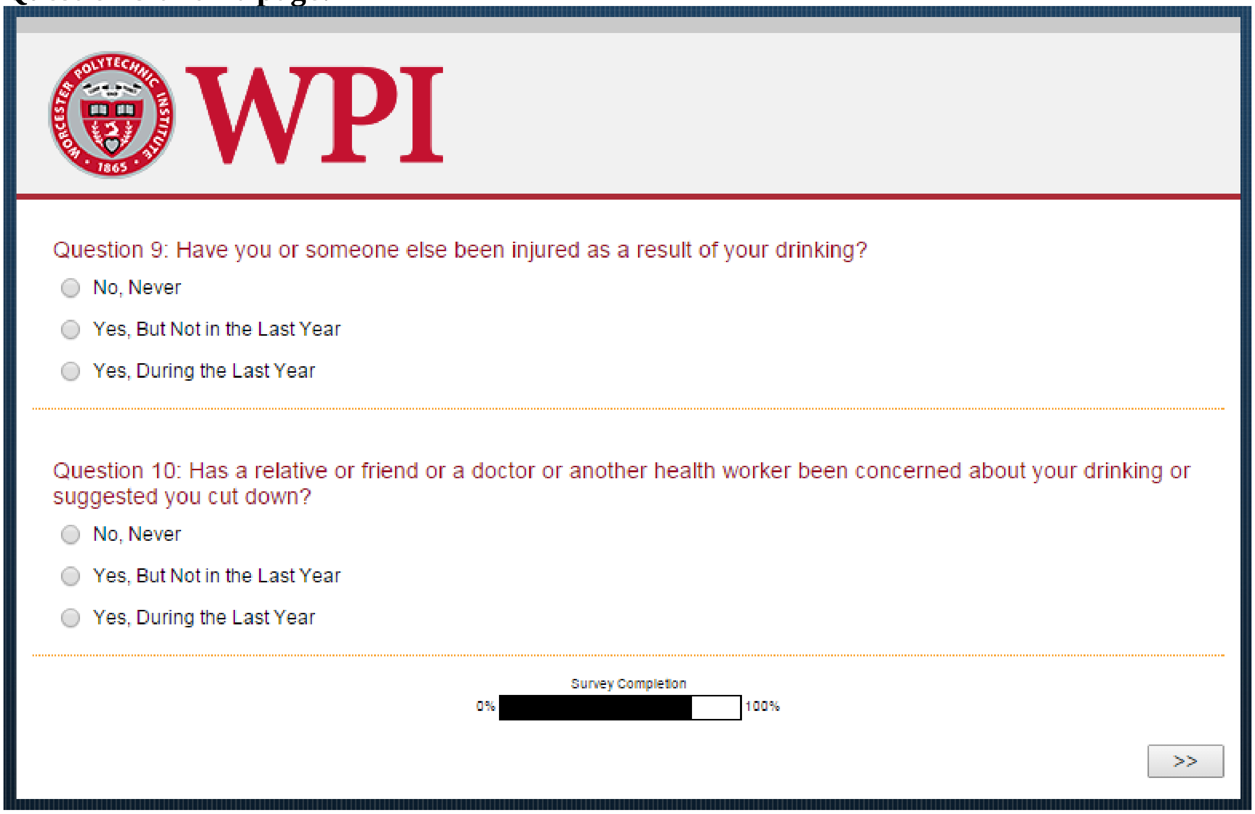
Survey Completion

0% 100%

>>

Continue to Gift Card page

Question 9 and 10 page:



WPI

Question 9: Have you or someone else been injured as a result of your drinking?

- No, Never
- Yes, But Not in the Last Year
- Yes, During the Last Year

Question 10: Has a relative or friend or a doctor or another health worker been concerned about your drinking or suggested you cut down?

- No, Never
- Yes, But Not in the Last Year
- Yes, During the Last Year

Survey Completion

0% 100%

>>

Gift Card page:

WPI

If you would like to be entered into a $25 Visa gift card drawing and/or be contacted for further participation in the study, then please answer the following questions. Otherwise, please continue to submit the survey.

Name

Email

Survey Completion

0% 100%

>>

Submit Survey page:

IV. Study Consent Form

The University of
Science and Technology.
And Life.

Informed Consent Agreement for Participation in a Research Study

Investigators: Zachary Arnold, Danielle LaRose
Contact Information: smartphonegait@wpi.edu
Title of Research Study: Smartphone Gait Inference

Introduction:
You are being asked to participate in a research study. Before you agree, however, you must be fully informed about the purpose of the study, the procedures to be followed, and any benefits, risks or discomfort that you may experience as a result of your participation. This form presents information about the study so that you may make a fully informed decision regarding your participation.

Purpose of the study:
In this study, we will gather data to investigate a more accurate way to measure intoxication levels. This information will be used to develop a smartphone application to more accurately measure a user's level of intoxication based on their gait.

Procedures to be followed:
You will be asked to download one smartphone application. The data collection application will continuously run in the background while recording accelerometer and gyroscope data from your smartphone. The application will also be used to record the amount of any alcohol consumed and when it was consumed through self-reporting.
You will initially be asked to activate the data collection application and do a demonstration walk of 30 meters in order for your unique gait to be recorded and paired with your smartphone. You should then proceed with your everyday, unaltered routine. The study will continue over the following two weeks with the possibility of continuing for another two weeks, depending on the amount and quality of the overall data collected. During this time, you will be asked to record any alcohol consumption from the previous day on the recording application. This will include the type and amount of alcohol as well as an approximation of when it was ingested. Note that you are not required or encouraged to consume alcohol, and any alcohol you consume is of your own choice or volition.

Risks to study participants:
There are no direct risks to the participants of the study. You are asked to go about each day as they normally would. Any alcohol consumed is of your own volition. You do not give up any of your legal rights by signing this statement.

Benefits to research participants and others:
There are no direct benefits to the participants of the study. If the study is successful, the overall results of this experiment will provide more accurate information to future users on their intoxication levels without the hassle of reporting their activities.

Record keeping and confidentiality:
Records of your participation in this study will be held confidential so far as permitted by law. However, the study investigators and, under certain circumstances, the Worcester Polytechnic Institute Institutional Review Board (WPI IRB) will be able to inspect and have access to confidential data that identifies you by name. Any publication or presentation of the data will not identify you.

Compensation or treatment in the event of injury:
Since there are no direct risks associated with this research, there will be no compensation provided in the event of injury.

Cost/Payment:
You may enter to win a $25 Visa gift card.

For more information about this research or about the rights of research participants, or in case of research-related injury, contact:
WPI Institutional Review Board Chair:
Professor Kent Rissmiller, Tel. 508-831-5019, Email:
kjr@wpi.edu WPI's University Compliance Officer:
Michael J. Curley, Tel. 508-831-6919, Email:
mjcurley@wpi.edu Primary Investigator:
Professor Emmanuel Agu, Email:
emmanuel@cs.wpi.edu Co-Investigator:
Professor Joseph Petruccelli, Email:
jdp@wpi.edu Student Investigators:
Zachary Arnold, Email:
zparnold@wpi.edu Danielle LaRose,
Email: dmlarose@wpi.edu

Your participation in this research is voluntary. Your refusal to participate will not result in any penalty to you or any loss of benefits to which you may otherwise be entitled. You may decide to stop participating in the research at any time without penalty or loss of other benefits. If you choose to do so, you will be given the option to erase all previous data and to have it not used in the study. There will not be any repercussions from the university, including grades or academic standing. The project investigators retain the right to cancel, postpone, or extend the experimental procedures at any time they see fit.

By signing below, you acknowledge that you have been informed about and consent to be a participant in the study described above. Make sure that your questions are answered to your satisfaction before signing. You are entitled to retain a copy of this consent agreement.

______________________________ Date: _________________________
Study Participant Signature

APPROVED
WPI IRB 1
1/21/15 to1/20/16

____
Study Participant Name (Please print)

___________________________________ Date: __________________________
Signature of Person who explained this study

V. MATLAB Scripts

a. Data Collection

```
% Starts data collection for Mobile MATLAB and does simple feature
% calculation
function dataCollection(weight,gender,drinks)

m = mobiledev();

[y,Fs] = audioread('ding.mp3');
sound(y,Fs);
m.AccelerationSensorEnabled = 1;
m.logging = 1;
pause(12);
m.logging  = 0;
sound(y,Fs);

[a, t] = accellog(m);

copy = [a,t];

ar = featuresForSingleStreamOfData(a(:,1),a(:,2),a(:,3),t);

ar = [ar weight drinks];

dlmwrite('C:\Users\Zach\Documents\MATLAB\matlab\MQP\3212015DataCollection.csv
',ar,'-append');

clear m
end
```

b. Feature Calculation

```
%calculate the number of steps for each trial
function ar = featuresForSingleStreamOfData(x,y,z,t)

retArray = [];

[steps,loc] = numSteps(x,y,z);
[skew,kurt] = skewAndKurt(x,y,z);
cadence = averageCadence(steps,t,loc);
retArray(length(retArray)+1) = steps;
retArray(length(retArray)+1) = cadence;
retArray(length(retArray)+1) = skew;
retArray(length(retArray)+1) = kurt;
retArray(length(retArray)+1) = gaitVelocity(cadence,steps);
retArray(length(retArray)+1) = diffAvgStepLength(steps,t,loc,13);
retArray(length(retArray)+1) = ratio(x,y,z);
retArray(length(retArray)+1) = diffAvgStepTime(steps,t,loc);

mag = (x.^2+y.^2+z.^2);
magNoG = mag - mean(mag);
```

```
retArray(length(retArray)+1) = bandpower(magNoG);
retArray(length(retArray)+1) = snr(magNoG);
retArray(length(retArray)+1) = thd(magNoG);

ar = retArray;

end
```

c. Number of Steps

```
%Calculate the number of step taken
%x is the x output
%y is the y output
%z is the z output
function [s,l] = numSteps(x, y, z)

[mag] = sqrt(x.^2+y.^2+z.^2);

%adjust for gravity
magNoG = mag-mean(mag);

%count stemps
minPeakHeight = std(magNoG);

[pks, locs] = findpeaks(magNoG, 'MinPeakHeight', minPeakHeight);

s = numel(pks);

l = locs;

end
```

d. Skewness and Kurtosis

```
function [w,k] = skewAndKurt(x,y,z)

[mag] = sqrt(sum(x.^2+y.^2+z.^2,2));

%adjust for gravity
magNoG = mag-mean(mag);

w = skewness(magNoG);

k = kurtosis(magNoG);

end
```

e. Average Cadence

```
%Determines the average cadence for the person
%s is the number of steps
%t is the time
```

```
%l is the location of each step
function ac = averageCadence(s,t,l)

timeOfStep = [t(l(1))];
stepTime = timeOfStep(1) - 0;
stepTimeArray = [stepTime];

for i=2:s
    timeOfStep = [timeOfStep t(l(i))];
    stepTime = timeOfStep(i) - timeOfStep(i-1);
    stepTimeArray = [stepTimeArray stepTime];

end

totalTime = sum(stepTimeArray);

ac = s/totalTime;

end
```

f. Gait Velocity

```
%Calculates the average gait velocity
%c is the average steps/second
%   (from averageCadence)
%s is the average steps/meter
%   (from averageStepLength)
function gv = gaitVelocity(c,s)

gv = c/s;

end
```

g. Difference from Average Step Length

```
%Calculates the difference from the average in the length of each step
%s is the number of steps
%t is the time
%l is the location of each step in the data rows
%len is the length (in meters) of the area
function dasl = diffAvgStepLength(s,t,l,len)

c = averageCadence(s,t,l);
asl = averageStepLength(s,len);
v = gaitVelocity(c,asl);

timeOfStep = [t(l(1))];
stepLength = v*(timeOfStep(1) - 0);
stepLengthArray = [stepLength];

for i=2:s
    timeOfStep = [timeOfStep t(l(i))];
    stepLength = v*(timeOfStep(i) - timeOfStep(i-1));
```

```
    stepLengthArray = [stepLengthArray stepLength];

end

diffAvg = asl - stepLengthArray(1);

for j=2:s
    difference = asl - stepLengthArray(j);
    diffAvg = [diffAvg difference];
end

dasl = sum(diffAvg);

end
```

h. Ratio

```
%Ratio of high and low frequencies
function r = ratio(x,y,z)

sp = spectralPeaks(x,y,z);
sp = sp(1:floor(end/2));

numSP = length(sp);

totalSP = sum(sp);

averageSP = totalSP/numSP;

dev = std(sp);

high = [];
low = [];

for i=1:numSP
    if(sp(i) < (mean(sp)))
        low = [low sp(i)];
    elseif (sp(i) > (mean(sp)))
        high = [high sp(i)];
    end
end

r = sum(low)./sum(high);

end
```

i. Difference from Average Step Time

```
%Calculates the difference from the average in the time of each step
%s is the number of steps
%t is the time
%l is the location of each step
```

```matlab
function dast = diffAvgStepTime(s,t,l)

timeOfStep = [t(l(1))];
stepTime = timeOfStep(1) - 0;
stepTimeArray = [stepTime];

for i=2:s
    timeOfStep = [timeOfStep t(l(i))];
    stepTime = timeOfStep(i) - timeOfStep(i-1);
    stepTimeArray = [stepTimeArray stepTime];

end

totalTime = sum(stepTimeArray);

as = s/totalTime;

diffAvg = as - stepTimeArray(1);

for j=2:s
    difference = as - stepTimeArray(j);
    diffAvg = [diffAvg difference];
end

dast = sum(diffAvg);

end
```

VI. Usability Post Survey

WPI

Heavy drinkers have a tendency to forget how much, who they were with, or where they were when they drank. This application introduces passive machine detection of alcohol intoxication at no cost to the user. It provides them with real-time feedback on their approximate intoxication level. In the future, a log of a smartphone user's intoxication levels, along with contexts in which users drank in the past (times, locations, and with who), will also be presented. By being able to receive objective feedback, the user is able to self-correct their behavior before it becomes problematic.
Towards the envisioned usage scenario, the current application prototype simply infers a user's alcohol intoxication level. It can be used to infer whether the user is sober (0-2 drinks), tipsy (3-6 drinks), or drunk (>6 drinks).

Do you see this envisioned app being useful when complete?

- Yes
- No

Would you recommend this envisioned application to others when it is complete?

- Yes
- No

Would you actually use this envisioned application?

- Yes
- No

Did you like the current application? Why or why not?

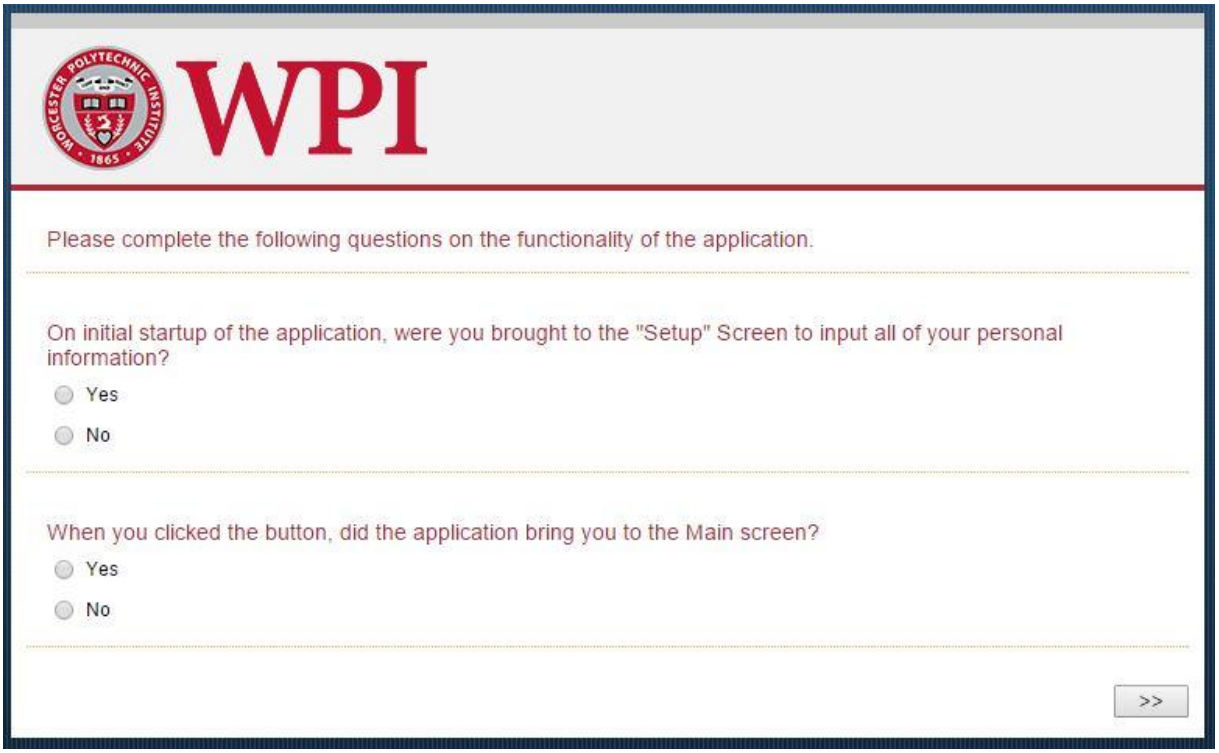
Did you like the appearance of the application? Why or why not?

How accurate were its inferences on your intoxication levels when you used it?

Not at all accurate — Extremely accurate

0	1	2	3	4	5	6	7	8	9	10

>>



WPI

Please complete the following questions on the functionality of the application.

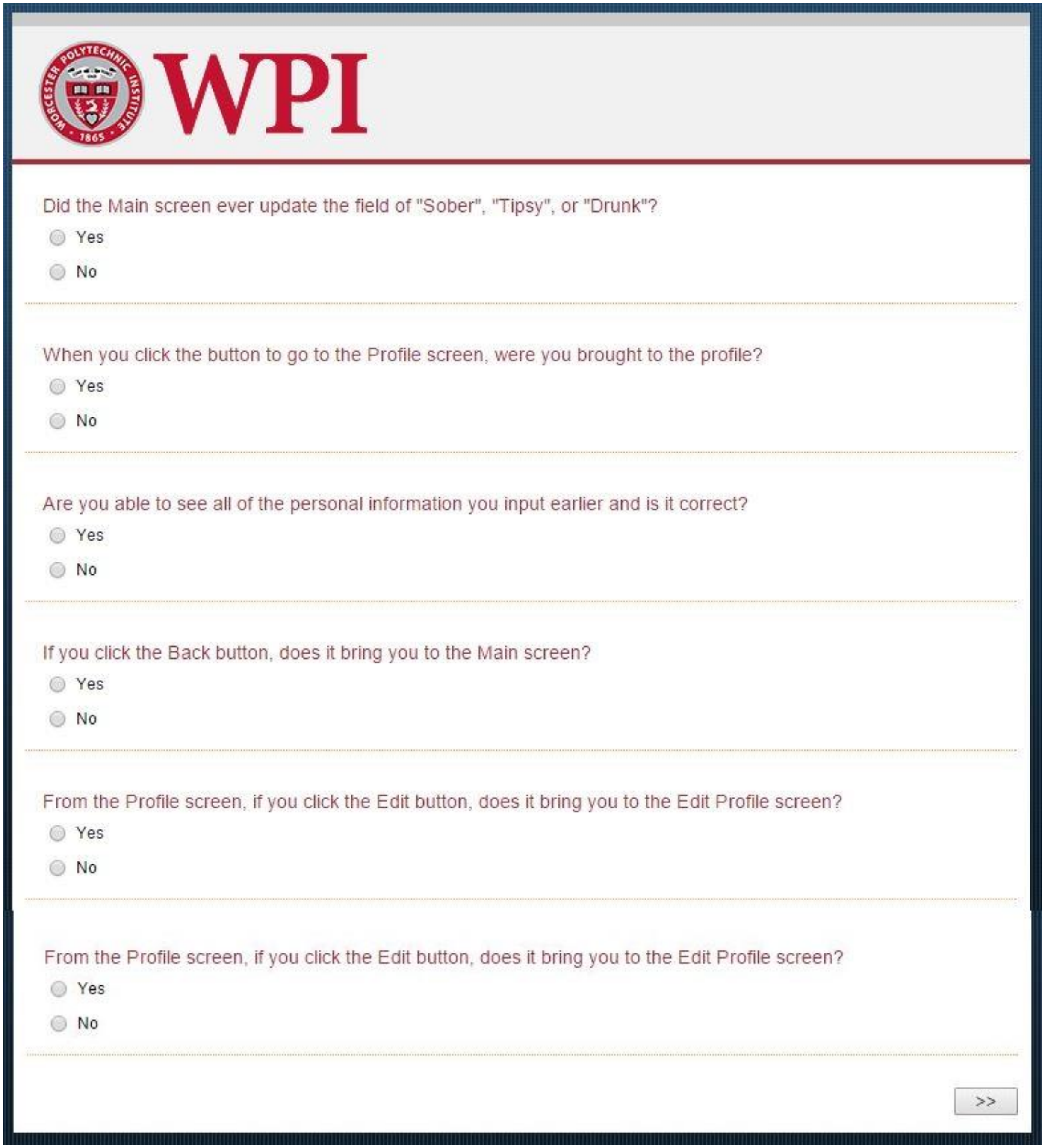
On initial startup of the application, were you brought to the "Setup" Screen to input all of your personal information?

- Yes
- No

When you clicked the button, did the application bring you to the Main screen?

- Yes
- No

>>



WPI

Did the Main screen ever update the field of "Sober", "Tipsy", or "Drunk"?

- Yes
- No

When you click the button to go to the Profile screen, were you brought to the profile?

- Yes
- No

Are you able to see all of the personal information you input earlier and is it correct?

- Yes
- No

If you click the Back button, does it bring you to the Main screen?

- Yes
- No

From the Profile screen, if you click the Edit button, does it bring you to the Edit Profile screen?

- Yes
- No

From the Profile screen, if you click the Edit button, does it bring you to the Edit Profile screen?

- Yes
- No

>>

WPI

Does the Edit Profile screen correctly display the previous information?

- Yes
- No

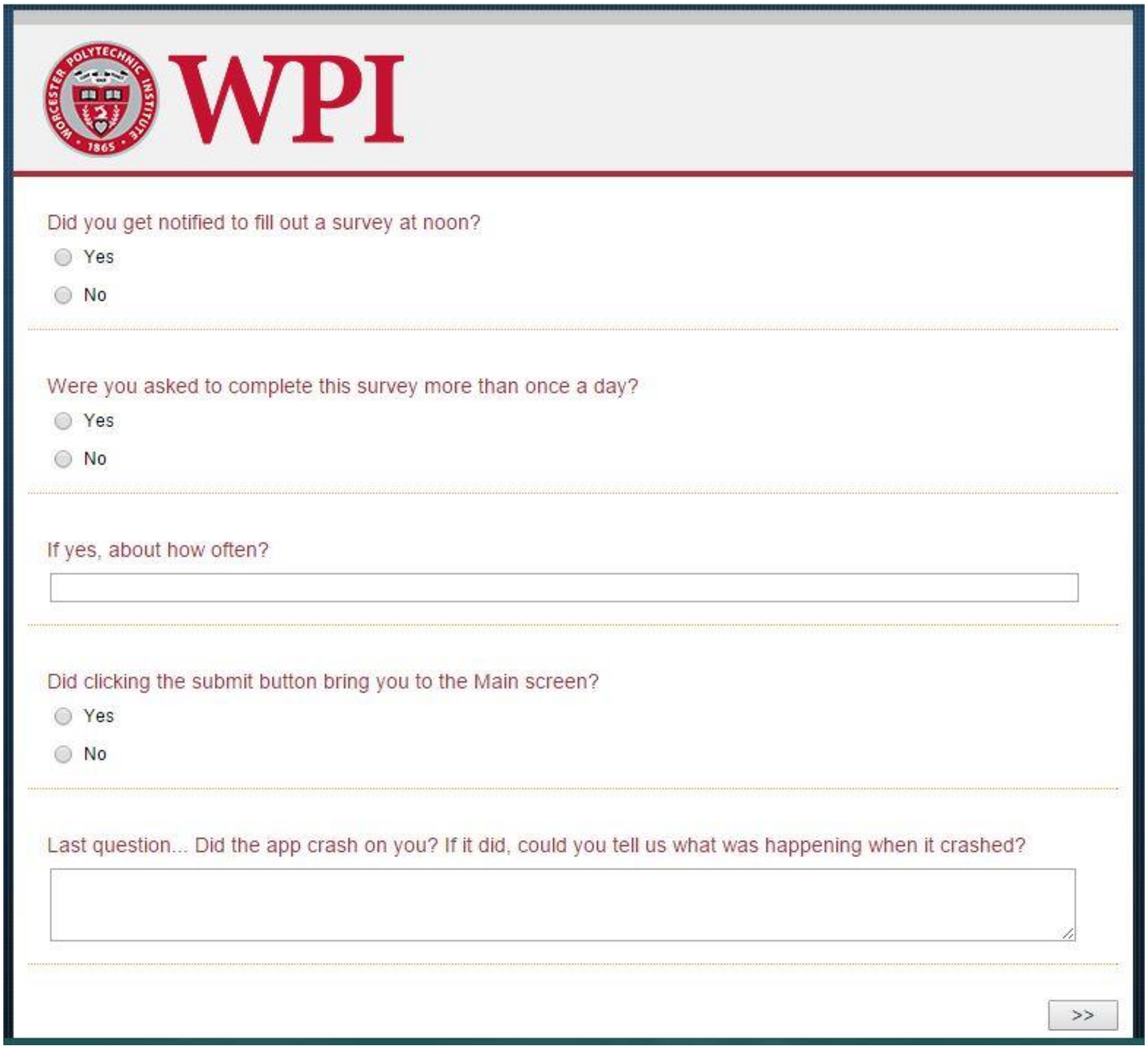
Did the Cancel button return you to the Profile screen with no changes?

- Yes
- No

Did the Save button return you to the Profile screen with your changes?

- Yes
- No

>>


Did you get notified to fill out a survey at noon?

- Yes
- No

Were you asked to complete this survey more than once a day?

- Yes
- No

If yes, about how often?

Did clicking the submit button bring you to the Main screen?

- Yes
- No

Last question... Did the app crash on you? If it did, could you tell us what was happening when it crashed?

>>


We thank you for your time spent taking this survey.
Your response has been recorded.

Survey Completion

0% 100%

VII. Usability Study Results

Initial Report

Last Modified: 04/24/2015

1. Do you see this envisioned app being useful when complete?

#	Answer		Response	%
1	Yes		6	86%
2	No		1	14%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	2
Mean	1.14
Variance	0.14
Standard Deviation	0.38
Total Responses	7

2. Would you recommend this envisioned application to others when it is complete?

#	Answer		Response	%
1	Yes		7	100%
2	No		0	0%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	7

3. Would you actually use this envisioned application?

#	Answer		Response	%
1	Yes		4	57%
2	No		3	43%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	2
Mean	1.43
Variance	0.29
Standard Deviation	0.53
Total Responses	7

4. Did you like the current application? Why or why not?

Text Response
Notifications to submit my number of drinks are spammed. Also, don't ask when i started/stopped drinking when i said zero drinks
Right now its just a printed status on a single screen, and there is no indication if the app is running/analyzing or actually doing anything. It would be nice if there were some indicator (persistent notification ?) that told me its running and my intoxication
Yes - functional and easy to use
Yes
Yes, I found it to be very insightful
I do, because it works.
yes

Statistic	Value
Total Responses	7

5. Did you like the appearance of the application? Why or why not?

Text Response
it was fine
In terms of UI, what it is currently is fine. But for a better User Experience, see comment above
Easy to use
Yes it's shnazzy
Yes but it could use more work
Not really. It could be improved, but there is some basic functionality there.
yes

Statistic	Value
Total Responses	7

6. How accurate were its inferences on your intoxication levels when you used it?

#	Answer		Response	%
0	0		0	0%
1	1		0	0%
2	2		0	0%
3	3		0	0%
4	4		0	0%
5	5		1	14%
6	6		0	0%
7	7		0	0%
8	8		1	14%
9	9		1	14%
10	10		4	57%
	Total		7	100%

Statistic	Value
Min Value	5
Max Value	10
Mean	8.86
Variance	3.48
Standard Deviation	1.86
Total Responses	7

7. On initial startup of the application, were you brought to the "Setup" Screen to input all of your personal information?

#	Answer		Response	%
1	Yes		7	100%
2	No		0	0%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	7

8. When you clicked the button, did the application bring you to the Main screen?

#	Answer		Response	%
1	Yes		7	100%
2	No		0	0%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	7

9. Did the Main screen ever update the field of "Sober", "Tipsy", or "Drunk"?

#	Answer		Response	%
1	Yes		4	57%
2	No		3	43%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	2
Mean	1.43
Variance	0.29
Standard Deviation	0.53
Total Responses	7

10. When you click the button to go to the Profile screen, were you brought to the profile?

#	Answer		Response	%
1	Yes		7	100%
2	No		0	0%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	7

11. Are you able to see all of the personal information you input earlier and is it correct?

#	Answer		Response	%
1	Yes		7	100%
2	No		0	0%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	7

12. If you click the Back button, does it bring you to the Main screen?

#	Answer		Response	%
1	Yes		7	100%
2	No		0	0%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	7

13. From the Profile screen, if you click the Edit button, does it bring you to the Edit Profile screen?

#	Answer		Response	%
1	Yes		7	100%
2	No		0	0%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	7

14. Does the Edit Profile screen correctly display the previous information?

#	Answer		Response	%
1	Yes		7	100%
2	No		0	0%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	7

15. Did the Cancel button return you to the Profile screen with no changes?

#	Answer		Response	%
1	Yes		7	100%
2	No		0	0%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	7

16. Did the Save button return you to the Profile screen with your changes?

#	Answer		Response	%
1	Yes		7	100%
2	No		0	0%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	7

17. Did you get notified to fill out a survey at noon?

#	Answer		Response	%
1	Yes		6	86%
2	No		1	14%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	2
Mean	1.14
Variance	0.14
Standard Deviation	0.38
Total Responses	7

18. Were you asked to complete this survey more than once a day?

#	Answer		Response	%
1	Yes		6	86%
2	No		1	14%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	2
Mean	1.14
Variance	0.14
Standard Deviation	0.38
Total Responses	7

19. If yes, about how often?

Text Response
50
If the survey you're asking about is to state my drinking habits recently, then I would have a notification about this very frequently. Sometimes I would dismiss it, other times i would not, so it is hard to say how often it had prompted me vs being kept around if i hadn't answered yet.
Twice
3 times
5
often

Statistic	Value
Total Responses	6

20. Did clicking the submit button bring you to the Main screen?

#	Answer		Response	%
1	Yes		7	100%
2	No		0	0%
	Total		7	100%

Statistic	Value
Min Value	1
Max Value	1
Mean	1.00
Variance	0.00
Standard Deviation	0.00
Total Responses	7

21. Last question... Did the app crash on you? If it did, could you tell us what was happening when it crashed?

Text Response
no
No.
Nope!
Nope
nope
Once, but it was because it was deep in the background.
no

Statistic	Value
Total Responses	7