ARCHIVES
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
AUG 20 2015
LIBRARIES

A Data Analysis System for Mobile Phone

By

Yi-An Lai

B.S., Massachusetts Institute of Technology (2013)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHMOLOGY

August 2014 [september 2014]

© Massachusetts Institute of Technology 2014. All rights reserved.

Signature redacted

Author . . . .
Department of Electrical Engineering and Computer Science
August, 2014

Signature redacted

Certified by . . . .
Alex "Sandy" Pentland
Professor
Thesis Supervisor

Signature redacted

Accepted by . . . .
Albert R. Meyer
Chairman, Department Committee on Graduate Theses

A Data Analysis System for Mobile Phone

By

Yi-An Lai

Submitted to the Department of Electrical Engineering and Computer Science
on August 22, 2014, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Computer Science and Engineering

Abstract

This thesis shows the experiment of using the mobile phone application Social Health Tracker to collect participants' mobile phone data and survey answers regarding their social, physical activity and mental concentration level. It also shows the design and the implementation of a Data Analysis System that provides automation for generating Relationship Graphs of the mobile phone data and the survey answers. The system can be broken into three parts. The system first takes the raw data that are in CVS format to generate Probe Data Packages, which are the data from the mobile phone, and the Survey Data Packages, which are the data from survey answers. The system then takes each of the Probe Data Packages and their corresponding Survey Data Package to generate Formatted Data, which will be further used to create Relationship Graphs. Multiple Relationship Graphs have been generated to demonstrate the results of the experiment and the feasibility of the system from the development perspectives.

Thesis Supervisor: Alex "Sandy" Pentland
Title: Professor

Acknowledgements

First of all, I would like to thank my thesis advisor, Sandy Pentland, for allowing me to work in his group. Human Dynamics Group is by far the most interesting and diversed lab that I've ever worked in. Furthermore, Prof. Pentland has always been very helpful in providing insights and guiding the direction of the project.

I would like to thank Brian Sweatt for helping me with all the questions from the first day I joined the lab. He has been the best collaborator and mentor who would respond immediately to my lengthy emails with all the questions and always find time to have meetings with me to provide me suggestions on the project. I would like to thank Arek Stopcsynski for keep me company in the office and sending me different resources that are relevant to my research.

Lastly, I would like to think my parents, Linda Bergkamp and Steve Bergkamp for constantly reminding me to concentrate on my goal and to always try my best on whatever that I am working on. I would like to also thank my grandmother for always believing in me when I ran into any obstacles.

Contents

List of Figures

List of Tables

Listings

Chapter 1

Introduction

1.1 Motivation

Most smart phones now have built in sensors that measure motion, orientation and different environmental conditions. All these sensors are capable of providing raw data with high precision and accuracy, such as monitoring three-dimensional device movement or positioning or monitoring changes in the ambient environment near a device.[1] With the advancement of the smart phone technology, it has granted us an unprecedented ability to observe and collect users' behaviors, such as how physically active they are, how frequently do they make phone calls or send messages or where are the places they have been to, as quantified data and further analyze it.

Given the technology and smart phone's ubiquitous presence, which had reached 1 billion in 2012, a 47 percent increase from a year earlier, this provides scientists a large amount of data and gives rise to the computational social science. [2] Computational social science is the interdisciplinary science of complex social system and it studies individual people or in groups using big data, such as mobile phone history, emails, GPS traces, Wi-Fi points, and data from social networks. [3] In the research paper, Computational Social Science, Lazer, Pentland, et al. mention that digital data could potentially increase our understanding of the underlying information of individuals. [4]

Due to the passive nature of smart phone's data collection where the mobile device is able to collect information without any interaction with the user, the information could potentially provide us insights of user's day-to-day routines. The goal of this project is to utilize the data collected from the mobile phone and survey answers gathered from the users to infer individual's level of socialization, mental focus ability and physical activity level and utilize effective data visualization to present the data. The project aims to investigate the relationships between different signals and survey responses from the participants. If we can demonstrate certain data's ability to infer those three subjects, the predictions generated from the mobile phone data can potentially be used medically as a tool to provide people suggestions on improving the conditions on those three areas or be used by doctors as a reference to better understand patients' social and physical behavior in order to better assist them.

1.2 Related Works

Utilizing a large data set that is collected over a long duration of time for learning about participants' behaviors and health conditions in an attempt to improve them has been a rather common practice in various research fields. Nevertheless, in the past, collecting such data is not only costly but also required enormous amount of effort and time. Moreover, it is also hard to retain a large number of research participants over a long period of time when the data collection requires researchers to seek information from the participants actively. As the result, the throughput of data samples in research that utilize this kind of practice is often very low.

However, given the recent advancement of the mobile phone technology, mobile phones are always continuously collecting information from the users and this requires no active interaction with the users for their input of the information. Furthermore, we are able to leverage on mobile phone's ubiquitousness to obtain data that has very high sampling rate.

1.2.1 Mobile Phones Data Set

In "Reality mining: sensing complex social system," Eagle and Pentland designed a system for sensing complex social systems with data collected from 100 mobile phones over the courses of 9 months. They demonstrated that using standard Bluetooth-enabled mobile telephones to collect data, such as call records, Bluetooth proximity logs, could provide us great insight on social network and individual's behavior pattern. [5]

In "Inferring friendship network structure by using mobile phone data," Eagle, Pentland, etal. compares observational data from mobile phones with standard self-report survey data. [6] They found that information of the two data sources is overlapping but distinct, and they demonstrated that it is possible to accurately infer 95% of friendships based on the observational data alone. [6] And the behavioral patterns could potentially predict the individual-level outcomes such as their job satisfaction.

1.2.2 Individual Based Data Collection

Although we could benefit from the large scale data sets provided from the service providers, this information is constrained in specific areas, such as call history, emails, texts, financial transactions. In "Social Sensing for Epidemiological Behavior Change," Madan et al. [7] tackled the problem of gathering individual's data with an alternative approach by collecting this information through survey questionnaire. This approach allows the researchers to obtain contextual information on the end users. [7] Madan et al. present the results based on mobility and communication pattern that mobile phone sensing data could potentially be used for measuring and predicting the health condition of the users. [8]

Chapter 2

Methodology

2.1 Overview

The research goal of this project is to better understand the potential correlation between participants' health and social behaviors and the information collected from their mobile phone data. In this experiment, a mobile phone application, Social Health Tracker, was required be downloaded by the participants and the application will send survey questionnaires to the participants to collect their information. The survey questionnaires required the users to assess their own health, social and mental status and report to the application. The mobile phones were used as sensors to gather information of users' activities on the phone, such as phone calls, text messages, Bluetooth signals, Wi-Fi points and the history of the accelerometer. Users were required to have an Android OS based mobile phones to download the application and participate in the study. The project did not sponsor the mobile device or the phone plans, so the participants would have their own android phones for the study.

2.2 Related Works

In "Social fMRI: Investigating and shaping social mechanisms in the real world," Aharony et al.[2] developed a mobile-phone-based social and behavioral sensing system and deployed in the wild for over 15 months (see Figure 2-1). This sensing system runs on Android operating- system based mobile device and it can passively and continuously collect a broad range of data signals without any active input from the users. Each type of signal collected by the system is encapsulated as a conceptual "probe" object. [9] The probes terminology is used rather than "sensors" as probes include traditional sensors such as GPS or accelerometer, but also other types of information not traditionally considered as sensor data, such as file system scans or logging user behavior inside applications. [9] The back-end system processes all the incoming SQLite files, inserts them into a central MySQL database, and sends email reports to investigators about status of phones and alerts of any issues . [9] An object-relational-mapper (ORM) then enables representing all data as code objects. [9]

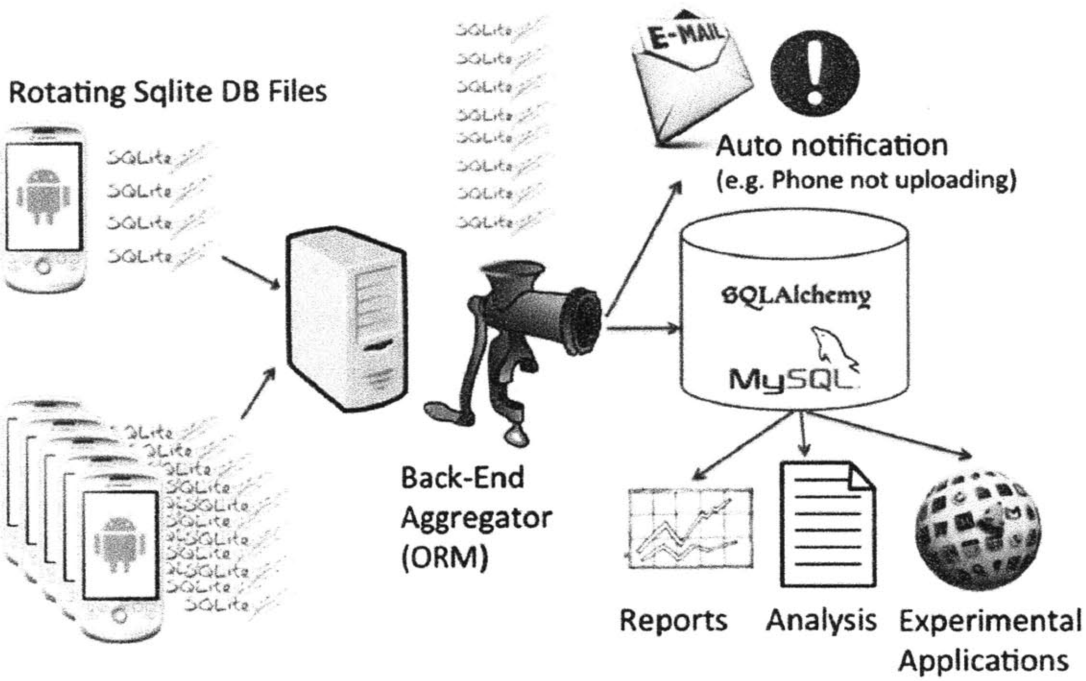


Figure 2-1: The overview of the back-end data flow

The dataset that we obtained in this research was collected using this system with a configuration that included over 25 different types of data signals. In addition to this signal collection system, we also included an on-phone survey component, which the mobile application, Social Health Tracker, was to be downloaded by the participants on the mobile phone and it would send out surveys to the participants twice a day.

2.3 Mobile Phone Data Collection System

Android operation system based mobile phones with social and behavior software sensing platform were used to collect users' survey answers and data from the mobile phone sensors (see Figure 2-2). The software for sensing platform, Funf Opening Sensing Framework, configured to periodically sense and record over 25 different types of data signals, such as call logs, text logs, Bluetooth signals, Wi-Fi signals, accelerometer history. We required our participants to download the Social Health Tracker application that's available on Google Play. This application sends out surveys to the users twice a day (see Figure 2-3). The application enables the users to see their activity levels in a timeline fashion and also presents the result of their activity levels compared to other users (see Figure 2-4). This system is designed to continuously run in the background and will trigger to make sure itself restarts when the phone turns on or after the service is terminated. [9] The system schedules different data collection actions and the configuration is set so that battery intensive actions are performed in intervals allowing usefulness while minimizing battery drain. [9] We did not provide mobile phone plans or data plans. Participants used their own Android phones with desired provider and the mobile phones had to be the primary phone for the duration of the study.

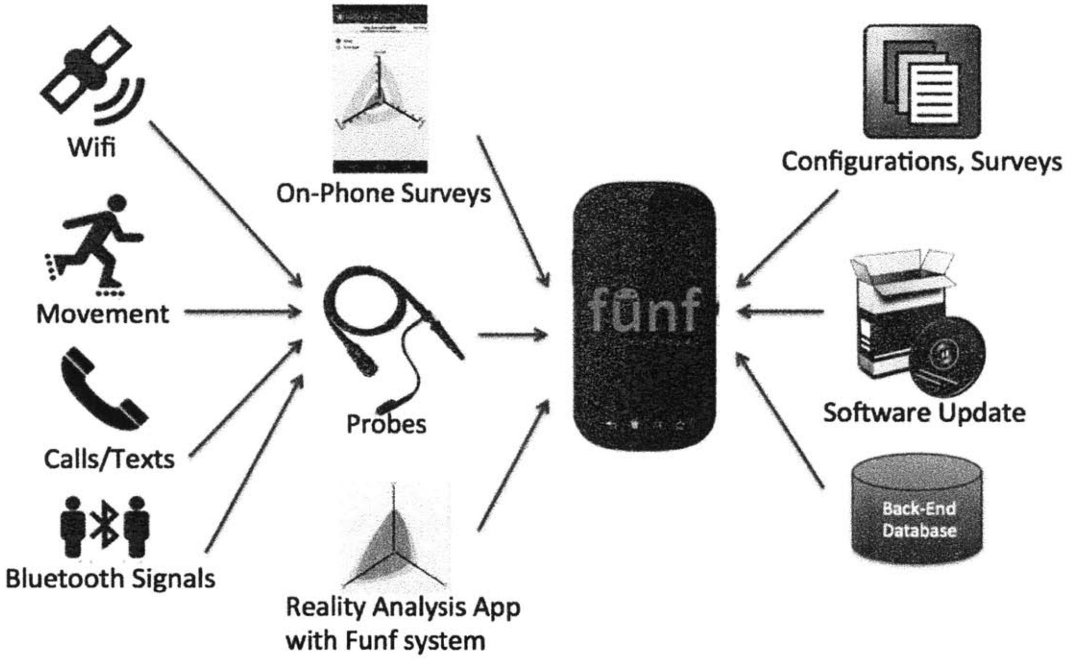
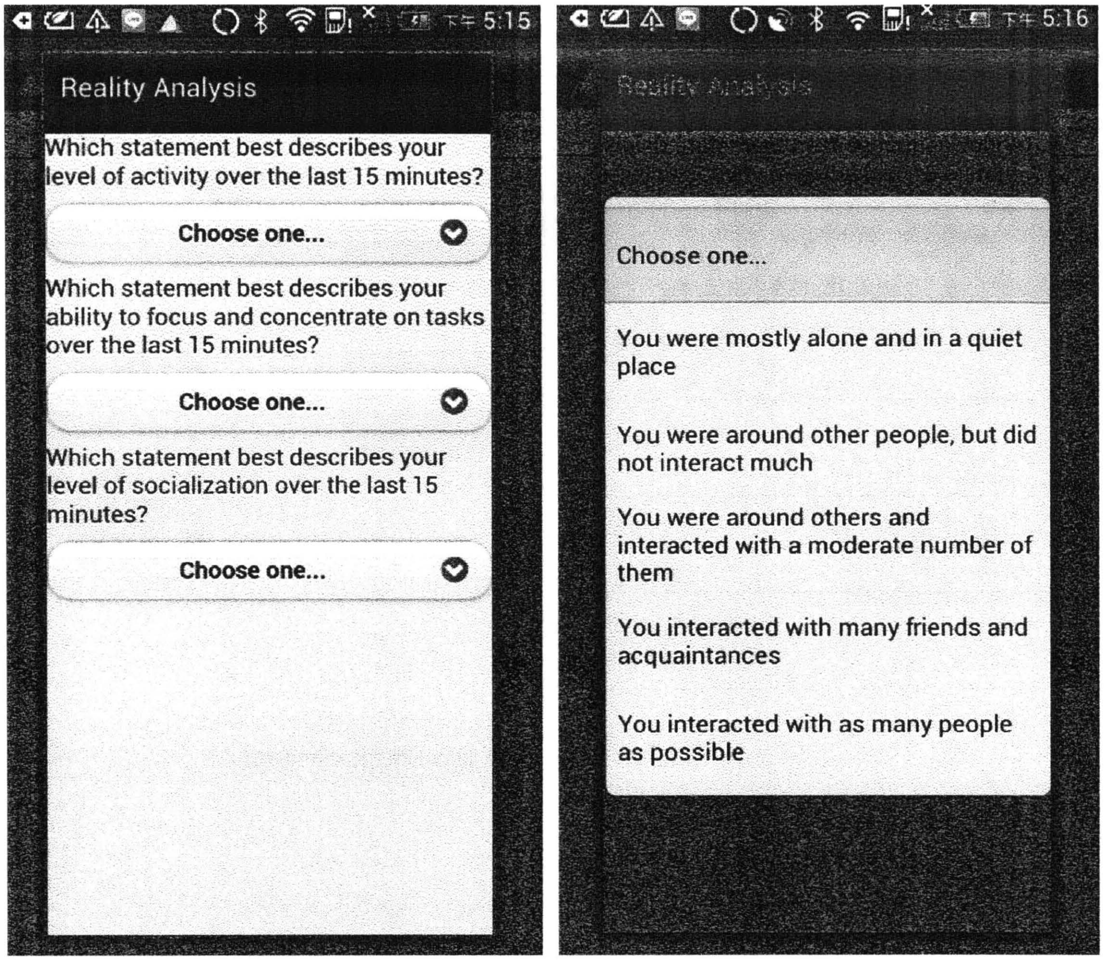


Figure 2-2: Overview of the Phone System



Which statement best describes your level of activity over the last 15 minutes?

Which statement best describes your ability to focus and concentrate on tasks over the last 15 minutes?

Which statement best describes your level of socialization over the last 15 minutes?

You were mostly alone and in a quiet place

You were around other people, but did not interact much

You were around others and interacted with a moderate number of them

You interacted with many friends and acquaintances

You interacted with as many people as possible

Figure 2-3: Sample survey on participant's mobile phone screen

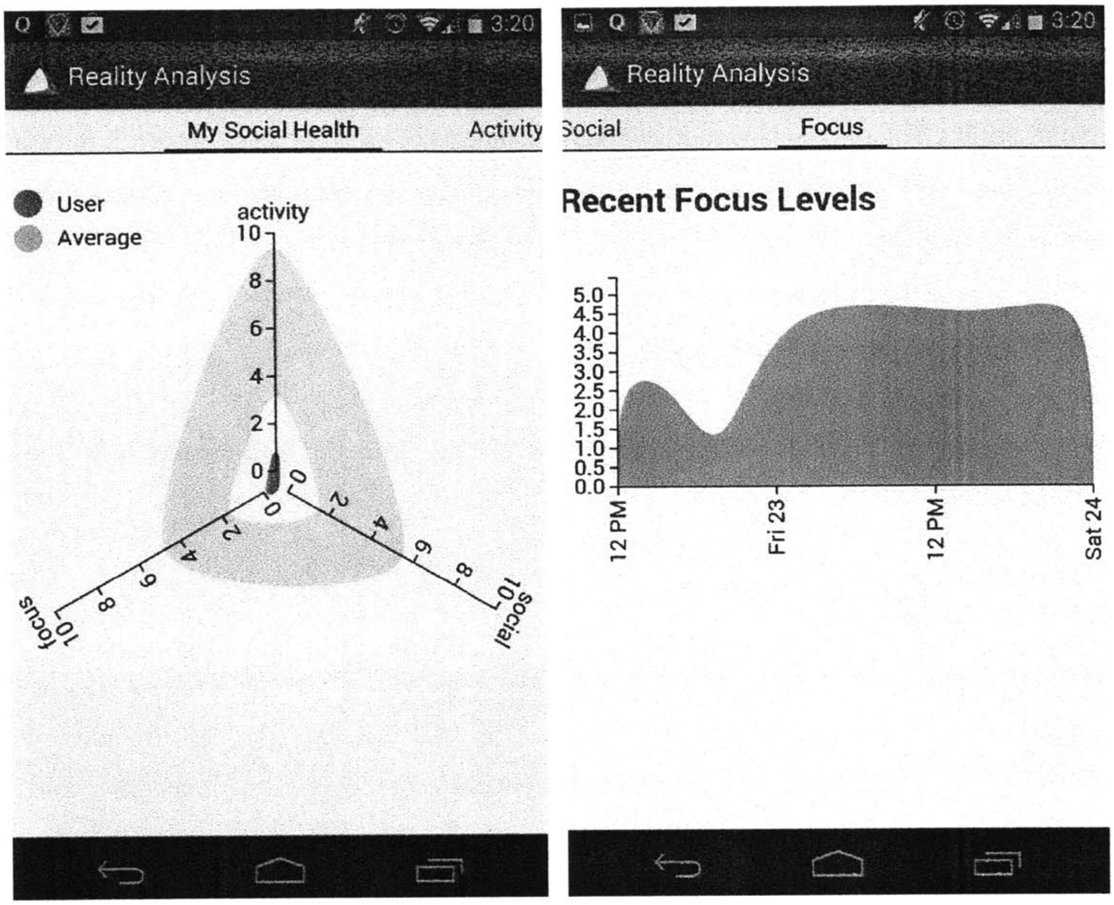


Figure 2-4: Sample result that shows participant's levels of physical activity, mental focus ability and social activity in comparison to other users (left). Sample result shows participants mental focus levels on a time line (right).

2.3.1 Privacy Considerations

This study was conducted under strict protocol guidelines. Our primary concern for the design of the research was to protect the privacy and sensitive information of our participants. To achieve this, all participants' real world personal identifiers are linked to coded identifiers. For instance, participants are represented as numbers in our database and there's no sufficient information available in the database that will allow anyone to link the participants and the numbers together. All human-readable information, such as phone numbers, text messages, is encrypted as hashed identifiers. No information is ever saved in clear text. The study aimed to be unobtrusive as possible to the participants' life routines. The data collection of the mobile phone sensors did not require any interactions from the users. The only out-of-routine behavior that asked of participants was filling out surveys twice per day. And the surveys did not ask for anything personal information from the participants.

2.4 Community Overview

The pilot test of this research started in 4/18/2013 with 21 participants. All the participants are members of Human Dynamics Group.

2.5 Probe Data

To collect the mobile phone data, we used the Funf Open Sensing Framework, an Android-based extensible framework developed by MIT Media Lab. This framework provides us a reusable set of functionalities enabling the collection, uploading, and configuration for a broad range of data type. [10] The data collected is automatically saved in SQLite format. When the user is in the absence of network access, the phone will accumulate the collected database files locally; when server connection is available, the system will them upload the files to the central system. [9] In order to minimize the impact on the user, probes built using Funf coordinate with each other to reduce power and processor usage. [10] The following table lists different signals that are collected by different probes during different interval of time.

Signal	Interval	Notes
ActivityProbe	60s	Information about how active the person is. Uses the AccelerometerProbe data to calculate how many intervals the variance of a device's acceleration is above a certain threshold. Intervals are 1 seconds long. DURATION (default 5s)
AndroidInfoProbe	3600s	Information about the version of Android the device is running.
BatteryProbe	300s	Information about the type and current state of the battery in the device.
BluetoothProbe	300s	Detects Bluetooth devices within range.
CallLogProbe	3600s	Information about the calls that are made by the device. Sensitive information is normalized and hashed consistently and can be compared to contacts on this device, or with other devices.
CellProbe	1200s	Information about the ids for the current cell tower the device is connected to.
HardwareInfoProbe	300s	Information about the specific hardware the device is running, including component identifiers.
LightSensorProbe	3600s	Information about the ambient light level in SI lux units. DURATION (default 60s)
LocationProbe	1800s	Information about the longitude and latitude.
ProximityProbe	3600s	Information about far the front of the device is from an object. DURATION (default 60s)
ScreenProbe	N/A	Information about when the screen turns off and on.

Signal	Interval	Notes
SmsProbe	3600s	Information about messages sent and received by this device using SMS. Sensitive data is hashed for user privacy.
TimeOffsetProbe	21600s	Checks NTP servers to determine the devices current offset from the times ervers.
WifiProbe	1200s	Information about the available Wi-Fi access points.

Table 2.1: The detailed information of each signal that is collected by the Funf Open Sensing Framework.

2.6 Survey Data

In this experiment, participants would complete the survey on a regular basis on their mobile phone. The surveys would be sent out to them twice a day at random time. They would be asked questions regarding their level of socialization, ability to concentrate on tasks and level of physical activity level of the last 15 minutes. The data included the following signals:

(1) Socialization:

Socialization refers to any in person face to face interaction. Interactions that are conducted via phone calls, web mails or online social network do not belong to this category. The socialization information was self-reported by participants twice a day on a daily basis where the options of the level of socialization provided included:

Which statement best describes your level of socialization over the last 15 minutes?	
1	You interacted with as many people as possible
2	You interacted with many friends and acquaintances
3	You were around others and interacted with a moderate number of them
4	You were around other people, but did not interact much
5	You were mostly alone and in a quiet place

Table 2.2: Survey question of socialization

(2) Mental concentration

Mental concentration is defined by how focus participants are on any kind of task they are doing. The socialization information was self-reported by participants twice a day on a daily basis where the options of the level of mental concentration provided included:

Which statement best describes your ability to focus and concentrate on tasks over the last 15 minutes?	
1	There was extreme focus and you were "in the zone"
2	You were very focused and could easily concentrate on the task(s) at hand
3	There was a moderate level of concentration and focus
4	You were somewhat focused and could concentrate a little
5	You could not focus and concentrating was impossible

Table 2.3: Survey question of mental concentration

(3) Activity

Activity is defined by how physically active participants are. This category is not limited to sports, activities in the gym or outdoor exercise. For instance, walking around in the office frequently could be considered as physically active. The socialization information was self-reported by participants twice a day on a daily basis where the options of the level of physical activity provided included:

Which statement best describes your level of activity over the last 15 minutes?	
1	An extremely high level of physical activity was performed
2	A very active level of physical activity was performed
3	A moderate level of physical activity was performed
4	A little physical activity was performed
5	Mostly sedentary and engaged in little to no physical activity

Table 2.4: Survey question of physical activity

Chapter 3

Implementing the Data Analysis System

3.1 Overview

Analyzing the data involves many steps from formatting and cleaning the raw data to generating graphs that show the relationship between the signal data collected from probes and the survey answers provided by the participants. First, the system takes the raw data to generate Probe Data Packages and Survey Data Packages, which are data that have been reorganized with selected features that we desired. Next the system takes each of the Probe Data Packages and their corresponding Survey Data Package to create data files that are in a standard format. The system then takes those data files to generate the Relationship Graphs that present the relationship of the mobile phone signals and survey answers.

3.2 Data Package Creation

The raw data from the mobile phone, including the probe data and the survey data, are originally saved in the CSV file format and they are around 2 GB. This CVS file includes the signal data from the mobile phone probes and the collected survey answers. The system first takes the CVS file to process it into the format with timestamps as the header, and the desired features in each row. This step limits the size of the data packages to 5MB each. The process of transforming the raw CSV file into individual data package is the longest step in the set up process. However, once the individual data packages are created, the system will not need to repeatedly run the entire CSV file when it is retrieving data of the probes or the survey answers.

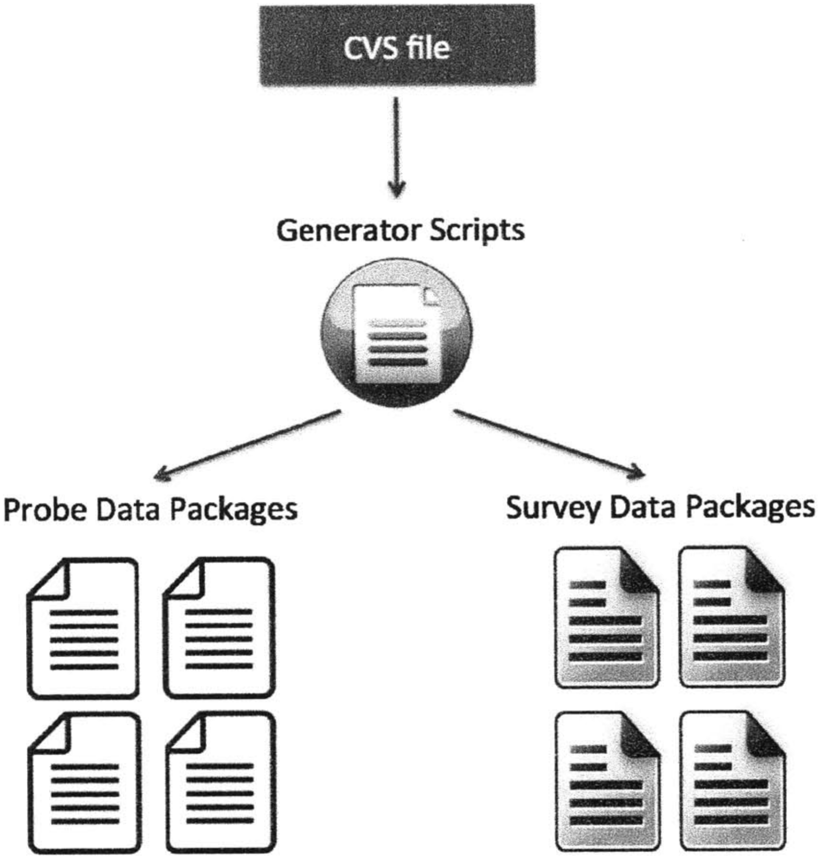


Figure 3.1: Overview of Data Package Creation – the system takes the raw CVS file and feeds it to the Generator Scripts. The Generators Scripts then creates Probe Data Packages and Survey Data Packages.

3.2.1 Data Formatting

Instead of taking all the features of the data, as seen in the following examples, we only select certain features of the data by deleting other columns in the data set. Generator.py takes the raw CSV file and generates the Probe Data Packages and the Survey Data Packages with selected features.

Probe data example from the raw CVS file:

```
2|BluetoothProbe|1386098113.684|{u'android-bluetooth-
device-extra-device': {u'maddress':
u'CB:6E:44:D1:18:1D'}, u'timestamp': 1386098113.684,
u'android-bluetooth-device-extra-rssi': -91, u'android-
bluetooth-device-extra-class': {u'mclass': 7936},
u'android-bluetooth-device-extra-name': u'One'}
```

Survey data example from the raw CSV file:

```
2|ActivityPast3Days|1380208365.416|2
3|ActivityPast3Days|1380208369.778|3
4|ActivityPast3Days|1384965021.495|4
```

The followings are the processed CSV file with selected header and features. For the probe data, Generator.py selects desired features and eliminates the rest, then makes the time stamp as the header.

Processed probe data example from raw CSV file:

```
1380063872.102|{u'android-bluetooth-device-extra-device':
{u'maddress': u'04:0C:CE:21:6A:15'}, u'timestamp':
1380063872.102, u'android-bluetooth-device-extra-class':
{u'mclass': 3801356}, u'android-bluetooth-device-extra-
rssi': -95}
```

For the survey data, the Generator.py selected the time stamp as the header and survey answer as the desired feature in the second column.

Processed survey data example from raw CSV file:

```
1380208365.416|2
1380208369.778|3
1384965021.495|4
```

In the end, Generator.py creates 6 Probe Data Package for each of the following signals: activity, Bluetooth, call logs, text logs, screen on-off history, and Wi-Fi signals. It also creates 3 Survey Data Package for the following survey categories: physical activity, social activity, and mental concentration.

3.3 Generating the Formatted Data

Among the 14 signals collected by the probes, we chose 6 signals to further study the relationships between those signal data and the survey answers. There is a Sensor Script for each signal to take the Probe Data Package and the corresponding Survey Data Package as input to organize data into a dictionary as following format:

{timestamp: [probe data, survey answer]}

For instance, the Accelerometer.py takes the Accelerometer Data Package and all of three Survey Data Packages to produce the Formatted Data. Since survey questionnaire asks the participants about their condition in the past 15 minutes, the Sensor Script selects all the data that are collected 15 minutes prior from the point that the participants answer the survey.

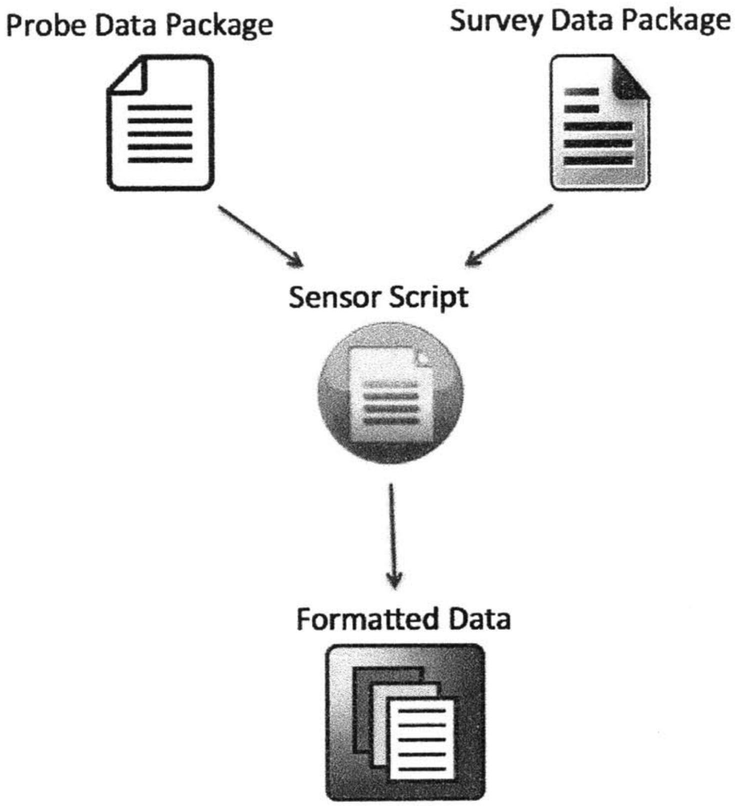


Figure 3.2: Overview of how the system generates Formatted Data – once the Probe Data Package and Survey Data Packages are created, Sensor Script takes each of the Probe Data Packages and their corresponding Survey Data Package to create the Formatted Data.

3.3.1 Activity

Activity sensor records how activate the person is. It uses the data from the Accelerometer Probe to calculate how many intervals the variance of a device's acceleration is above a certain threshold and the intervals are 1 seconds long.[10] The default duration of the probe is 5 seconds and the default period is 60 seconds.

Example Data:

```
{
  u'timestamp': 1386098903.072,
  u'total_intervals': 14,
  u'high_activity_intervals': 0,
  u'low_activity_intervals': 0
}
```

The survey questionnaires ask the participants about their condition in the past 15 minutes, therefore Activity.py calculates the activity level by summing up the all data of the high_activity_intervals in the past 15 minutes from the point when the participants submit their surveys.

Listing 3.3.1: Generating the Formatted Data from the Accelerometer Probe

```
#This will give me the value of the data from funf

data_value = ast.literal_eval(j[1])
high_activity_intervals = data_value['high_activity_intervals']
low_activity_intervals = str(data_value['low_activity_intervals'])

# If the timestamp of the funf data is 15 minutes(900 seconds) prior from the
point when the survey data is collected, then we add up the
high_activity_intervals value

if (survey_time - data_time) >= 0 and (survey_time - data_time) <= 900:
      [survey_value,temp_high_activity_intervals]
```

```
temp_high_activity_intervals = temp_high_activity_intervals +
high_activity_intervals

final_data[survey_time] =
[survey_value,temp_high_activity_intervals]
```

3.3.2 Bluetooth

Bluetooth sensor detects the Bluetooth devices within range from your mobile phone. The default period is 300 seconds.

Example Data:

```
{
u'android-bluetooth-device-extra-device':
        {
        u'maddress': u'CB:6E:44:D1:18:1D'
        },
u'timestamp': 1386098113.684,
u'android-bluetooth-device-extra-rssi': -91,
u'android-bluetooth-device-extra-class':
        {
        u'mclass': 7936
        },
        u'android-bluetooth-device-extra-name': u'One'
}
```

Each device has its unique media access control address (MAC address). MAC address is represented as 'maddress' in the data and is a combination of 12 alphanumeric characters that can contain numbers from 0 to 9 and letters from A to F. [12] MAC address is an unique identifier assigned to network interfaces for communications on the physical network segment. [11] In Bluetooth.py, it sums up all the unique 'maddress' within the 15 minutes prior the survey is collected to calculate the number of unique devices around participant's mobile phone.

Listing 3.3.2: Generating the Formatted Data from the Bluetooth Probe

```
#This will give me the value of the data from funf
```

```
data_value = ast.literal_eval(j[1])
pre_maddress = data_value['android-bluetooth-device-extra-device']
maddress = str(pre_maddress['maddress'])

# If the funf data is 15 minutes(900 seconds) within the survey data, then we
add up the unique 'maddress'

      if (survey_time - data_time) >=0 and (survey_time - data_time)
<=900:
      temp_set.add(maddress)
      final_data[survey_time] = [survey_value,len(temp_set)]
```

3.3.3 Call Logs

Call log sensor records the calls that are made by the device. All the sensitive information, such information, such as the name of the receiver and the phone number, is normalized and hashed consistently and can be compared to contacts on this device, or with other devices. [10] The default period is 300 seconds.

Example Data:

```
{
u'name':
      u'{
      "ONE_WAY_HASH":"8af6611da3b93cfc392fb216dc319a52f58daaf7"
      }',
u'numberlabel':
      u'{
      "ONE_WAY_HASH":""
      }',
u'numbertype':
      u'{
      "ONE_WAY_HASH":"77de68daecd823babbb58edb1c8e14d7106e83bb"
      }',
u'number':
   u'{
      "ONE_WAY_HASH":"d72a18dc360226be5bd0bf96cb94eefb1fcc3374"
      }',
u'date': 1380571060066L,
u'timestamp': 1380571060.066,
u'duration': 102,
u'_id': 12,
u'type': 1
}
```

As shown in the example, all the sensitive information is hashed. " u'duration' " represents the duration of the phone call in seconds. CallLog.py sums up the all the phone call duration during the 15 minutes time frame prior the survey is submitted by the participants.

Listing 3.3.3: Generating the Formatted Data from the Call Log Probe

```
#This will give me the value of the data from funf
data_value = ast.literal_eval(j[1])
call_duration = data_value['duration']

#This help us prevent collecting the data into final_data when there is no
matching between the survey and the data

    if (survey_time - data_time) >= 0 and (survey_time - data_time) <=
    900:

    call_count = call_count + call_duration
   final_data[survey_time] = [survey_value,call_count]

    elif (survey_time - data_time) < 0:
    break
```

3.3.4 SMS

SMS sensor records messages sent and received by the device using SMS and all the sensitive data are hashed for users' privacy. [10] The default period is 3600 seconds.

Example Data:

```
{
  u'body':
  u'{
        "ONE_WAY_HASH":"ca6ca17b1b8bc4b9e134f498f97a406593824b90"
  }',
  u'service_center': u'+12404492162',
  u'locked': False,
  u'person': u'{
        "ONE_WAY_HASH":"9f682df245668969bbcd5395bdc2882591eeecde"
             }',
  u'read': True,
  u'timestamp': 1380632699.765,
  u'reply_path_present': False,
    u'thread_id': 10,
    u'status': -1,
    u'address':u'{
          "ONE_WAY_HASH":"eff26f4c007e64b1311e439762bfbc5ac8371e86"
                 }',
    u'date': 1380632699765L,
    u'protocol': 0,
    u'type': 1,
    u'subject': u'{
          "ONE_WAY_HASH":""
                 }'
    }
```

The data have hashed information for the content of the messages, name of the sender, and the address of the sender. " u'read' " indicates whether the mobile phone user has read the text message, and "u'reply_path " indicates whether the user has replied to the text message. Sms.py calculates the number of unique messages that have been read by the participants.

Listing 3.3.4: Generating the Formatted Data from the SMS Probe

```
#This will give the time stamp of the survey, it doesn't need to be
converted using ast.literal_eval.
#If used, it will give error.

survey_time = i[0]

#This will give me the value from Survey

survey_value = ast.literal_eval(i[1])

#This is the temporary value for the count of 'screen_on' == True

sms_count = 0

for j in data:

#This will give me the time stamp of the data from funf

data_time = j[0]

#This will give me the value of the data from funf

data_value = ast.literal_eval(j[1])
sms_read = data_value['read']
```

```
#This help us prevent collecting the data into final_data when there is no
matching between the survey and the data
# If the funf data is 15 minutes(900 seconds) within the survey data, then
we add up the high_activity_intervals value

     if (survey_time - data_time) >= 0 and (survey_time - data_time) <=
900:

#We only collect the data if the user has read the data
     if sms_read is True:

             sms_count = sms_count + 1
             final_data[survey_time] = [survey_value,sms_count]
```

3.3.5 Screen

Screen sensor records when the screen turns off and on. [10]

Example data:

```
{
     u'timestamp': 1386022720.941,
     u'screen_on': False
}
```

Screen.py calculates all number of records with the screen to be on within the 15 minutes prior from the survey is collected.

Listing 3.3.5: Generating the Formatted Data from the Screen Probe

```
#This will give me the value of the data from funf

data_value = ast.literal_eval(j[1])
screen_value = data_value['screen_on']

#This help us prevent collecting the data into final_data when there is no
matching between the survey and the data
# If the funf data is 15 minutes(900 seconds) within the survey data, then
we add up the high_activity_intervals value

     if (survey_time - data_time) >= 0 and (survey_time - data_time) <=
900:
#We only collect the data if the scr
een is on.

     print 'Found'
     screen_count = screen_count + 1
     final_data[survey_time] = [survey_value,screen_count]
```

```
# We remove funf data that we have visited and added to the list. In this
way, when we loop a new i, we don't have to visit
# the data that we have visited before.
# If the funf data is beyond 15 minutes of the survey data, then we stop
the loop and choose the next survey data

    elif (survey_time - data_time) < 0:

        print 'not found'
```

3.3.6 Wi-Fi

Wi-Fi sensor records all the available Wi-Fi access points detected by the mobile phone.

Example data:

```
{
        u'distancesdcm': -1,
        u'ssid': u'MIT GUEST',
        u'bssid': u'00:26:cb:f4:83:ee',
        u'level': -87,
        u'timestamp': 1386098103.864,
        u'capabilities': u'[ESS]',
        u'distancecm': -1,
        u'frequency': 5320,
        u'wifissid': {
                u'octets': {
                        u'count': 9,
                        u'buf': [77, 73, 84, 32, 71, 85, 69, 83, 84, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
                            }
                        }
}
```

Every Wi-Fi access point has its unique Service Set Identifier (SSID), which is a case sensitive, 32 alphanumeric character identifier attached to the header of packets sent over a wireless local area network that acts as a password when a mobile device tries to connect to the basic service set. [13] The system is designed to scan the Wi-Fi points around the device every 15 minutes. We would like to use the Wi-Fi points to estimate the relative movement of the participants. Therefore, Wifi.py takes the probe data from the Probe Data Package and the following equation to calculate the Wi-Fi score:

$$\text{Wi-Fi score} = \frac{(\text{Previous wifi SSID set}) \cup (\text{Current wifi SSID set})}{(\text{Previous wifi SSID set}) \cap (\text{Current wifi SSID set})}$$

Wifi.py first generate a dictionary of Wi-Fi score and its matching timestamp. Then it takes the data from the Survey Package, find all the Wi-Fi data with timestamp that's within 15 minutes prior from the point when the survey is submitted and generates the final dictionary with survey answers, Wi-Fi score and their corresponding timestamp.

Listing 3.3.6: Generating the Formatted Data from the Wifi Probe

```
#This will give me the value of the data from funf

data_value = ulta[key]

# If the funf data is 15 minutes(900 seconds) within the survey data, then
we add up the high_activity_intervals value

      if (survey_time - data_time) >= 0 and (survey_time - data_time) <=
900:
      final_data[survey_time] = [survey_value, data_value]
```

3.4 Data Graphing

Once the Sensor Script generates the Formatted Data, the Graphing Script takes the data and creates the Relationship Graphs between the sensor data and the survey answers.

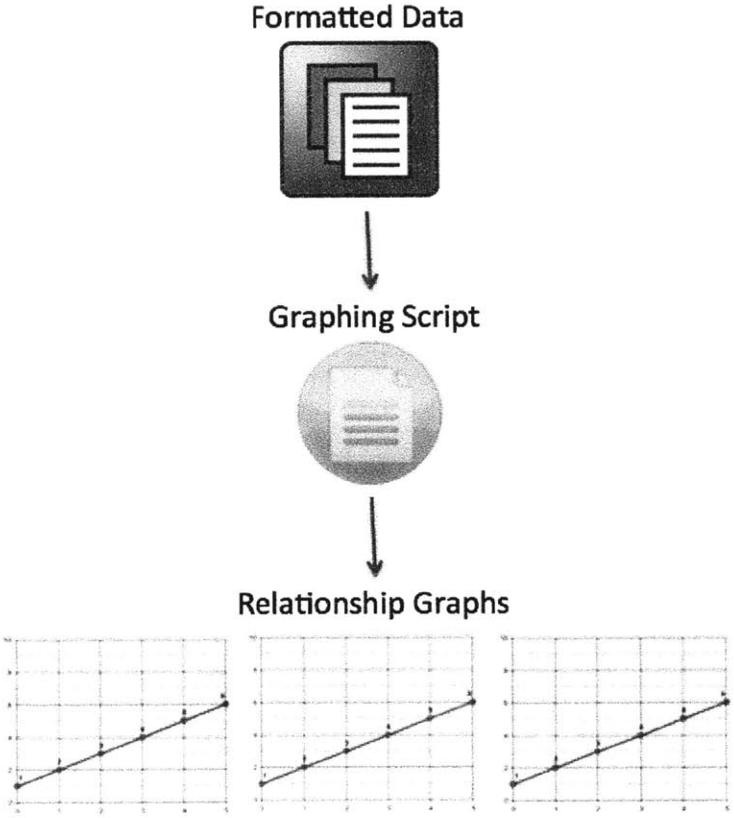


Figure 3.3: Overview of how the Relationships Graphs are generated – the Graphing Script takes the Formatted Data generated by the Sensor Script to create corresponding Relationship Graphs of the probe data against the data of socialization, physical activity and mental concentration.

For instance, the Graphing Script takes the data of activity probe against the data of the survey answers regarding participant's social, activity and physical levels (See Figure 3.4).

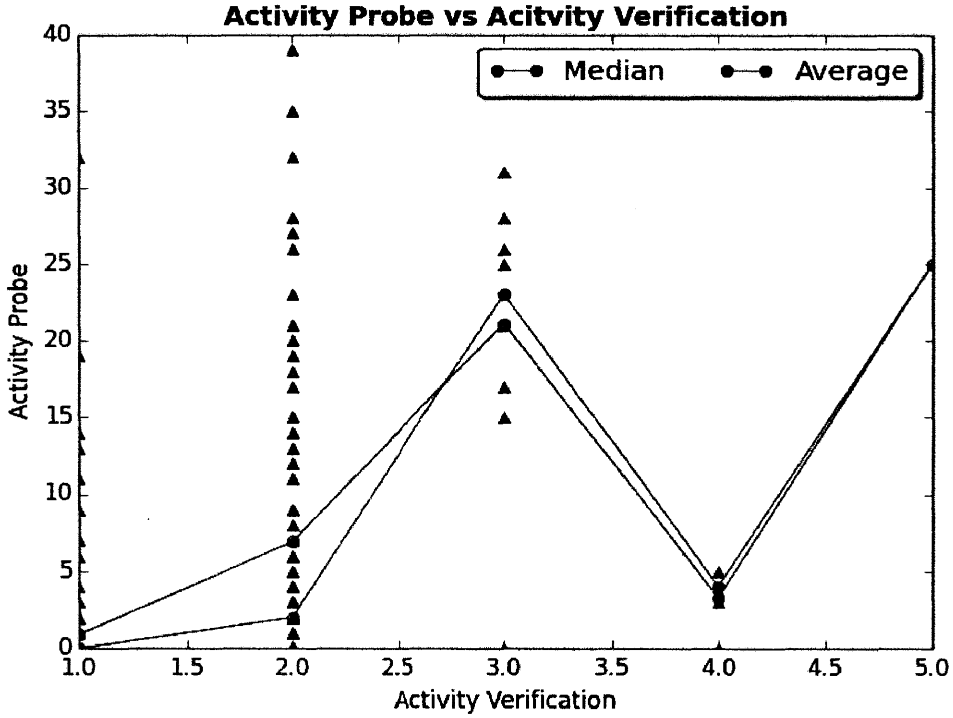


Figure 3.4: The Relationship Graph of the data of Activity Probe against the data of physical activity.

Chapter 4

Conclusion and Future Work

Conclusion The data analysis system allows researchers to generate Relationship Graphs in a cost effective and efficient manner. The system provides a framework for easily using the scripts to generate formatted data that are flexible enough for researchers to manipulate the data and create graphs other than relationship graphs, such as graphs that plot mobile data against its corresponding timestamps. For the actual results of the data that we have collected (see Appendix A), there are some graphs that show a linear relationship. However, due to the fact that the number of samples size is quite small, we are unable to conclude that there are any solid relationships between the mobile phone data and the survey answers collected from our participants.

Future Work In this experiments, participants were required to answer two surveys per day over several months. The challenge of this design is that we were unable to make sure our users follow this instruction everyday. In fact, there were a lot of corrupted and missing data resulting from users skipping surveys from time to time. The other challenge is that since this experiment is a pilot test, the sample size we collected was too small to conclude that there are any relationships between the mobile phone data and the survey answers. For future work, in addition of having a large participant group, giving participants certain incentive or rewards to answer the surveys would help curate the overall collected data.

One of the other challenges regarding the quality of the data is that the data of self-answering survey is very noisy. However, we might be able to get a more accurate observation of a person from getting other people to answer questions regarding this user. In the future, we can utilize the Bluetooth feature on the mobile phone to detect other mobile phones around our participants. When other mobile phones are detected, the system could send surveys to these mobile phone users and ask them survey questions regarding this user. For example, when your mobile phone detected your friend's phone around you, your friend would get a survey asking about how you are doing today, do you look happy or depressed. Using human in addition to mobile phone as a medium might be more accurate than the result from the self-reported data.

Bibliography

[1] Eagle, N., A. Pentland, and D. Lazer. "From the Cover: Inferring Friendship Network Structure by Using Mobile Phone Data." *Proceedings of the National Academy of Sciences* 106.36 (2009): 15274-5278. Web.

[2] Eagle, Nathan, and Alex (Sandy) Pentland. "Reality Mining: Sensing Complex Social Systems." *Personal and Ubiquitous Computing* 10.4 (2006): 255-68. Web.

[3] "Home | Www.css.gmu.edu." *Home | Www.css.gmu.edu.* Web. 07 Aug. 2014.

[4] Lazer, D., A. Pentland, L. Adamic, S. Aral, A.-L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, T. Jebara, G. King, M. Macy, D. Roy, and M. Van Alstyne. "SOCIAL SCIENCE: Computational Social Science." *Science* 323.5915 (2009): 721-23. Web.

[5] "Sensors Overview." *Android Developers*. Web. 07 Aug. 2014.

[6] "Smartphones in Use Surpass 1 Billion, Will Double by 2015." *Bloomberg.com.* Bloomberg. Web. 07 Aug. 2014.

[7] Madan, A., et al., Social sensing for epidemiological behavior change, in Ubiquitous Computing/Handheld and Ubiquitous Computing. 2010. p. 291-300.

[8] Y. Altshuler, N. Aharony, M. Fire, Y. Elovici, A. Pentland (2011). "Incremental Learning with Accuracy Prediction of Social and Individual Properties from Mobile-Phone Data". arXiv:1111.4645

[9] Aharony, Nadav, Wei Pan, Cory Ip, Inas Khayal, and Alex Pentland. "Social FMRI: Investigating and Shaping Social Mechanisms in the Real World." *Pervasive and Mobile Computing* 7.6 (2011): 643-59. Web.

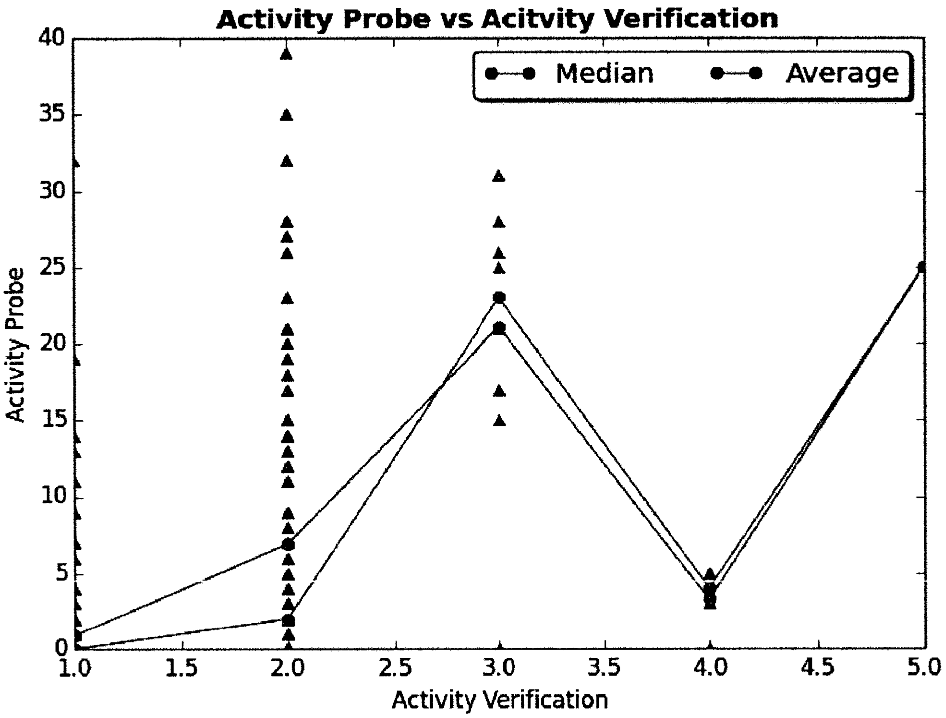
[10] "Get Started -." *Funf-open-sensing-framework - Android-based Framework for Phone-based Sensing and Data-collection.* Web. 07 Aug. 2014.

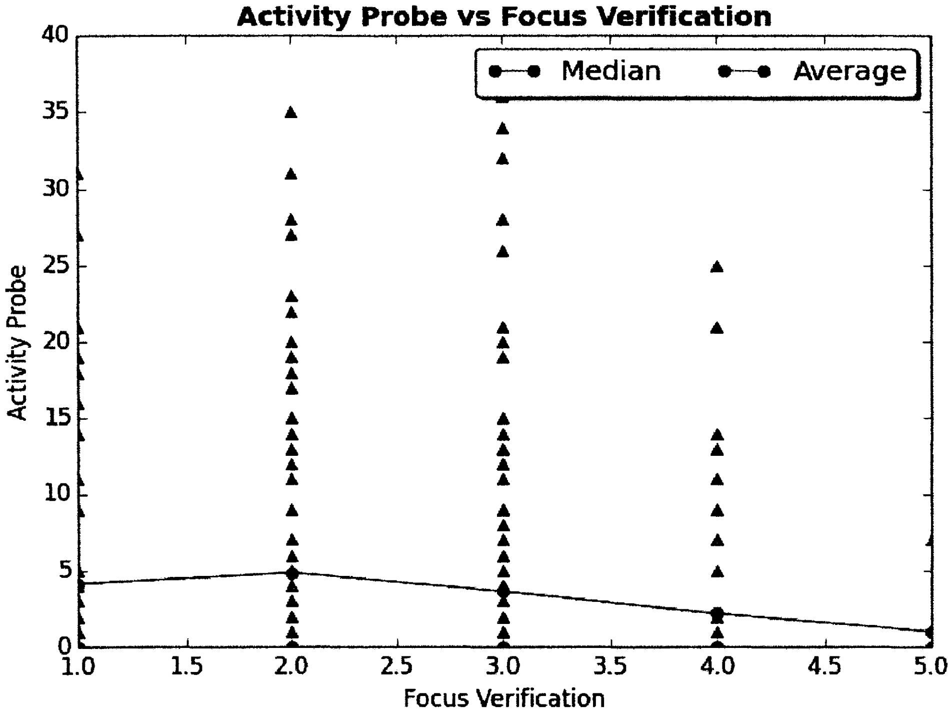
[11] "MAC Address." *Wikipedia.* Wikimedia Foundation, 22 July 2014. Web. 06 Aug. 2014.

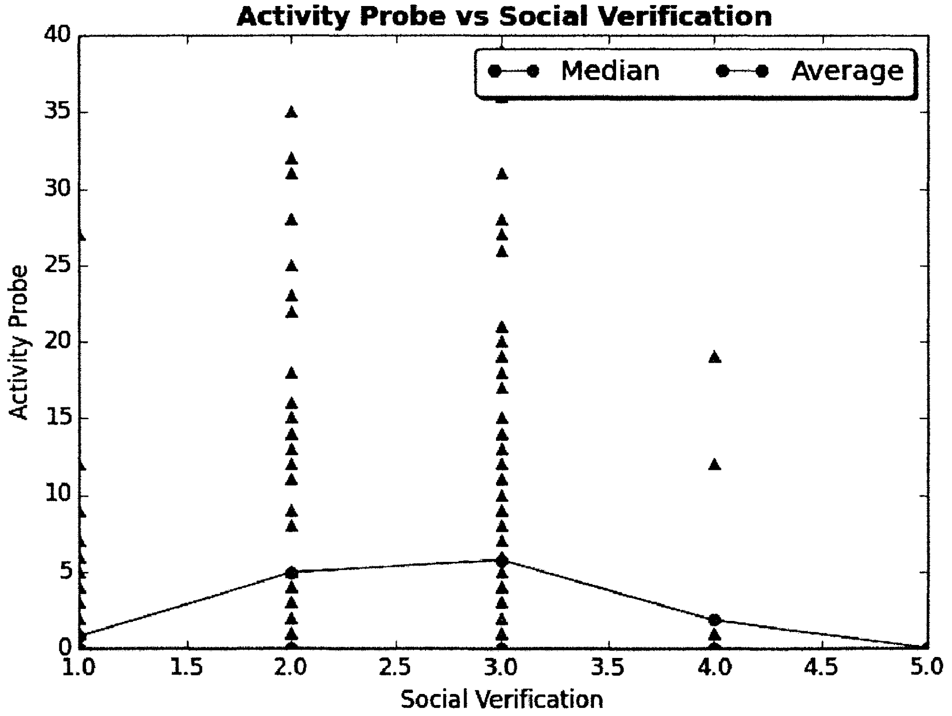
[12] "How to Find a Bluetooth Address (BD_ADDR)." *How to Find a Bluetooth Address (BD_ADDR).* Web. 07 Aug. 2014.

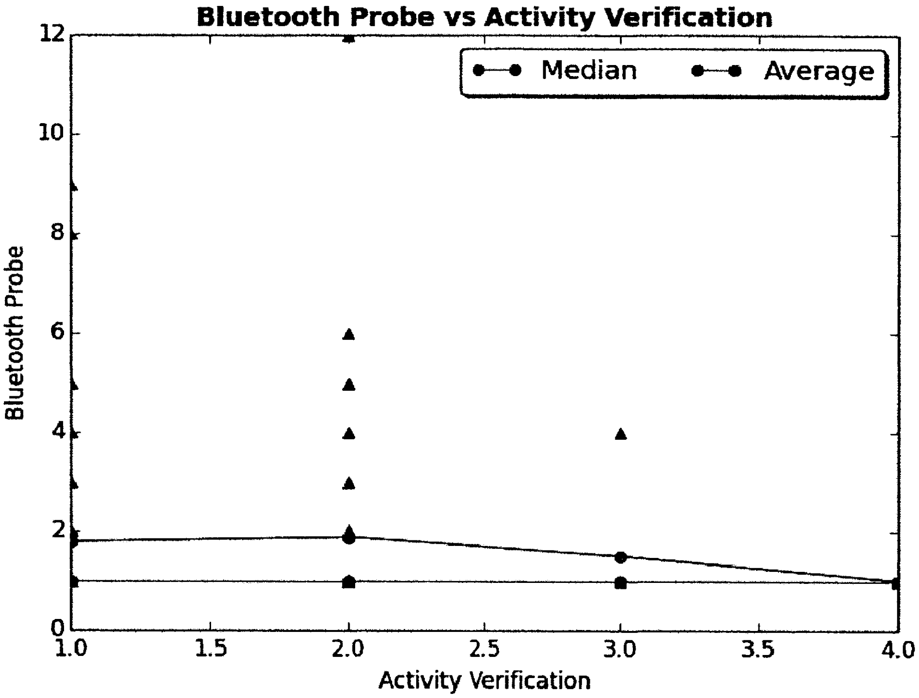
[13] "SSID - Service Set Identifier." *What Is Service Set Identifider (SSID)? Webopedia.* Web. 07 Aug. 2014.

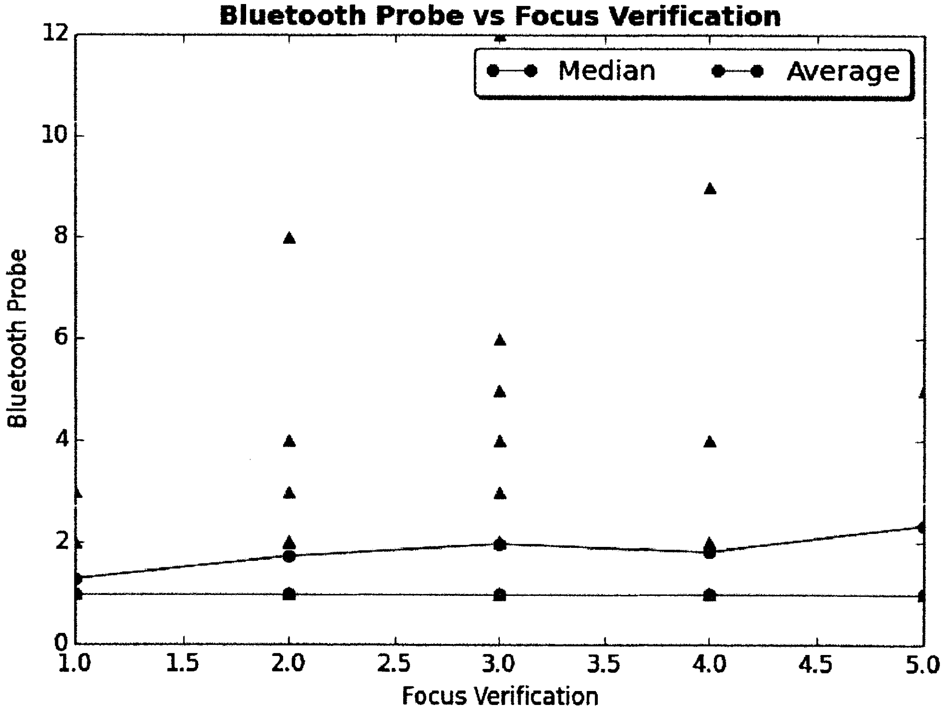
Appendix A

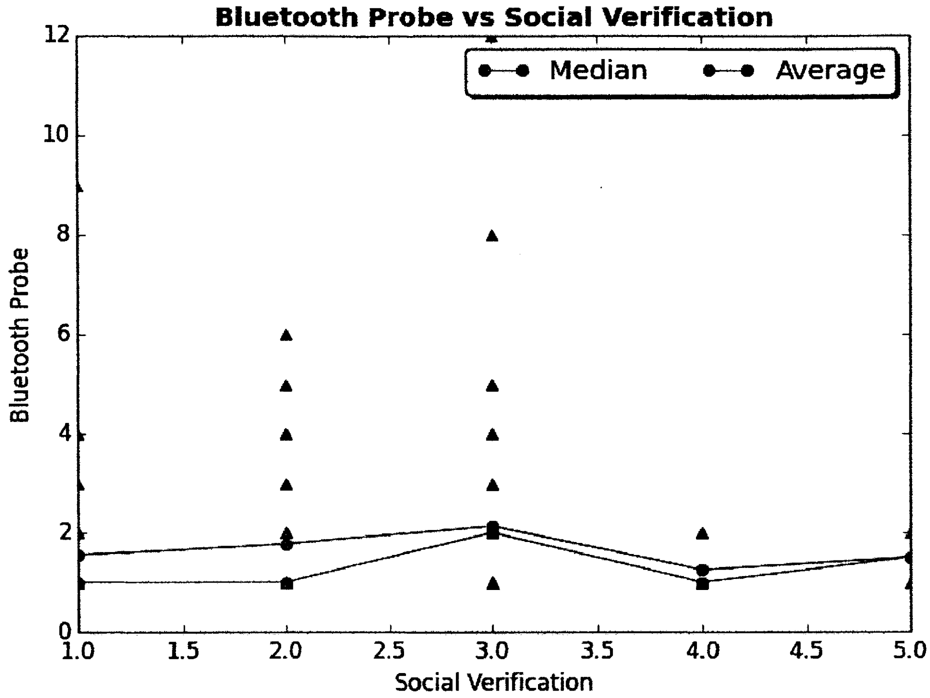

Activity Probe vs Acitvity Verification
Median
Average
Activity Probe
0
5
10
15
20
25
30
35
40
Activity Verification
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

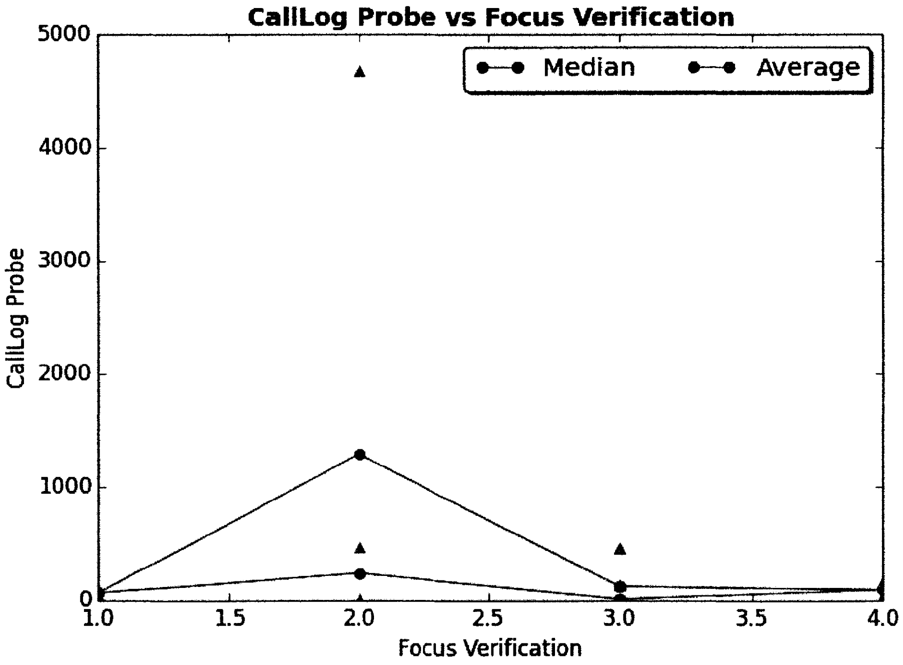

Activity Probe vs Focus Verification
Median
Average
Activity Probe
Focus Verification
0
5
10
15
20
25
30
35
40
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

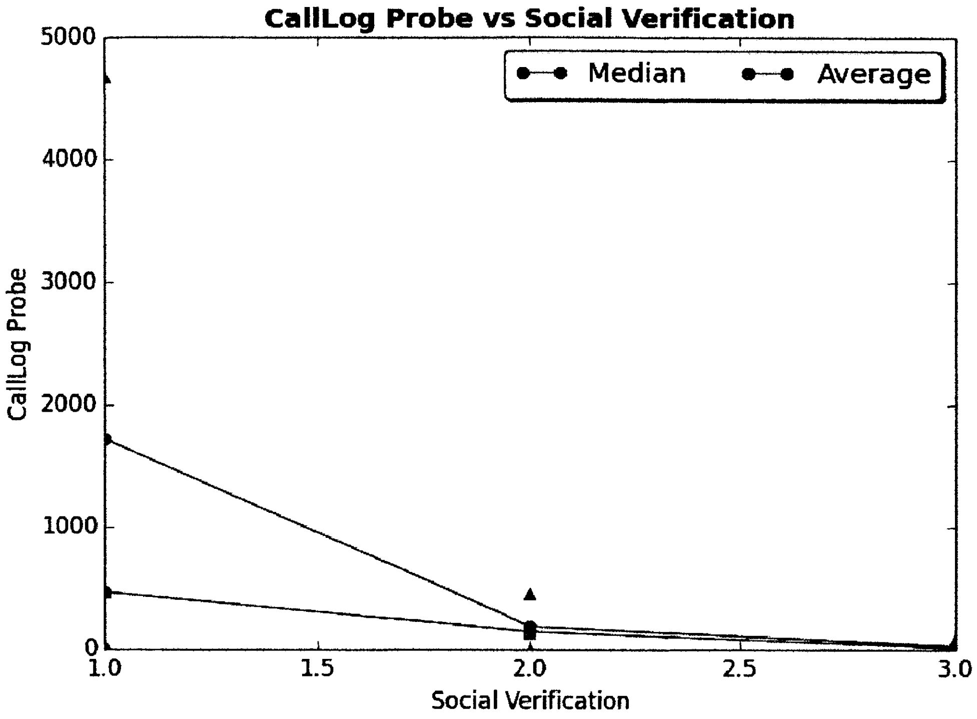

Activity Probe vs Social Verification
Median
Average
Activity Probe
Social Verification
0
5
10
15
20
25
30
35
40
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

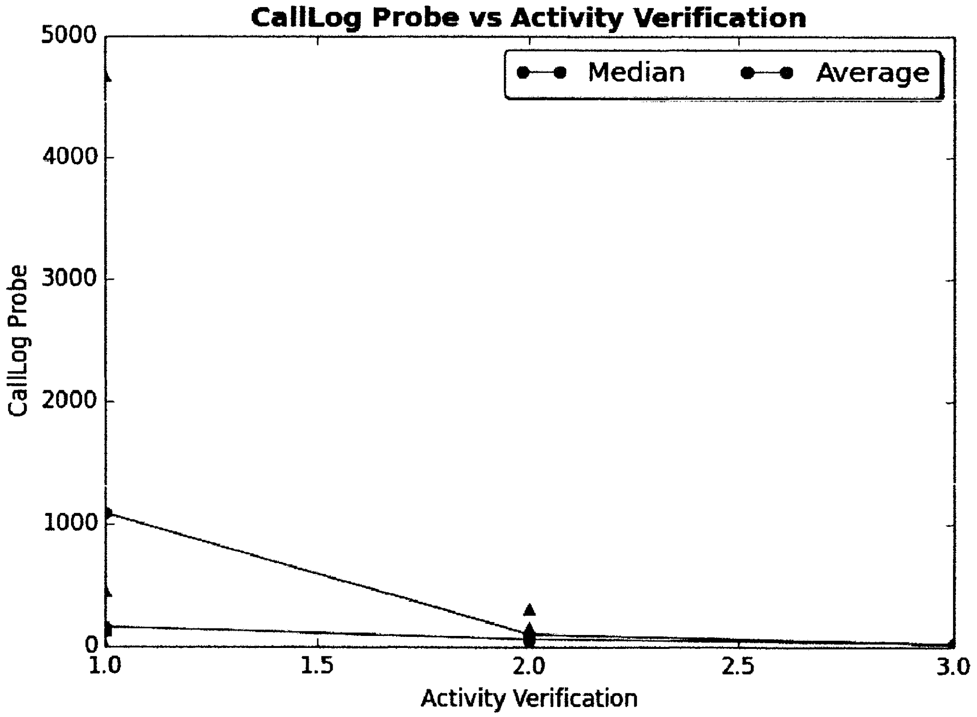

Bluetooth Probe vs Activity Verification
Median
Average
Bluetooth Probe
Activity Verification
0
2
4
6
8
10
12
1.0
1.5
2.0
2.5
3.0
3.5
4.0

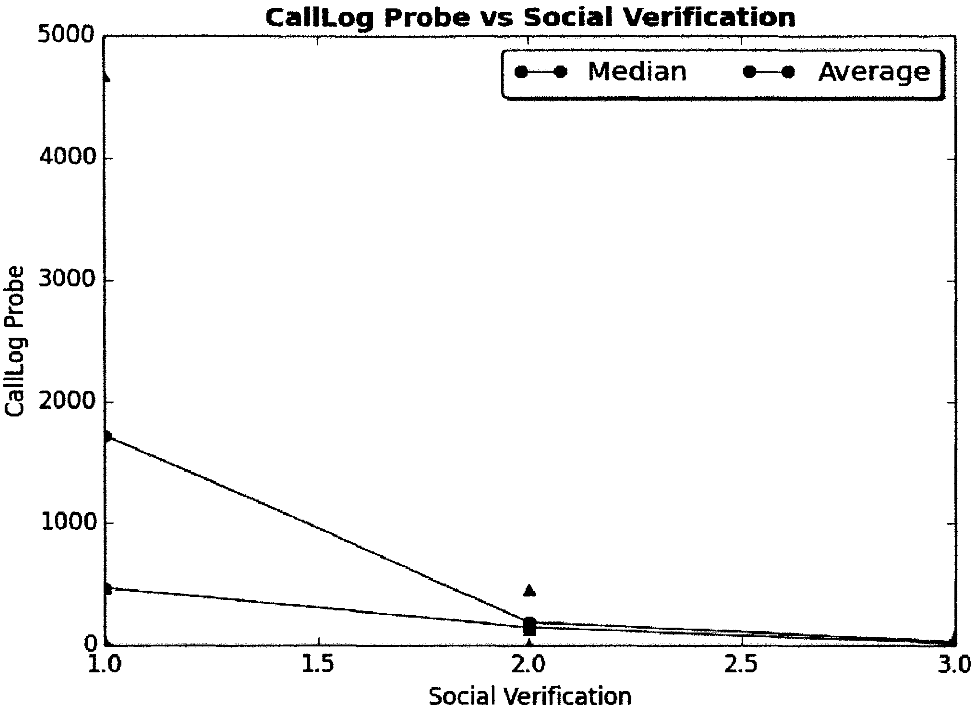

Bluetooth Probe vs Focus Verification
Median
Average
Bluetooth Probe
Focus Verification
0
2
4
6
8
10
12
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

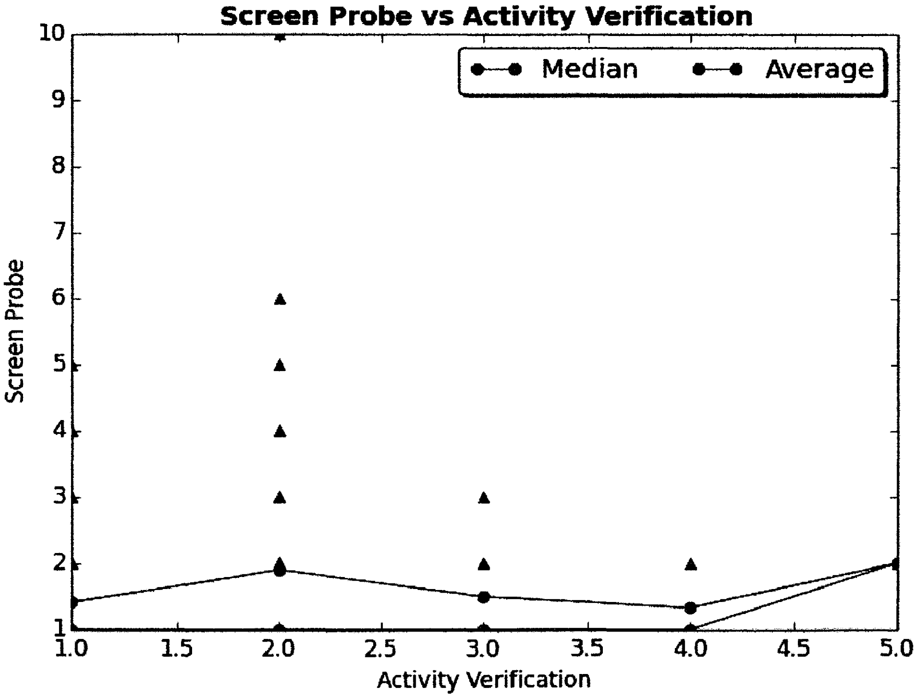

Bluetooth Probe vs Social Verification
Median
Average
Bluetooth Probe
Social Verification
0
2
4
6
8
10
12
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

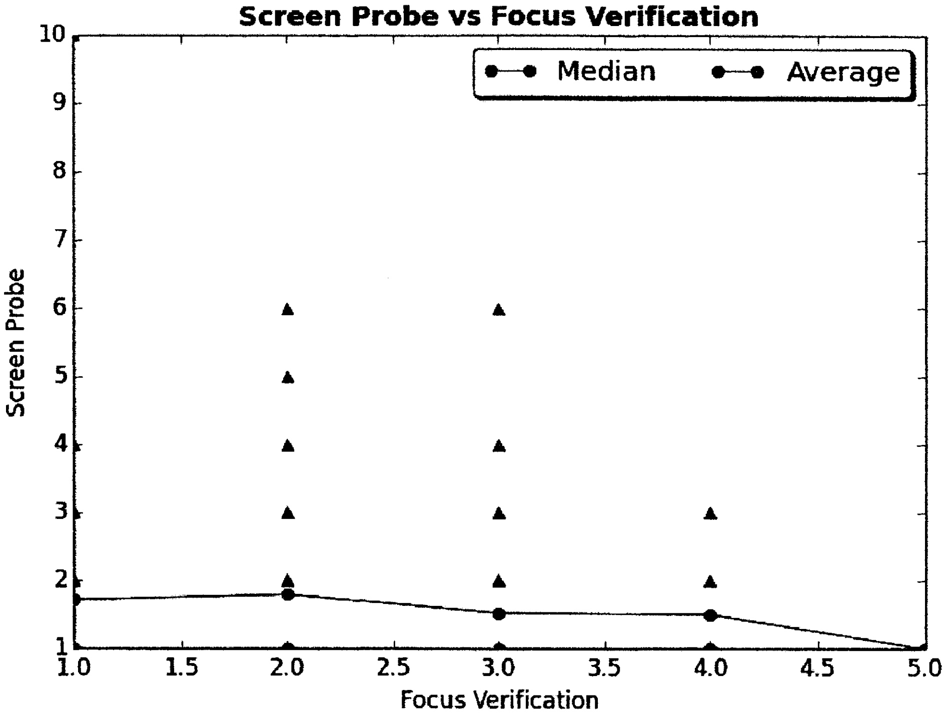

CallLog Probe vs Focus Verification
Median
Average
CallLog Probe
Focus Verification
0
1000
2000
3000
4000
5000
1.0
1.5
2.0
2.5
3.0
3.5
4.0

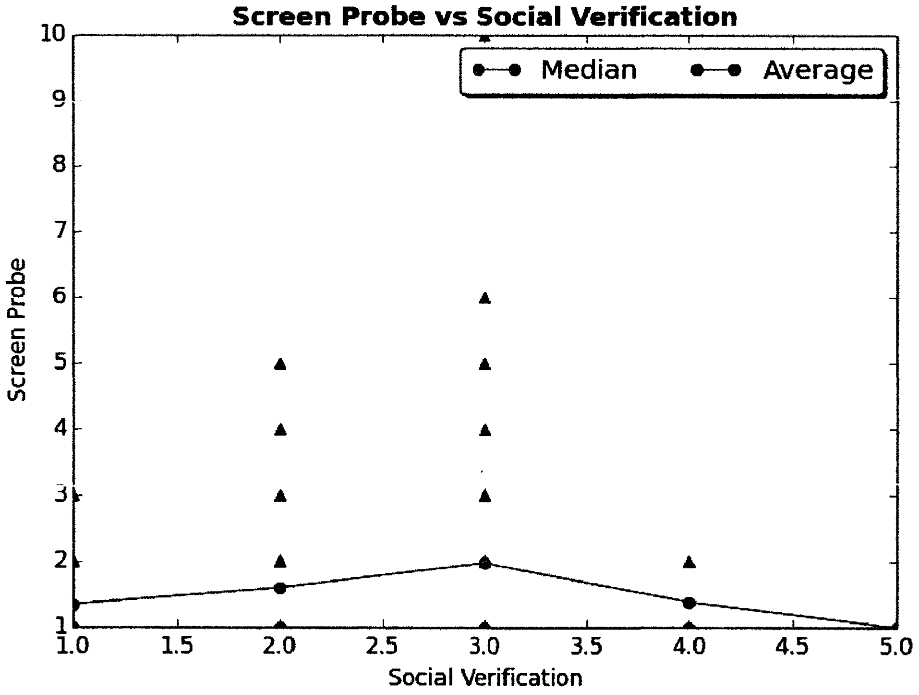

CallLog Probe vs Social Verification
5000
4000
3000
2000
1000
0
CallLog Probe
1.0
1.5
2.0
2.5
3.0
Social Verification
Median
Average

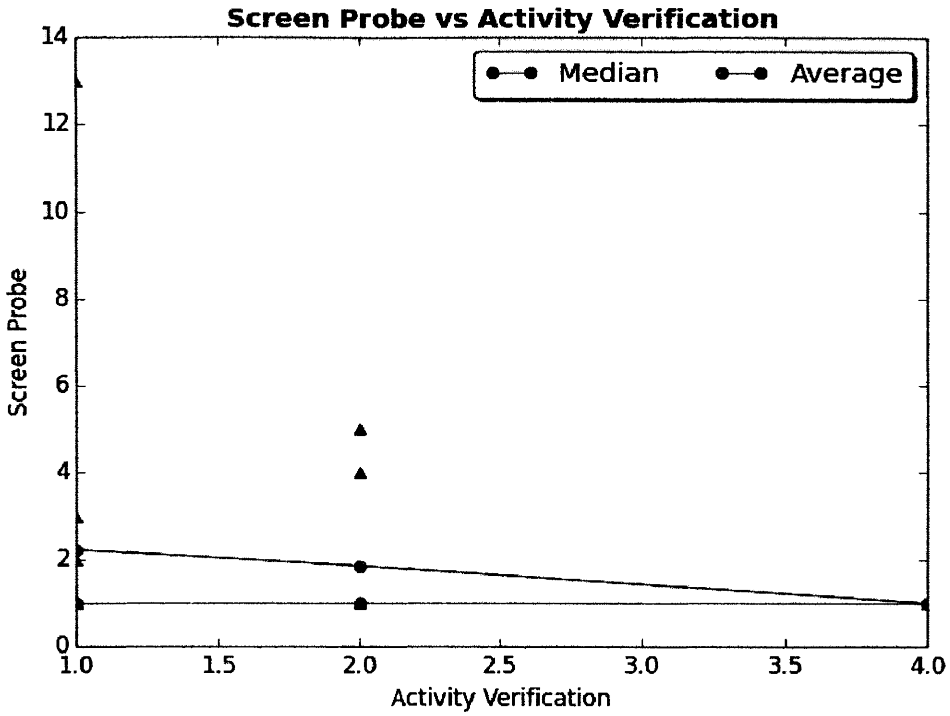

CallLog Probe vs Activity Verification
5000
4000
3000
2000
1000
0
CallLog Probe
1.0
1.5
2.0
2.5
3.0
Activity Verification
Median
Average

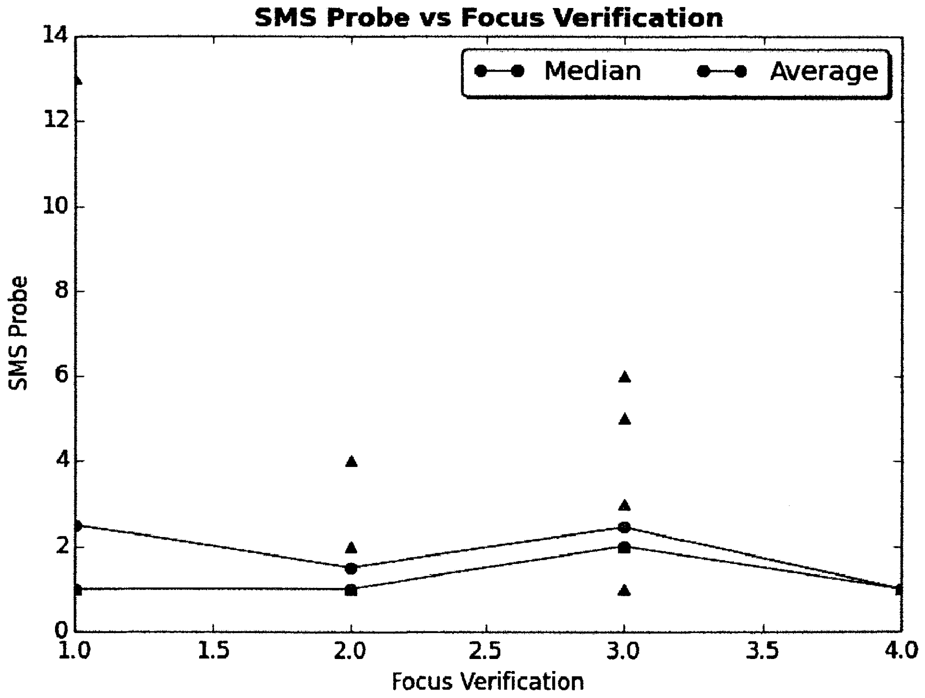

CallLog Probe vs Social Verification
Median
Average
CallLog Probe
5000
4000
3000
2000
1000
0
Social Verification
1.0
1.5
2.0
2.5
3.0

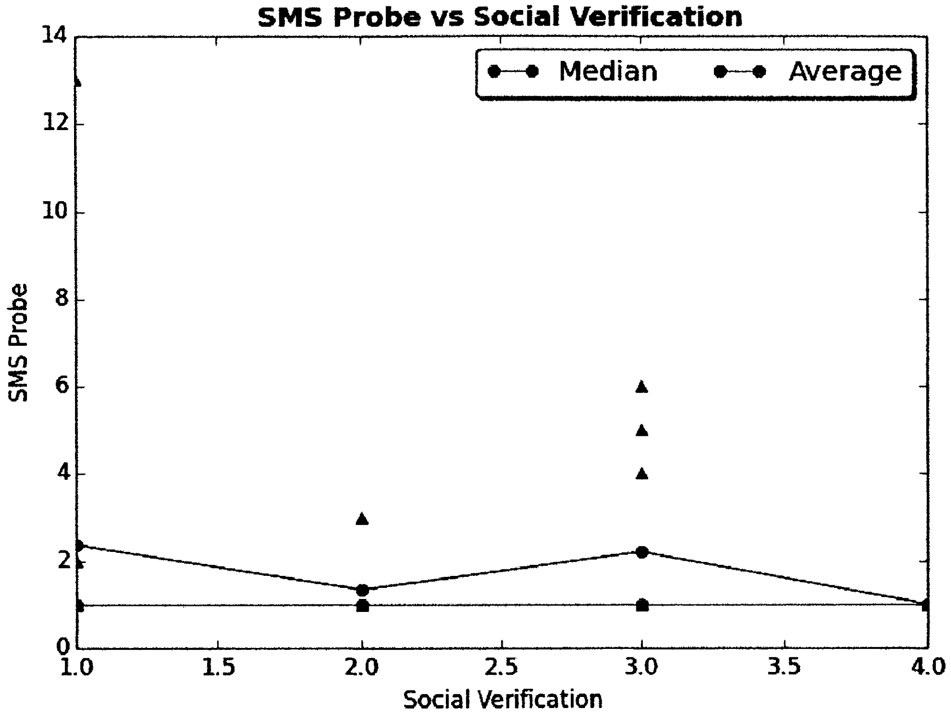

Screen Probe vs Activity Verification
Median
Average
Screen Probe
10
9
8
7
6
5
4
3
2
1
Activity Verification
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Screen Probe vs Focus Verification
Median
Average
Screen Probe
Focus Verification
1
2
3
4
5
6
7
8
9
10
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Screen Probe vs Social Verification
Median
Average
Screen Probe
Social Verification
1
2
3
4
5
6
7
8
9
10
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0


Screen Probe vs Activity Verification
Median
Average
Screen Probe
Activity Verification
0
2
4
6
8
10
12
14
1.0
1.5
2.0
2.5
3.0
3.5
4.0


SMS Probe vs Focus Verification
Median
Average
SMS Probe
Focus Verification
0
2
4
6
8
10
12
14
1.0
1.5
2.0
2.5
3.0
3.5
4.0


SMS Probe vs Social Verification
Median
Average
SMS Probe
Social Verification
0
2
4
6
8
10
12
14
1.0
1.5
2.0
2.5
3.0
3.5
4.0