

University of North Carolina Charlotte
Data Science and Business Analytics

Big Data Challenge 2 Project
Project Report

Vijay S Chauhan, Sunil Kuman MN and Soumya Boyanapalli

DSBA 6160 - Big Data Design, Storage and Provenance

Professor Liyue Fan

May 5, 2020

Introduction	2
Configuration of Sensus App	2
Setting up (AWS) S3 Bucket	6
IAM Account	7
Visualization	8
Battery Level (polling) Visualization	8
Interpretation	8
Compass Heading (listening) Visualization	9
Interpretation	9
Acceleration (listening) Visualization	10
Interpretation	10
GPS Location (polling) Visualization	12
Interpretation	12
Bluetooth Visualization	14
Interpretation	14
Challenges	16
Sensus App Challenges	16
Bluetooth Data collection Challenges	19
AWS & S3 Bucket Challenges	19
SensusPy Challenges	19
data_retrieval.py	20
data_operations.py	20
plot.py	20
Applying Differential Privacy to our data	21
Conclusion	23



Introduction

Throughout the process of Challenge 2, we as a team were able to expand our knowledge in many different topics. Topics such as data collection, data storage, data analysis and much more. Though there were many difficult obstacles we had to surpass, at the end we reached our goal. Our goal of collecting a variety of data by using the sensus up, storing that data in a AWS S3 bucket, combining the stored data using SensusPy and finally being able to analyze the data using visualization tools. Our study's duration lasted from April 13, 2020 till May 5, 2020. The biggest skill that helped us achieve the end goal was teamwork. Though for parts of the project we worked individually at the end of the day we had to come together to achieve every goal step.

Configuration of Sensus App

For data collection, we used the Sensus app to collect data individually. There were 3 mobiles used for this project; 2 were iOS and 1 was an Android. In the Sensus app each of us used the same setting so we would be able to collect information on the same features but also collect different data for those features overall. Our collection of data lasted for a total of 14 days, the start date was April 17, 2020 and the end date was May 2, 2020. Below are screenshots of how some of our app settings looked like:

11:48

< Your Studies Protocol

Id:*

f0aa66f5-83b5-492b-a52f-fc86597e9248

Name:*

Questionnaire 3

Description:

Testing for Big Data Design Class Project

Start Immediately:

☒

Start Date:

4/17/2020

Start Time:

4:00 PM

Continue Indefinitely:

☒

End Date:

5/2/2020

End Time:

4:00 PM

Participation Horizon (Days):*

14

Contact Email:

sboyana@uncc.edu

Groupable:

☒

11:48

< Your Studies Protocol

Groupable:

☒

Reward Threshold:

0.95

GPS - Desired Accuracy (Meters):*

25

GPS - Minimum Time Delay (MS):*

5000

GPS - Minimum Distance Delay (Meters):*

50

Variables:

(iOS) GPS - Pause Location Updates:

☒

(iOS) GPS - Pause Activity Type:

Other

(iOS) GPS - Significant Changes:

☒

(iOS) GPS - Defer Location Updates:

☐

(iOS) GPS - Deferral Distance (Meters):

500

(iOS) GPS - Deferral Time (Mins.):

5

11:49

< Your Studies Protocol

Alert Exclusion Windows:

Notifications in alert exclusion windows:

☒

Asymmetric Encryption Public Key:

Allow View Data:

☒

Allow View Status:

☐

Allow Submit Data:

☒

Allow Participation Scanning:

☒

Allow Copy:

☒

Allow Protocol Share:

☒

Allow Local Data Share:

☒

Allow ID Reset:

☐

Allow Tagging:

☒

11:49

< Your Studies Protocol

Available Tags:

Allow Pause:

☐

Allow Snooze:

☐

Max. Snooze (Mins.):

1440

Allow Test Push:

☒

Start Confirmation Mode:*

None

Push Notification Hub:

Push Notifications Shared Access Signature:

Compatibility:*

CrossPlatform

(Android) Display Participation:

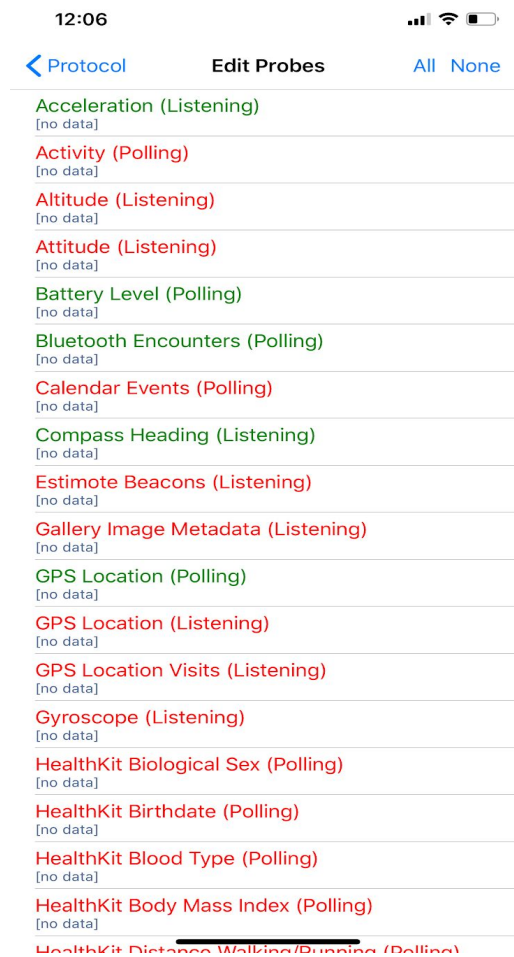
☒

(Android) Continue listening on AC power:

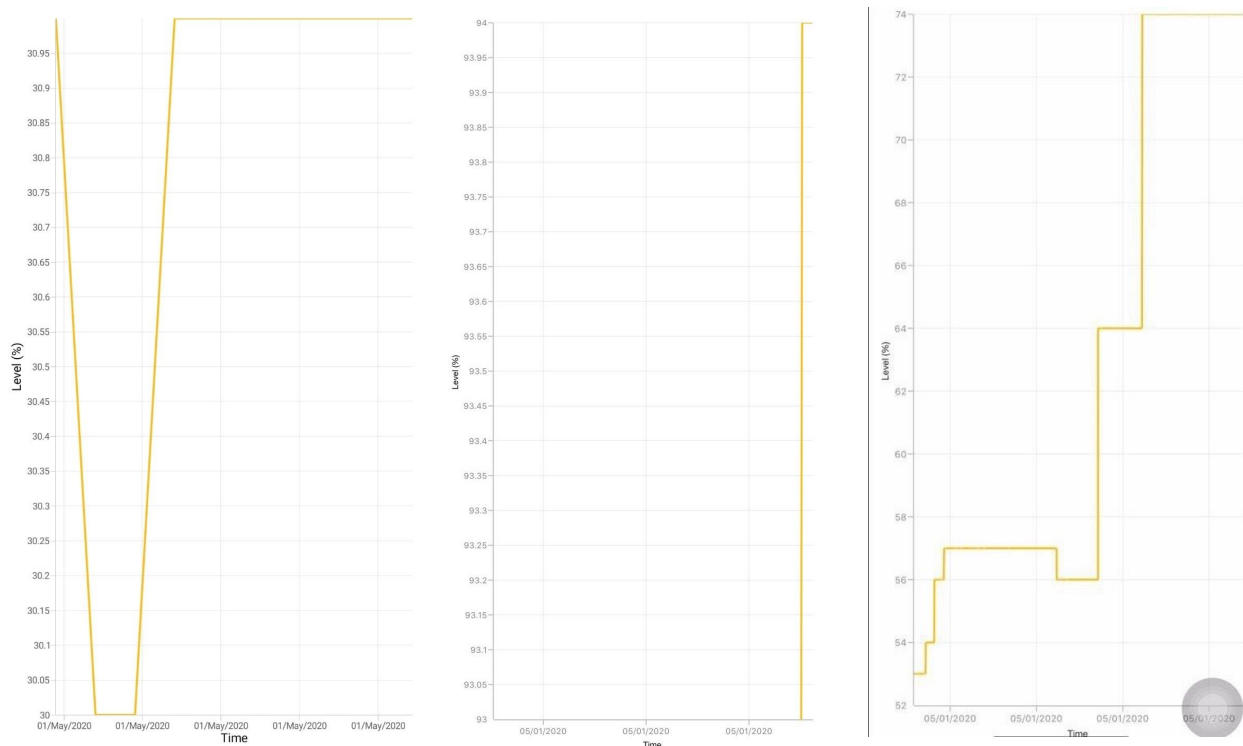
☐

[Copy Study Identifier](#)

In our app we used the required probes [Acceleration (listening), Battery Level (polling), Bluetooth Encounters (polling), Compass Heading (listening) and GPS Location (polling)] to collect data.



For some of the probes we were able to collect plots through the app. These plots showed our usage of that specific probe for a certain time period. For example below are three plots (one from each member in the team) for the Battery Level (polling) probe.



The plots above are plotted with the axis' Time vs Battery Percent Level. It can be seen from the first two plots' battery percentage leveled off at a constant number whereas the third plot's battery percentage rose quite a bit. The first plot stayed at a constant 30%, barley increasing. The second plot also barely increased (only rising by 1%) from 93% to 94%. Lastly the third plot increased from about 53% to 74%!

After setting up all of these features on the app we finally had to make sure that all our data was able to be stored in one safe place. To do this we created an AWS S3 bucket. Below is a picture of the the information we had to enter in to make sure that our data was stored in the same bucket. The information we provided on the app included; region, bucket name, folder name as well as the IAM Account ID.

2:29

< Protocol Remote Data Store

Amazon S3

Region:*

us-east-1

Bucket:*

team5-4be5ce38-fa05-41ae-ad04-429dfd3d4cc5

Folder:

Acceleration

IAM Account:*

AKIA3LREGMAI7GNFPNGN:5s+wHjyr7QEGLD2rlB...

Pinned Service URL:

Pinned Public Key:

Write Delay (MS):*

3600000

Write Timeout (Mins.):*

5

Write on Power Connect:

☒

Write on Wifi Connect:

☐

Require WiFi:

☒

Require Charging: ☐

Looking at the image above, it can be noticed that there is a label named 'folder'. For this label we had to identify the folder name that you want to store the data in. This folder is located in the S3 bucket in AWS. Our team created five different folders for the five different probes, each folder belonging to one probe.

Setting up (AWS) S3 Bucket

Using AWS we were able to create a S3 bucket. Like mentioned previously, this bucket was used to store all our data into the same place. To do this we had to configure AWS which gave us an IAM account. With this we were able to gain our personal account id, access key id and secret key to be able to login into our own personal IAM amounts, where we would be able to access the S3 bucket.

Account ID (12 digits) or account alias






780686221329

Access key ID : AKIA3LREGMAISTIYHPBL

secret key : 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

Having access to these gave everyone in the team the chance to utilize the shareable S3 bucket. Below is a picture of our S3 bucket as well as the five different folders that are located in the bucket (as previously mentioned).

	Name ▾	Region ▾	Access ▾	Bucket created ▲
<input type="radio"/>	team5-4be5ce38-fa05-41ae-ad04-429dfd3d4cc5	US East (N. Virginia) us-east-1	Not public	2020-04-21T02:39:27.000Z

<input type="checkbox"/>	Name ▾
<input type="checkbox"/>	 Acceleration
<input type="checkbox"/>	 BatteryLevel
<input type="checkbox"/>	 Bluetooth
<input type="checkbox"/>	 GPS
<input type="checkbox"/>	 SoumyaData_CompassHeading

[IAM Account](#)

Welcome to Identity and Access Management

IAM users sign-in link:

<https://780686221329.signin.aws.amazon.com/console> 

[Customize](#)

IAM Resources

[Users: 4](#)

[Roles: 6](#)






[Groups: 1](#)

[Identity Providers: 0](#)

[Customer Managed Policies: 0](#)

Security Status

 3 out of 5 complete.

	Delete your root access keys	▼
	Activate MFA on your root account	▼
	Create individual IAM users	▼
	Use groups to assign permissions	▼
	Apply an IAM password policy	▼

Add user

Delete user

Find users by username or access key

Showing 4 results

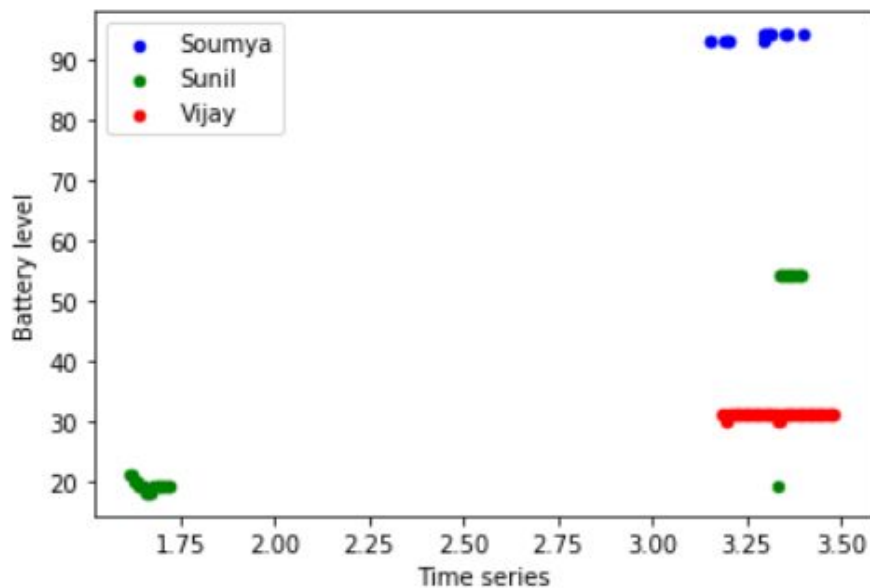
<input type="checkbox"/>	User name	Groups	Access key age	Password age	Last activity	MFA
<input type="checkbox"/>	smumudin	None	<div><div></div>10 days</div>	13 days	Today	Not enabled
<input type="checkbox"/>	sowmyab	None	<div><div></div>13 days</div>	12 days	Today	Not enabled
<input type="checkbox"/>	team5-4be5ce38-fa05-41ae-...ad04-429dfd3d4cc5-device-group	team5-4be5ce38-fa05-41ae-ad04-429dfd3d4cc5-device-group	<div><div></div>13 days</div>	None	Today	Not enabled
<input type="checkbox"/>	vijay	None	<div><div></div>3 days</div>	3 days	Today	Not enabled

Visualization

After collecting data through the Sensus app and storing them in AWS, we were able to download our member's JSON files data from the S3 bucket. After downloading the JSON files we were able to combine them together and pre-process the files using SensusPy. We then used some python code to make visualizations for each of the probes used in the app. The following contains visualizations for Acceleration (listening), Battery Level (polling), Bluetooth Encounters (polling), Compass Heading (listening) and GPS Location (polling).

Each of the plots below plot the data results for each of the three members in the group. Each member had a unique color that correlates with their data presented. Soumya has color blue, Sunil has color green and Vijay has color red. These colors for each individual are consistent throughout the plots except for the GPS plot and Bluetooth plot. For the GPS plot however, Soumya has color green, Sunil has color blue and Vijay has color red.

Battery Level (polling) Visualization

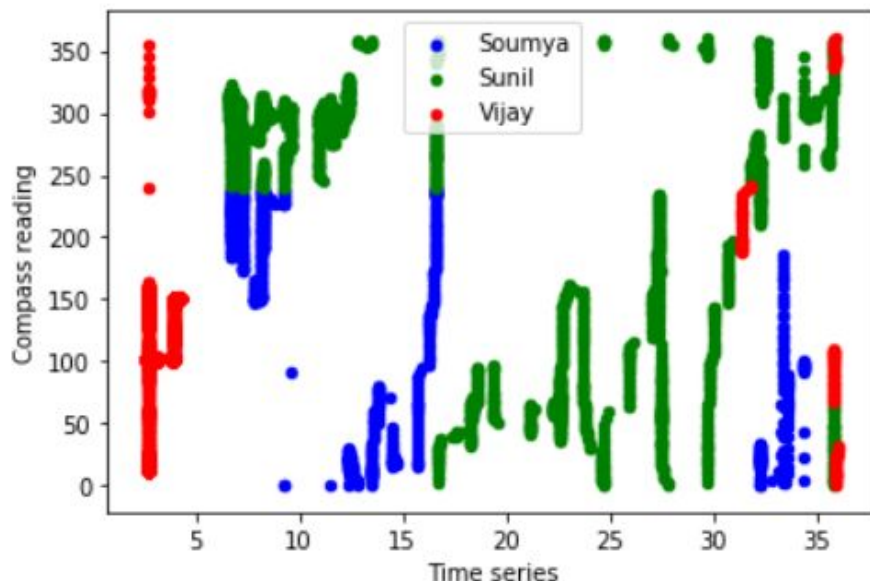


Interpretation

Shown above is the graph for battery level data collection. The graph displays the battery level for each member during the study. X-axis is the duration of time for the study and y-axis is the battery level.

- Looking at the graph we can conclude that Soumya's study started with battery levels being at about 90%, this level stayed consistent throughout her duration of study and increased slightly to about 94% by the end.
- Vijay's study started with battery levels being at about 30%, this level also stayed consistent throughout his duration and increased very slightly to maybe about 31% by the end.
- Sunil's study started off with battery levels around 22%. At the start there was a very small initial drop but over time the battery level increased to about 55% by the end.

Compass Heading (listening) Visualization



Interpretation

Shown above is the graph for compass heading level data collection. The graph displays the compass heading level for each member during the study. X-axis is the duration of time for the study and y-axis is the degree of compass heading level. Looking at the graph we can see that the lines are all over the place but if we look at each member at a time we can conclude the following things:

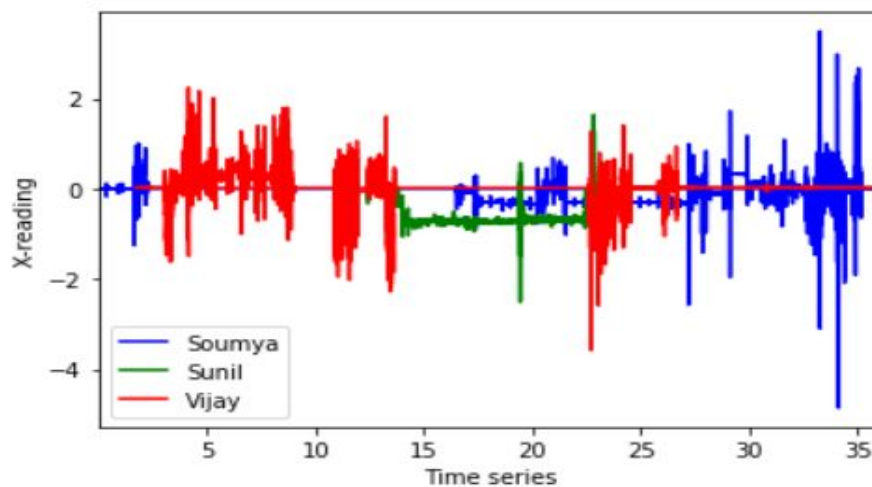
- Starting with Soumya, her compass started off with being in about 150 degrees and it fluctuated over time by going as low as 0 degrees but ended with about 100 degrees.

- Sunil's compass started off with being at about 225 degrees, increased a little and then jumped down to being pretty low around the 0 - 5 degree mark and then ended with a high degree around the 300 mark.
- Vijay's compass started with being about 150 degrees and had a drop to about 0 - 100 degrees. His compass ended with being about 200 degrees.

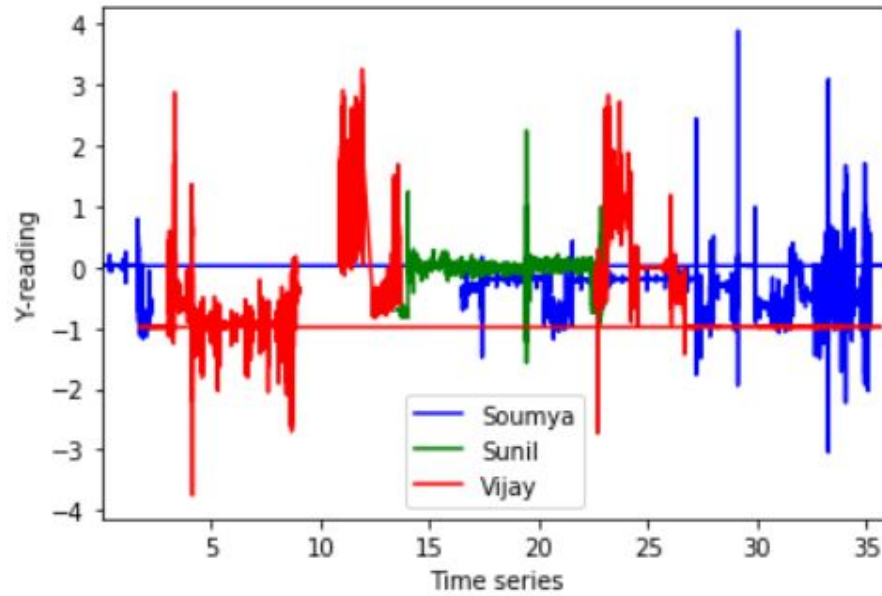
Acceleration (listening) Visualization

Interpretation

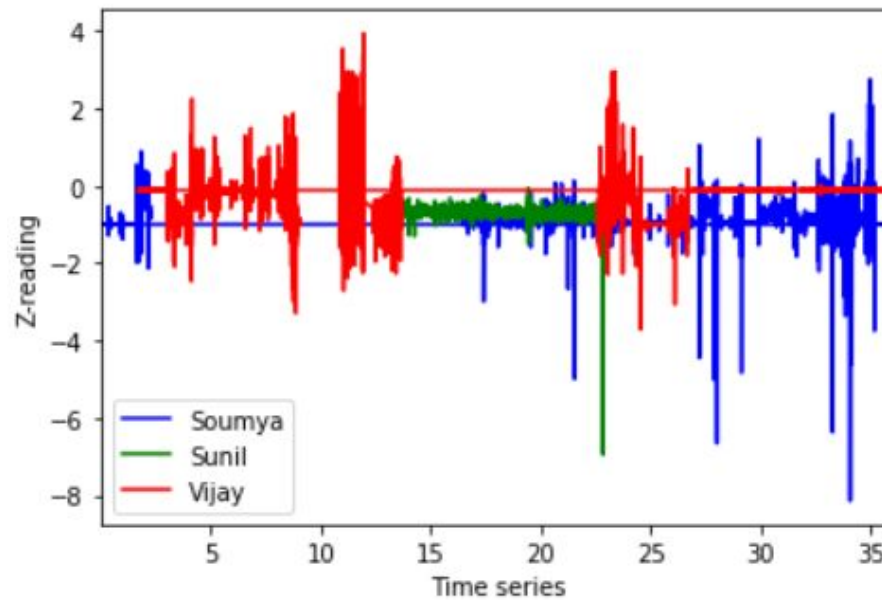
Shown below are graphs for the acceleration data collection. The plots display the acceleration level for each member during the study. The y-axis' change for each plot, according to which coordinates of the accelerometer we are looking at. X-axis is the duration of time for the study and y-axis is the coordinates of the accelerometer.



The graph above plots the data for the X-coordinates of accelerometer over a duration of time.



The graph above plots the data for the Y-coordinates of accelerometer over a duration of time.

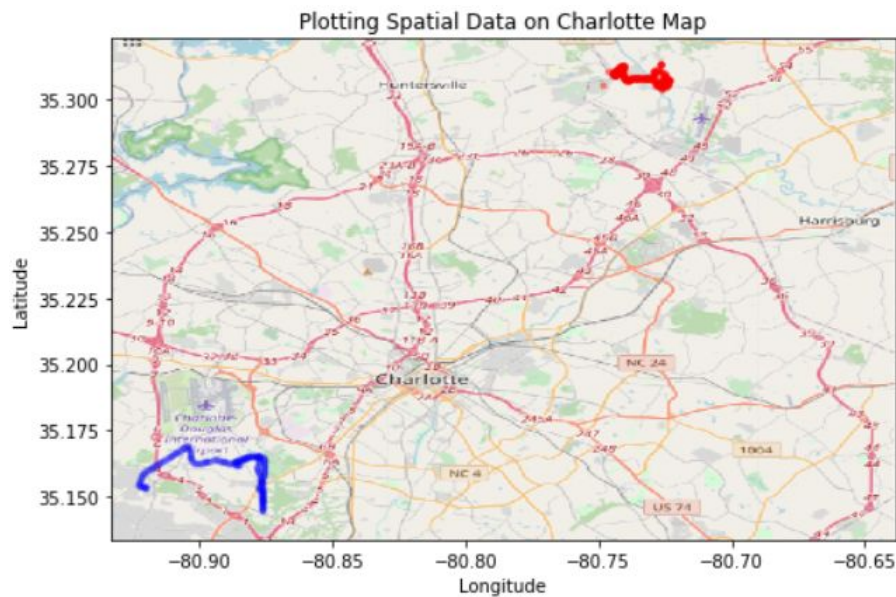


The graph above plots the data for the Z-coordinates of accelerometer over a duration of time.

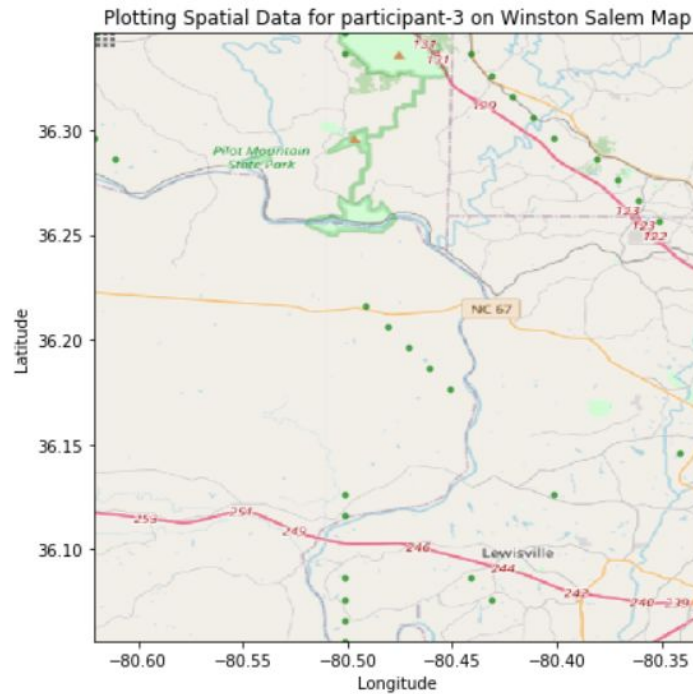
GPS Location (polling) Visualization

Interpretation

Shown below are graphs for the GPS location collection. The plots display the locations for each member during the study by. X-axis is the longitude and y-axis is the latitude. There are a total of four graphs below; the first graph displays the data of two members and the rest are graphs that represent each individual member's location data. Every point on the plots can be located using the longitude value as well as the latitude value.



Looking at the graph above you can see there are two different colors, each color representing a member GPS location during study. For this graph only two member's (Sunil and Vijay) data was plotted since they were both very close in location, the third member (Soumya) was not able to be shown clearly since she was in a further location compared to these too. You can see in this graph that though both Sunil and Vijay are in Charlotte they both traveled through different parts of the city.



Looking at the graph above you can see this member is located in Winston Salem. This member didn't record their data consistently and this is shown in the plot. By observing the green dots you can see that they don't follow in a straight line. But points of location was collected for different places visited by the member.

Bluetooth Visualization

Interpretation

After running sensuspy, there were two files that were created in the pickled folder: BluetoothDeviceProximityDatum and ProtocolDatum. Below we have shown the screenshot of different columns captured in BluetoothDeviceProximityDatum.

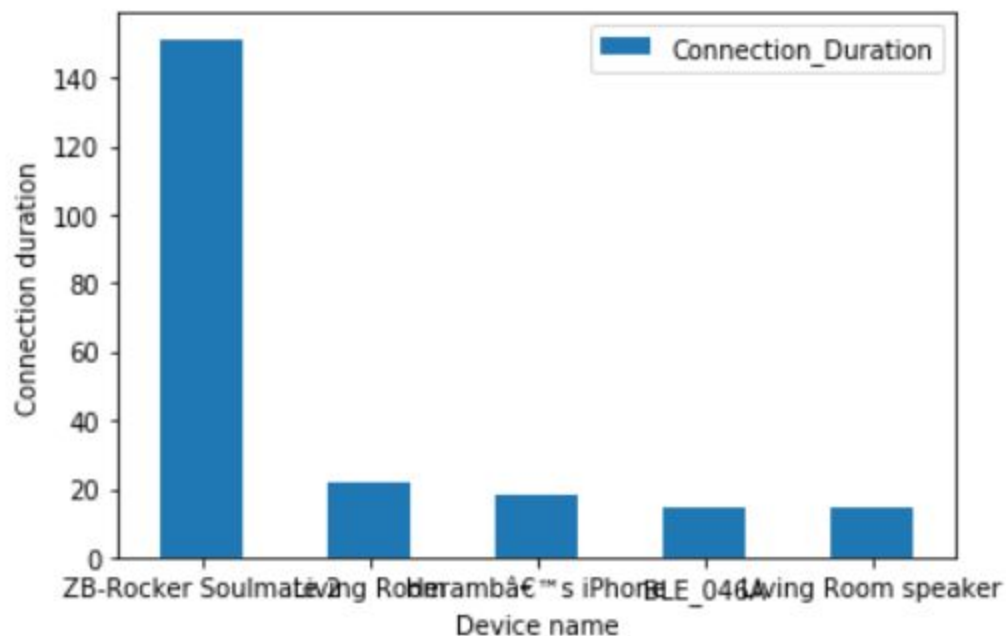
Address	Buildid	DeviceId	DeviceMa	DeviceMd	Encounte	Id	LocalOffs	Name	Operating	Paired	Participant	ProtocolC	Rssi	RunningSi	SensingA	TaggedEvi	TaggedEvi	Timestamp	Type	OS	Form
74:42:88:0B:15:18.0	fe2104e89c0	OnePlus	OnePlus5	7E1270F9-A2:44fcfd0-985	04:00:00	iPhone	Android P	TRUE				6d3fcb6c-071	-74	TRUE				2020-05-04 (BluetoothDe SensusAndra #####			
74:42:88:0B:15:18.0	fe2104e89c0	OnePlus	OnePlus5	7E1270F9-A2:901789b9-75	04:00:00	iPhone	Android P	TRUE				6d3fcb6c-071	-74	TRUE				2020-05-04 (BluetoothDe SensusAndra #####			
74:42:88:0B:15:18.0	fe2104e89c0	OnePlus	OnePlus5	7E1270F9-A2:267821de-41	04:00:00	iPhone	Android P	TRUE				6d3fcb6c-071	-74	TRUE				2020-05-04 (BluetoothDe SensusAndra #####			
74:42:88:0B:15:18.0	fe2104e89c0	OnePlus	OnePlus5	7E1270F9-A2:67612b02-45	04:00:00	iPhone	Android P	TRUE				6d3fcb6c-071	-74	TRUE				2020-05-04 (BluetoothDe SensusAndra #####			
74:42:88:0B:15:18.0	fe2104e89c0	OnePlus	OnePlus5	7E1270F9-A2:b11d7214-81	04:00:00	iPhone	Android P	TRUE				6d3fcb6c-071	-74	TRUE				2020-05-04 (BluetoothDe SensusAndra #####			
74:42:88:0B:15:18.0	fe2104e89c0	OnePlus	OnePlus5	7E1270F9-A2:284de8ac-c5	04:00:00	iPhone	Android P	TRUE				6d3fcb6c-071	-74	TRUE				2020-05-04 (BluetoothDe SensusAndra #####			
74:42:88:0B:15:18.0	fe2104e89c0	OnePlus	OnePlus5	7E1270F9-A2:a517b86b-54	04:00:00	iPhone	Android P	TRUE				6d3fcb6c-071	-74	TRUE				2020-05-04 (BluetoothDe SensusAndra #####			
74:42:88:0B:15:18.0	fe2104e89c0	OnePlus	OnePlus5	7E1270F9-A2:d42c6f85-c71	04:00:00	iPhone	Android P	TRUE				6d3fcb6c-071	-74	TRUE				2020-05-04 (BluetoothDe SensusAndra #####			

For capturing bluetooth data one member in the group set up a common protocol that was shared to all the other members in the group to download and use. Once we started the study we were

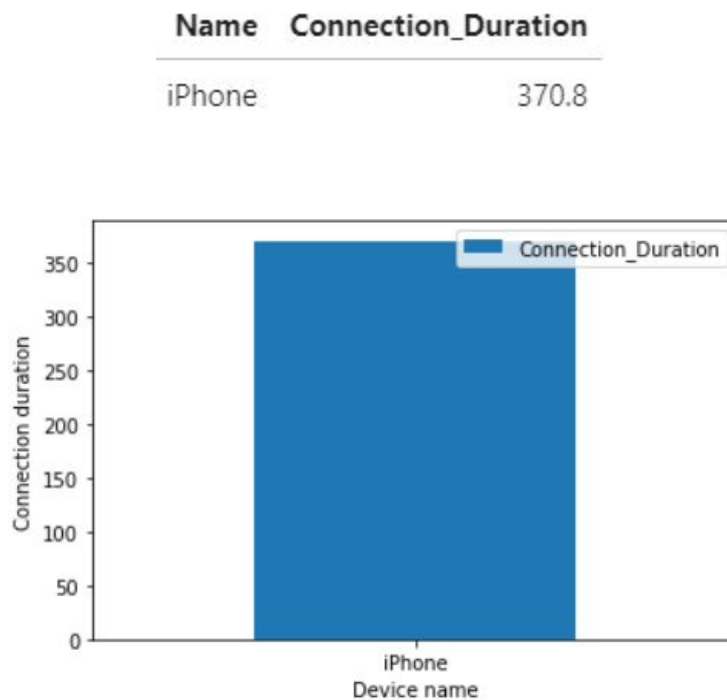
able to capture the device details of all the other devices that were connected to our own personal devices.

The following screenshot shows the names of various devices that were connected to one of the member's devices using bluetooth. This member was using an Android mobile. You can see that this mobile was able to connect to five different devices; ZB-Rocker Soulmate 2, Living Room, Harambae iPhone, BLE_046A and Living Room Speaker. Aggregation was performed on each device that was connected to calculate how long each device was connected for. A plot was also created (which is shown below) to visualize how long each of these five devices was connected to the Android mobile.

Name	Connection_Duration
ZB-Rocker Soulmate 2	151.2
Living Room	21.6
Harambae™'s iPhone	18.0
BLE_046A	14.4
Living Room speaker	14.4



The following screenshot shows the name of the device that was connected by another member's personal device in the group. This member was also using an Android. You can see below that this Android mobile was able to connect to an iPhone. Aggregation was also performed here to calculate how long the iPhone device was connected to the Android mobile. The graph has a bar plotted which represents the duration of how long the Android mobile was connected to the iPhone.



Challenges

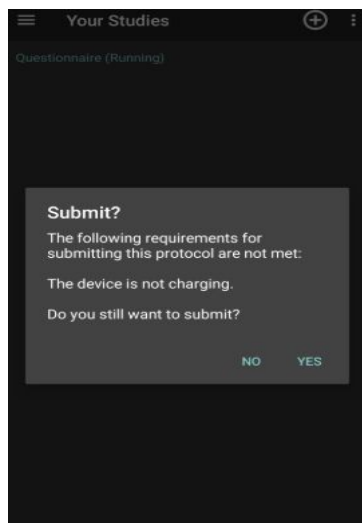
Throughout this project we faced a lot of issues while working with many of these different tools. These issues ranged from not being able to collect data correctly to not being able to set up the SensusPy virtual environment. All of our issues we face were with the Sensus app, AWS and SensusPy. Though we faced these challenges we worked as a team and were able to resolve the challenges. Not only did we run into challenges that we had to overcome we were also able to come across some important findings that helped us greatly with the project.

Sensus App Challenges

- The first challenge we ran into with the app was that we were not able to upload the data to our S3 bucket collected from the app even though all our information was correct in the remote data store section. To fix this issue we had to look through our logs on the

app. Looking at the logs we found out that there was a log that had an AWS uploading exception, with that log there was a message that said that there was some issue with the network. With this we figured out that the wifi was affecting us by not being able to upload the data.

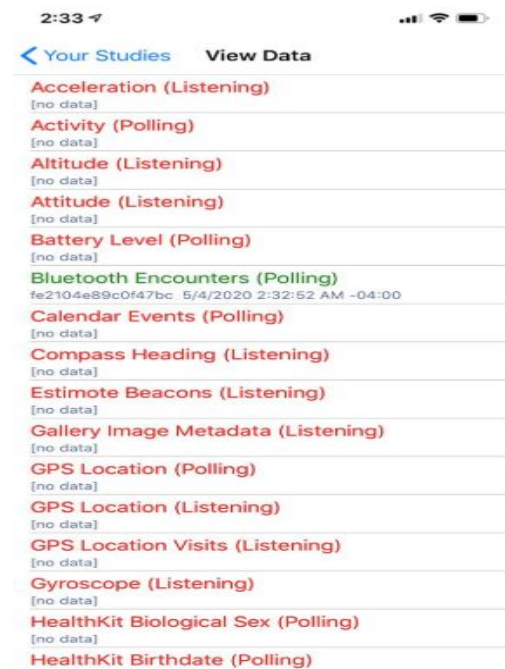
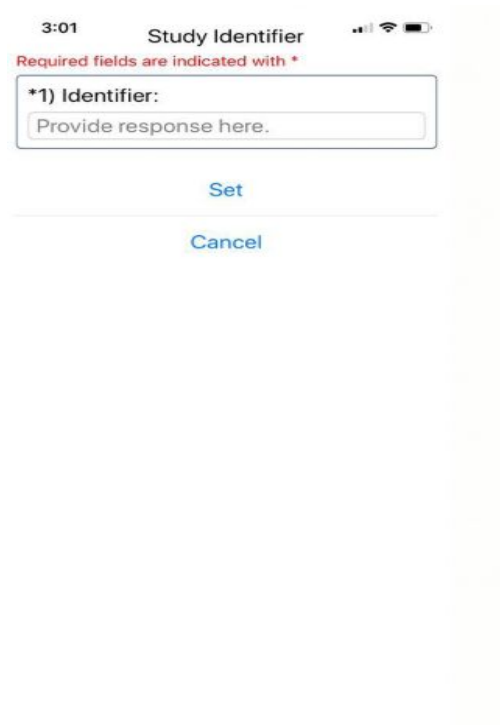
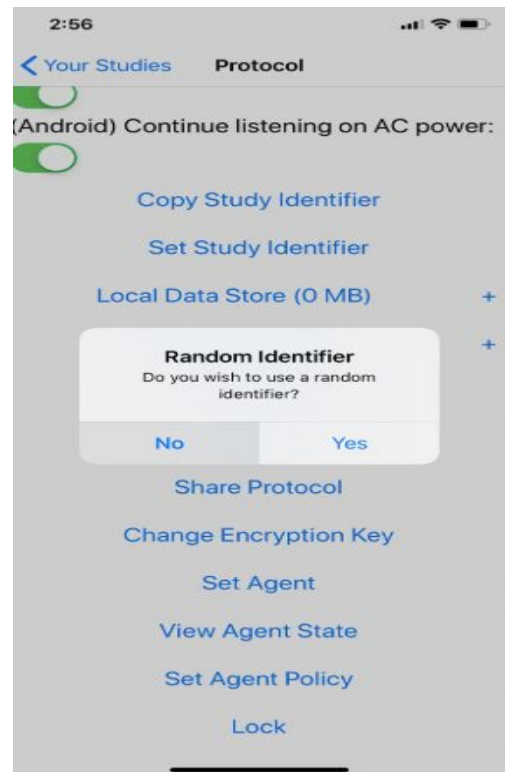
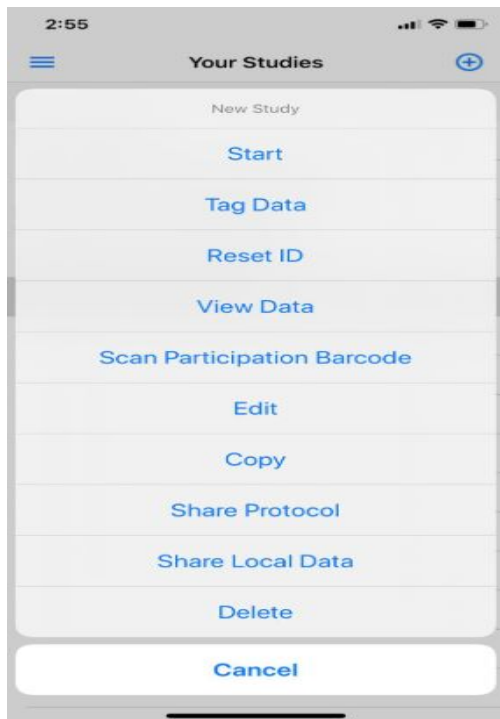
- The second challenge we had with the app also had to do with the submission of our data to the S3 bucket. When we tried to submit the data we received a popup message that said the device (the mobile) was not charging and if we wanted to still submit that data, even if we clicked ‘yes’ this message would show up every time we try to submit new data. The following picture shows the popup that was received to our mobile. We resolved this issue by disabling a feature on the app. We had to disable the “require charging” option, which was located in the Remote Data Store section.



- The first finding that we discovered with this app has to do with the collection of the bluetooth data. The collection of the bluetooth encounters (polling) probe is not exactly what we thought. At first we thought the way to collect the data was by starting the studies and connecting our mobiles to bluetooth devices but we were wrong. To collect data with this probe it actually scans the presence of other devices that are close by that are also running the current protocol. It will detect only if you are in the presence of other devices that have the Sensus app running in the background/foreground, this will read the ID of the other devices.

There was only one named ProtocolDatum getting created in the pickled folder. We were not able to file BluetoothDeviceProximityDatum which contains details regarding the encountered device and the duration we were connected to. We resolved this issue by sharing the protocol with other team members and then using a set study identifier to set

the identifier of the person who has shared the protocol and then it started collecting the data of encountered devices. Attached the screenshot for your reference.



AWS & S3 Bucket Challenges

- The first challenge we came across was when we had to create a S3 bucket on a Windows laptop, this was quite difficult. Since Windows didn't have an UUID generator by default we had to set it up. To do this we had to download and use the cygwin app and execute the `'/configure-s3.sh sensus3 us-east1-'` command script. By doing this we were able to create the S3 bucket and configure the protocol in the Sensus app and get started with collecting our data.
- The second challenge that we encountered with AWS being able to share the bucket that was created to other members in the team. Though there were instructions on how to create a bucket it wasn't very clear how to be able to share this bucket with others. To resolve this issue we used the following link: <https://docs.aws.amazon.com/AmazonS3/latest/dev/BucketAccess.html> and worked our way around to be able to make a shareable bucket.
- The last challenge we ran into was with installing the AWS client. The problem occurred on Mac laptops. Even though we followed the steps that were given to us to follow we still were having trouble installing the AWS client. We realized that not only do we have to follow the steps but we also had to download the MacOS PKG installer. By following the instructions we were given and downloading the MacOS PKG we were able to successfully install the AWS client.

SensusPy Challenges

- The challenge we had with sensuspy was when we were installing the package. While setting up the virtual environment we had an issue with the last step when we had to import sensuspy as sp so we would be able to successfully have suspy available as a package. We realized that we encountered this challenge because of the pip version we had installed. We originally had pip version 10.0 this version can't uninstall 'certifi,' which is a distutils installed project. Since we couldn't uninstall certifi' it couldn't accurately determine which files belong to it. To solve this we changed our pip version to version 9.0, after changing our version we were successfully able to set up our package.
- There were quite a few findings that we came across with sensuspy. All of these findings had to do with the github py files that were given to us. There were errors in the code for the following files; data_retrieval.py, data_operations.py, and plot.py.

Data_retrieval.py

For the data_retrieval.py file there is a variable called local_path, this variable was wrongly used instead with another variable called data_path, meaning at one point in the code the variable local_path should have been used but data_path was used instead. Below is the original code and our updated code.

Original code:

```
paths = glob.glob(data_path + '*/**/*.gz', recursive=True)
```

Updated code:

```
paths = glob.glob(local_path + '*/**/*.gz', recursive=True)
```

Data_operations.py

For the data_operations.py file the issue was the same as data_retrieval.py but instead of a variable that was wrongly used this time it was a function named drop_duplicated_from_datum. Function drop_duplicated_from_data function is calling drop_datum_duplicated instead of drop_duplicates_from_datum in the code. Below is the original code and our updated code.

Original code:

```
def drop_duplicates_from_data(data):  
    """Drops duplicate rows from each pandas dataframe in the data dictionary."""  
  
    deduplicated_data = {}  
    for datum in data:  
        deduplicated_data[datum] = drop_datum_duplicated(data[datum])  
    return deduplicated_data
```

Updated code:

```
def drop_duplicates_from_data(data):  
    """Drops duplicate rows from each pandas dataframe in the data dictionary."""  
  
    deduplicated_data = {}  
    for datum in data:  
        deduplicated_data[datum] = drop_duplicates_from_datum(data[datum])  
    return deduplicated_data
```

Plot.py

Lastly for the plot.py file the issue was with the imports at the beginning of the code. The issue was with one specific import, the code provided us with `import matplotlib.pyplot as pl`, which didn't work. Below is the original code and our updated code.

Original code:

```
import math
import gmpplot
import math
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.dates as mdates
from matplotlib.backends.backend_pdf import PdfPages
```

Updated code:

```
import math
import gmpplot
import math
import numpy as np

import matplotlib
matplotlib.use('TkAgg')
import matplotlib.pyplot as plt

import matplotlib.dates as mdates
from matplotlib.backends.backend_pdf import PdfPages
```

Extra Credit

Applying Differential privacy to our Data

Following value shows the difference between actual value and updated value by introducing an error of $\epsilon=1$.

This challenge requires us to capture statistics for the number of times each member's phone pointed between north and east. Since 0° on the compass represents north and 90° represents east, hence we will calculate number of times compass pointed between 0° and 90° .

```

Number of times Vijay phone points between North and East :: Heading    431
dtype: int64
Number of times Sunil phone points between North and East :: Heading    599
dtype: int64
Number of times Soumya phone points between North and East :: Heading    339
dtype: int64

```

Following value shows the difference between actual value and updated value by introducing an error of epsilon=1

Heading	Upd_Heading
82.4567	83
83.4567	83
84.4567	86
85.4567	87
86.4567	90

Following are the 3 CSV files with actual and updated values are generated. We have used 3 python scripts to develop the entire solution :-

VJUpd_CompassReading.csv

SNUpd_CompassReading.csv

SMUpd_CompassReading.csv

sensusAnalyze.py - This script is used to download zipped json files from AWS S3 bucket and process the json files to create pickled files for further analysis.

SensusAnalysis.ipynb - This script is used to read pickled files created by the sensusAnalyze.py and convert files into csv readable format.

ProbeAnalysis.ipynb - This script is used to perform analysis for all different types of probe and generate graphical statistics. It further implements the additional challenge in which only compass entries for values between north and east were supposed to be considered and further noise with epsilon value=1 was induced.

Conclusion

In conclusion, working with the Challenge 2 project gave us a chance to work in areas we haven't before. We were able to gain knowledge overall on how to collect data using the Sensus app, how to combine and pre-process files using Sensuspy and how to make graphs using these files and python. There were many obstacles that we came across while working with this assignment but it was very helpful since the members of the team had a lot of different skill sets, for example, one of the team member (Vijay) has very strong skill when it comes to working with Python, because of him we were able to make all the visualizations possible for this project! At the end of the day teamwork was very handy. By working together and being able to resolve the challenges and find further findings we completed the assignment and were able to reach our goal!