A real-world case study: smartphone-based indoor localization

Chang Fuh Yong
School of Computer Science and Engineering
Nanyang Technological University Singapore
ch0113ng@e.ntu.edu.sg

Brandon Chua Shaojie
School of Computer Science and Engineering
Nanyang Technological University Singapore
bran0026@e.ntu.edu.sg

Cheong Kok Yuet
School of Computer Science and Engineering
Nanyang Technological University Singapore
kcheong012@e.ntu.edu.sg

***Abstract*—Outdoor localization is performed with the help of GPS. As GPS cannot penetrate buildings, localization within buildings has to be achieved by other means. The most widely used device in the world is the smartphone. It has built in sensors that, with user's permission, can be used to record data that provides location information when the user is indoors. For this project, readings collected by the phone's sensors were used to visualize waypoints, geomagnetic heatmaps and RSS heatmaps of Wi-Fi access points. Dataset is taken from the Microsoft Indoor Location Competition 2.0. A deep learning-based location fingerprint model was also developed.**

Keywords—smartphone, indoor, localization

I. Introduction

Global Positioning System (GPS) is a satellite based system that provides the location of an object located outdoors. With the information provided by GPS, the user is able to navigate almost everywhere in the world. However, as the GPS signal is unable to penetrate buildings effectively, navigating indoors poses a problem for the user when he is not familiar with the location. For cases like this, indoor localization can be carried out by either device-free or device-based methods. For this project, we will be focusing on the device-based method. In order to learn one's location, the user has to rely on a device that he/ she carries all the time everywhere. A smartphone is the most popular device that a user carries. It has built-in sensors such as wifi, bluetooth, compass, gyroscope, accelerometer, GPS, light sensor and barometer. With the user's permission and the help of the sensors, information can be collected and analyzed. Prediction can be performed after the analysis to provide localization and contextual information based on the spatio-temporal data collected by the smartphones.

In this project, our group is given a dataset taken from the dataset used in the Microsoft Indoor Location Competition 2.0. The dataset contains readings collected by the smartphone sensors. Our group is tasked with using the readings to visualize way points (ground truth locations), geomagnetic heat maps and the RSS heat maps of Wi-Fi access points. To carry out the task of visualization, we used Matplotlib for Python. A deep learning-based location fingerprint model was also developed.

II. Dataset

The dataset from Microsoft Indoor Location Competition 2.0 contains dense indoor signatures of ground truth, geomagnetic field, Wi-Fi etc. About 30,000 traces from over 200 buildings are inside the dataset. The dataset is collected by holding an Android smartphone in front of the surveyor's body during a walk. A sensor data recording app is run during the walk to collect IMU, geomagnetic field, Wi-Fi and Bluetooth iBeacon readings.

For this project, a smaller dataset containing data from two multi-story commercial buildings, labelled as site 1 and site 2 is used. Site 1 contains five floors while site 2 contains nine floors. Each floor contains floor plan metadata, which consists of raster image, size and GeoJSON files, and trace files (*.txt) that correspond to the paths walked by a surveyor between two positions. The format of the trace file can be broken down into 3 categories. The first is Unix Time in milliseconds. The second is the data type, there are ten in total. The third is the value of the data collected. Table 1 shows an example of the trace file format while Table 2 shows the types of data type and its values.

Time	**Data Type**	**Values**
1574659531598	TYPE_WAYPOINT	196.41757 117.84907

TABLE 1: Trace file format example

Data Type	**Values**
TYPE_WAYPOINT	Coordinate x, Coordinate y
TYPE_ACCELEROMETER	x, y ,z, accuracy
TYPE_GYROSCOPE	x, y, z, accuracy
TYPE_MAGNETIC_FIELD	x, y, z, accuracy
TYPE_ROTATION_VECTOR	x, y, z, accuracy
TYPE_ACCELEROMETER_UNCALIBRATED	x_a, y_a, z_a, x_b, y_b, z_b accuracy
TYPE_GYROSCOPE_UNCALIBRATED	x_a, y_a, z_a, x_b, y_b, z_b accuracy
TYPE_MAGNETIC_FIELD_UNCALIBRATED	x_a, y_a, z_a, x_b, y_b, z_b accuracy
TYPE_WIFI	ssid, RSSI, frequency, last seen timestamp
TYPE_BEACON	UUID, MajorID, MinorID, Tx Power, RSSI, Distance, MAC Address, padding data

TABLE 2: Data Type and its Values

Data Type	Number of readings
TYPE_WAYPOINT	362
TYPE_ACCELEROMETER	85189
TYPE_MAGNETIC_FIELD	85189
TYPE_ROTATION_VECTOR	85189
TYPE_WIFI	228229

TABLE 3: Number of readings for different data types for Site 2 Floor 2

Table 3 shows the difference in number of readings between way points, accelerometer, magnetic fields, rotation vectors and Wi-Fi. The number of recorded way points is significantly lesser compared to the other data types. This suggests that magnetic field, accelerometer and rotation vector readings are recorded more frequently. The number of Wi-Fi readings is the highest. This suggests that Wi-Fi readings are recorded more frequently than way points. Another factor is also due to the surveyor being in proximity of multiple Wi-Fi access points, resulting in more Wi-Fi readings.

III. ESSENTIAL TASKS

For this project, there are three essential tasks to be done. The tasks are to visualize way points (ground truth locations), geomagnetic heat maps and the RSS heat maps of Wi-Fi access points.

A. *Task 1:Visualizing Way Points*

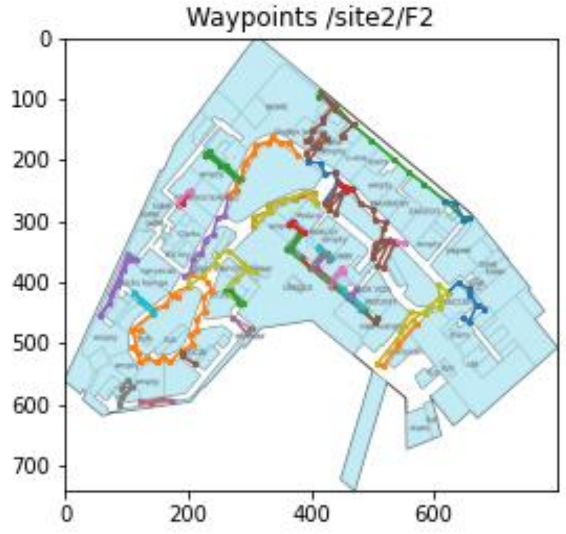


Figure 1: Way points of all paths on Site 2 Floor 2

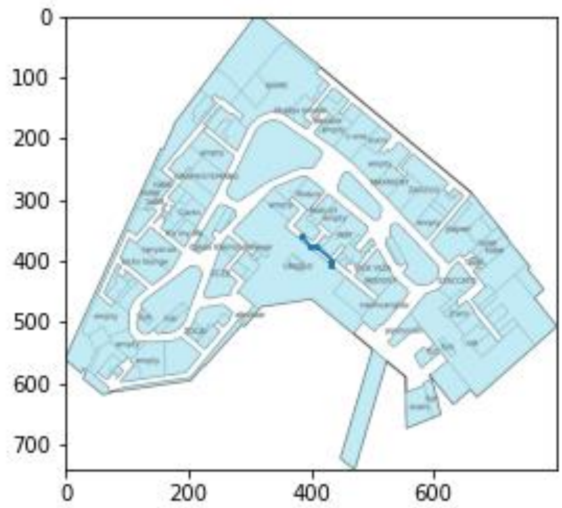


Figure 2: Way points of a path

For each floor of the sites, there are multiple trace files. Each trace file corresponds to a path taken by a surveyor. While the surveyor is walking from position A to position B, way points are recorded along the path taken. Inside the file, the text line containing data type TYPE_WAYPOINT has values that show the X and Y coordinates of the way points and the time at which it is recorded. For each floor, all TYPE_WAYPOINT instances are plotted using Matplotlib onto an image of the respective floors.

Figure 1 shows the plotting of the waypoints for the 2nd floor of site 2. Waypoints belonging to the same path have the same colour. Waypoints are represented by dots, while the lines connecting the dots are a rough estimate of the path taken from one way point to another way point. There are empty areas where no way points are plotted which indicates that the surveyor did not walk through that part of the floor. Some of the lines intersect, which suggests that some of the paths taken by the surveyor overlaps to some degree. Figure 2 shows the way points of one path. The plots of the way points for the rest of the floors can be found in the appendix.

B. *Step Positions*

From Figure 1 and Figure 2, it can be seen that the line plotted between waypoints is very abrupt and forms a zigzag pattern. This inaccurate estimate of the pathing that the surveyor walked is caused by having insufficient number of way points. This poses a problem for carrying out tasks 2 and 3, which is to visualise the geomagnetic and RSS heat maps. The geomagnetic and RSSI readings need to first be aligned with the way points before the heat maps can be generated. With less way points to align geomagnetic and RSS readings to, the heat maps generated will be less precise.Since the accuracy of the heat maps are dependent on the ground truths, using ground truth data that gives a more accurate representation of the path that the surveyor walked will therefore give a more accurate heatmap.

In the trace files, there are TYPE_ACCELEROMETER and TYPE_ROTATION_VECTOR readings. TYPE_ACCELEROMETER readings give information on the acceleration of the phone, while TYPE_ROTATION_VECTOR readings give information on the axis and angle the phone is facing in a three dimensional space. Using these two types of data, we can compute the positions of the steps that the surveyor took to walk from one way point to the other. Figure 3 shows the plot of step positions for the 2nd floor of site 2. All step positions are plotted as green dots. Comparing Figure 3 and Figure 1, it can be seen that there are significantly more step positions than way points. By using step positions as ground truth data, we now have a more precise representation of the path that the surveyor took. The step positions plot for the rest of the floors can be found in the appendix.

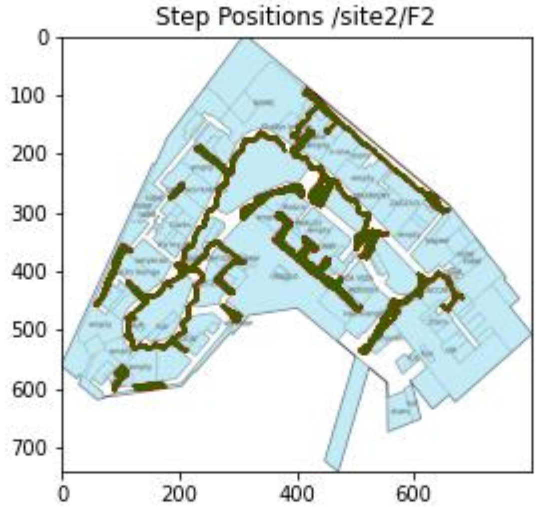


Figure 3: Step positions on Site 2 Floor 2

C. Task 2: Geomagnetic Heat Map

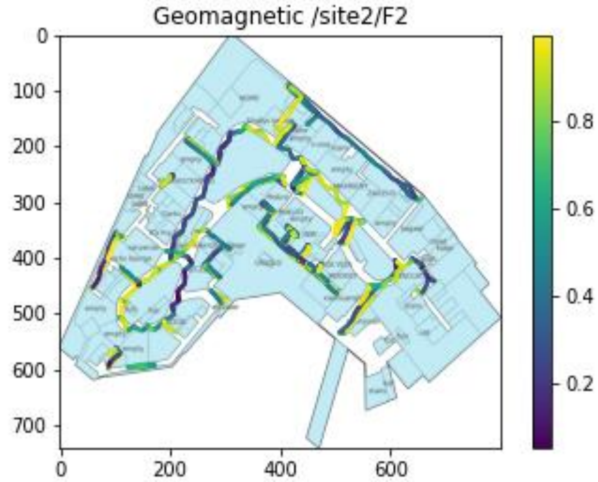


Figure 4: Geomagnetic heat map of Site 2 Floor 2

Using the step positions as ground truth, we align the TYPE_MAGNETIC_FIELD data to it to produce the geomagnetic heat map. This is carried out by comparing the timestamp of the TYPE_MAGNETIC_FIELD data to the timestamp of every step position. Having a smaller time difference indicates a smaller difference in distance between the location of the step position and the actual location at which the TYPE_MAGNETIC_FIELD is recorded. Therefore, having a zero time difference indicates that the TYPE_MAGNETIC_FIELD reading is recorded at that exact location of the step position. We then pair the TYPE_MAGNETIC_FIELD data to the step position that gives the smallest time difference.

Since there might be multiple TYPE_MAGNETIC_FIELD data aligned to the same step position, we will use Equation 1 to get a mean value for each step position.

$$Mean\ Magnetic\ Strength = \frac{1}{N}\sum_n \sqrt{x_n^2 + y_n^2 + z_n^2} \qquad (1)$$

$$where\ N = total\ number\ of\ TYPE\_MAGNETIC\_FIELD\ data$$

$$n = TYPE\_MAGNETIC\_FIELD\ datum$$

The mean magnetic strength is calculated by finding the mean of the sum of the square root of the sum of the x, y, and z coordinates squared. After calculating the mean magnetic strength for each step position, the geomagnetic heat map is plotted out. Figure 4 shows the geomagnetic heat map for the 2nd floor of site 2. As shown in the figure, areas with stronger magnetic strength are lighter in colour while areas with weaker magnetic strength are darker in colour. The heat maps for the other floors can be found in the appendix.

D. Task 3: RSS Heat Map

A WiFi heatmap is a visual representation of wireless signal distribution, typically showing signal strength using color-coded approaches, with lighter colours representing area with strong signal and darker colours representing area with a weak signal.

a. Localization of Instance

A WiFi heatmap is a visual representation of wireless signal distribution, typically showing signal strength using color-coded approached, with the color red representing area with strong signal and color green representing area with a weak signal.

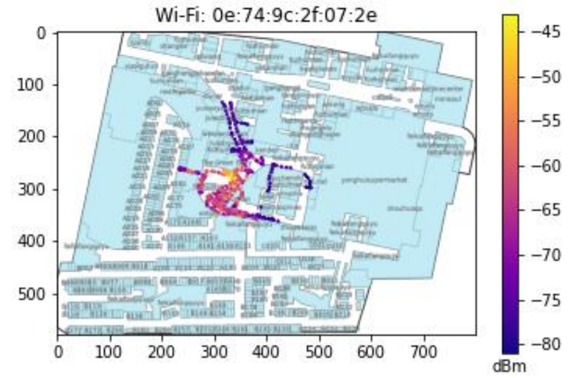
Localization of Instance WiFi

The instance $WIFI_i$ =($bssid_i$, t_i,r_i) where r_i is the WiFi signal strength of this instance and $bssid_i$ is the BSSID of the AP of this instance. Next, we compare the time t_i this instance is generated with all timestamps t_i^{WP} for the ground true way points. The way point position [Px, Py) whose timestamp is closest to the time the WiFi instance is obtained.

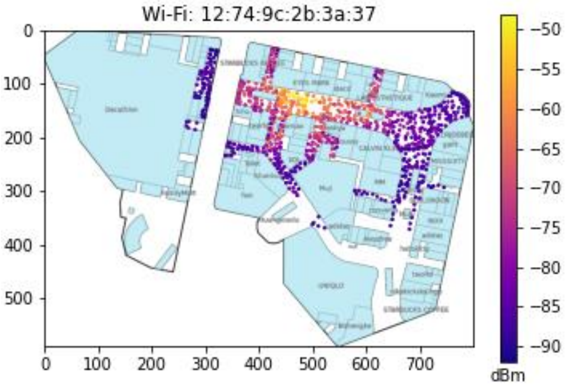
Generate the RSS Heat map

Firstly we construct a dictionary object wifi_rssi which will have the bssid as key and the value are way point location information (x,y) and the rssi signal strength (dbm)

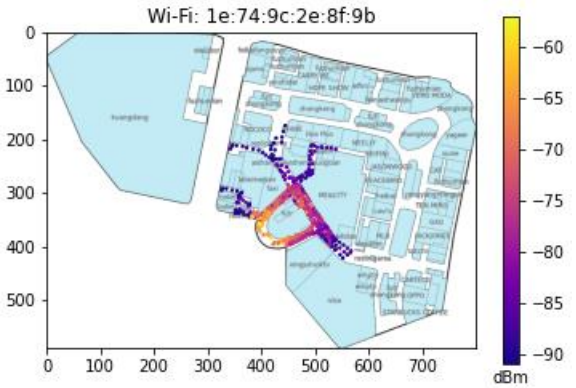
With the known bssid and rssi value, we generated the heat map. We randomly generate total of 70 heat maps with 3 maps for each floor. Refer to figure 5 for the heat maps as examples.



AP: 0e-74-9c-2f-07-2e at Site 1 B1

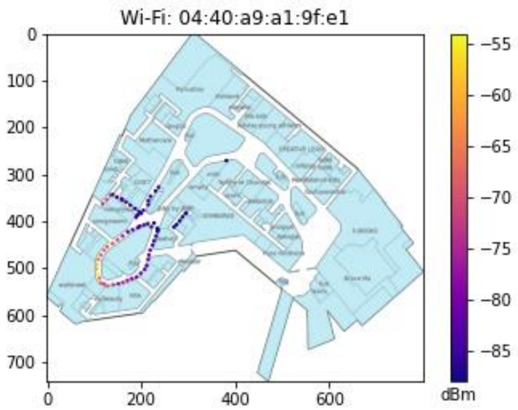


AP: 12-74-9c-2b-3a-37 at Site 1 F1

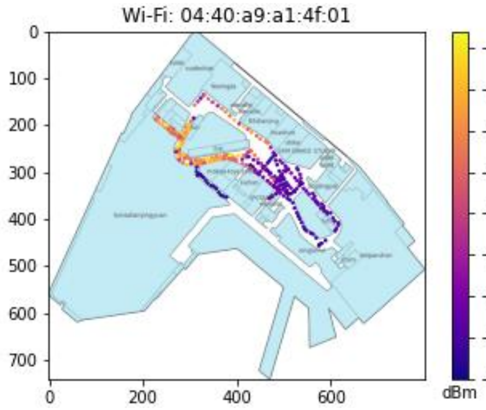


AP:

AP: 1e-74-9c-2e-8f-9b at Site 1 F2



AP: 04-40-a9-a1-9f-e1at Site 2 F3



AP: 04-40-a9-a1-4f-01at Site 2 F8

Figure 5: RSS heat maps of 5 different Aps from different floors.

IV. Bonus

A. *Deep Learning Location Fingerprint Model*

The objective of this model is to use deep learning to provide an indoor localization using the collected information. Indoor localization also known as indoor positioning depends on various types of indoor sensor technology such as infrared, wifi and proximity based system. In this use case, sensor data from smartphone is collected, processed and used to train the model for predicting the user's location. This section is divided into data preprocessing in preparation for training and testing of the model. Architecture of the model is then described to provide a general understanding of the design. Results are then provided for the various experiments conducted on the trained model.

1) Preprocessing: For the experiment, infromation is extracted from the **.txt* files. The data contains information collected with user's consent from smartphone's geomagnetic, wifi and ibeacon sensors. The ground truth location (x, y) recorded by the volunteers is the expected result used to verify the trained model. Information is analyzed and divided into four different groups for training, namely wifi input, magnetic input, beacon input and wifi+magnetic input:

$$Input_{wifi} = [ssid\ rssi\ x\ y]$$

$$Input_{magnetic} = [m_x\ m_y\ m_z\ x\ y]$$

$$Input_{ibeacon} = [uuid\ rssi\ x\ y]$$

$$Input_{wifi\_magnetic} = [ssid\ rssi\ m_x\ m_y\ m_z\ x\ y]$$

where x, y are waypoint's x, y coordinates. The coordinates are the label used in verifying the model. m_x, m_y m_z are geomagnetic data. *SSID* are wifi Access Point's (AP) ID and *UUID* are ibeacon's ID. *SSID* and *UUID* have to be encoded as the model cannot process alphabets but only numbers. Standardization is performed on the data such that each type of data or also known as feature behaves like a normal distribution. The whole dataset is split into 80% for training the model and 20% for testing.

2) Architecture: Figure 6 illustrates the architecture of the model. In this use case, the following information is used to design the deep learning model:

- Input: Sensors' data $Input_{wifi}$, $Input_{magnetic}$, $Input_{ibeacon}$ and $Input_{wifi\_magnetic}$ are fed into the model.
- Output: Predicted *x, y* waypoint coordinates is the expected outcome.
- Type: This use case is a supervised regression problem where the outcome in the format of real numbers representing the coordinates is provided for training.

Information is input into two groups of hidden layer, namely Encoder and Decoder.

- Encoder: It consists of two fully connected network to evaluate the input *features* and transforms the data into model's parameters represented by a function *E*.

$$outcome_E = E(features)$$

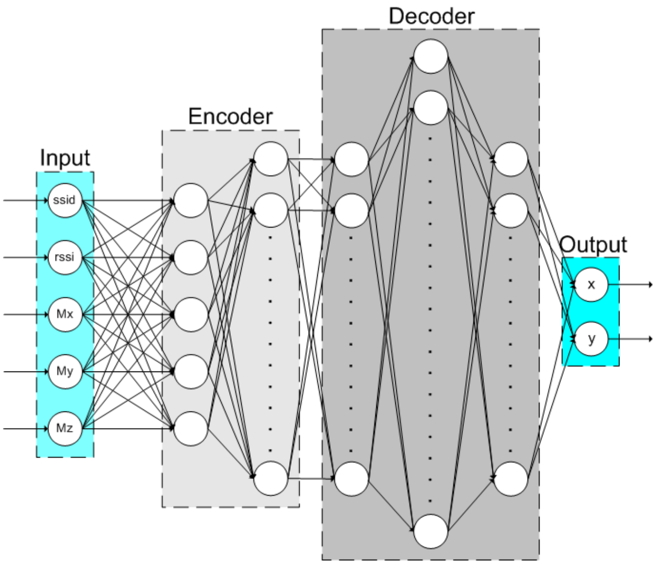


Fig. 6. Model architecture

- Decoder: It consists of three fully connected network to translate the Encoder's output $outcome_E$ using function *D* and presents the result as the expected output *label*.

$$label = D(outcome_E)$$

The model accepts the input *features* and predicts the *label*.

$$label = D(E(features))$$

a) Loss Function: This function measures the margin of error between the predicted *label* and the ground truth waypoint. This function is used to calculate the adjustments required on the model's parameters. Following are the annotations used:

$$MSE = \frac{1}{M}\sum_{i=1}^{M}(x_i - \hat{x})^2 + (y_i - \hat{y})^2 \qquad (2)$$

where *M* represents the total number of samples in the training set. x_i represents the predicted x coordinate for each iteration *i*, $\hat{x}$ represents the x coordinate of ground truth waypoint. y_i represents the predicted y coordinate for each iteration *i*, $\hat{y}$ represents the y coordinate of ground truth waypoint.

b) Error: In each training iteration, this metric measures the margin of error between each predicted waypoint *i* and the ground truth waypoint. The error is based on Euclidean distance of two points and a small error implies the model is performing well.

$$Error = \sqrt{(x_i - \hat{x})^2 + (y_i - \hat{y})^2} \qquad (3)$$

c) Optimizer: This function adjusts the model's parameters iteratively in order to minimize the loss calculated in *Loss Function*. The gradient of this function is the partial derivative of the loss function with respect to each model's parameter.

3) Training: Different number of epochs are used to train the model depending on the type of sensor data used.

a) Wifi: 350 epochs are used for training with wifi data,. However from Figure 7, it is observed that the loss and error stabalizes around 200 epochs. This suggests that 200 epochs are sufficient to train the model using wifi data.

b) Geomagnetic or Beacon: 150 epochs are used for training with geomagnetic or beacon data. From Figure 8 and Figure 9, it is observed that the loss and error stabalizes around 100. It suggests that 100 epochs are sufficient to train the model using geomagnetic or beacon data.

c) Wifi and Geomagnetic: 100 epochs are used for training with combined wifi and geomagnetic data. From Figure 10, it is observed that the loss and error stabalizes around 100. It suggests that 100 epochs are sufficient to train the model using the combined data.

4) Results: Site two's floor F8 is used to evaluate the model.

a) Wifi: Figure 7 illustrates the training loss of 0.925, training error of 1.19 and testing error of 1.2. The training loss, training error and testing error shows that the model can be trained using the huge quantity of wifi only data .

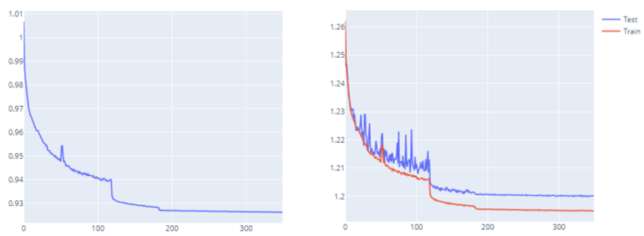


Fig. 7. Wifi training loss (left) and training + testing error (right)

b) Geomagnetic: Figure 8 illustrates the training loss of 0.35, training error of 0.65 and testing error of 0.65. The training loss, training error and testing error shows that the model can be trained using the sufficient quantity of geomagnetic only data.

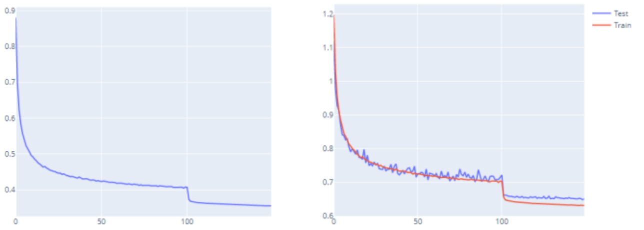


Fig. 8. Geomagnetic training loss (left) and training + testing error (right)

c) Beacon: Figure 9 illustrates the training loss of 0.89, training error of 1.18 and an unacceptable testing error. The training error and testing error shows that the model cannot be trained using the beacon only data as there is insufficient beacon data as shown in Table 1. The testing error plot fluctuates widely reflecting the model is not robust and more data is required to train the model using only beacon data.

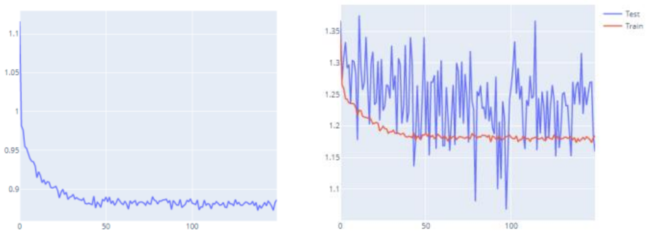


Fig. 9. Beacon training loss (left) and training + testing error (right)

d) Wift and Geomagnetic: Figure 10 illustrates the training loss of almost 0, training error of almost 0 and testing error of 0.75. The training loss, training error and testing error shows that the model can be trained using the data containing both wifi and geomagnetic information.

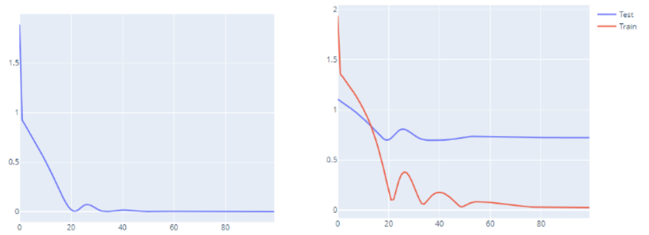


Fig. 10. Wifi + Geomagnetic training loss (left) and training + testing error (right)

Table 1 summarises the data points for each sensor type. The number of beacon data points is too sparse to be used for training resulting in the plot of Figure 9. A round of test with combined wifi and geomagnetic data which have the same waypoint coordinates was conducted to verify the idea that training with more data from other sensor types will result in a more accurate model. Figure 10 shows that the testing error of combined wifi and geomagnetic data was reduced from wifi's 1.2 to combined data's 0.75. This is an 37.5 % improvement using combined sensor data. The idea of combining all sensor data namly, wifi, geomagnetic and beacon data for localization was suggested since combining wifi and geomagnetic data provides significant performance improvement. However due to insignificant amount of beacon data available, it is deduced that the beacon data will only create flutation in the plots resulting in poor modeling.

TABLE I. SENSOR DATA STATISTICS

Sensor Type	**Data Points**	**Percentage**
Wifi	147,718	68.70
Geomagnetic	66,601	30.97
Beacon	697	0.32
Total	215,016	100.00

TABLE II. TRAINING TESTING ERROR LOSS

Sensor Type	**Training Loss**	**Training Error**	**Testing Error**
Wifi	0.925	1.19	1.2
Geomagnetic	0.35	0.65	0.65
Beacon	0.89	1.18	NA
Wifi + Geomagnetic	0	0	0.75

With the new model, prediction of waypoint is conducted using raw sensor data. A sensor item is randomly selected from a whole floor's sensor dataset. That random item is then input into the model for predicting the waypoint. Figure 11 shows the actual location of the watypoint. The model is unable to display the predicted waypoint location as the location is off the map. This is due to the error accumulated during fitting and normalization of data. It should be noted that deep learning model is not able to proceess any character string. Any form of data in string format needs to be converted to numerical format. In the case of *SSID* and *UUID*, they are first encoded to a number. All data are then normalised and fitted with different mean and standard deviation depending on the type of sensor type. The translation of various process accumulates error resulting in an inaccurate prediction of waypoint. Hence the predicted waypoint cannot be plotted on the floor map.

Fig. 11. Predicted waypoint based on wifi data

B. Multi-Modal Machine Learning

Based on the results improvement of combined wifi and geomagnetic sensor data, it can be deduced that any additional information from other type of sensor will improve the accuracy of the model, provided that there is sufficient quantity of data. Other modal type of sensor data namely visual information such as a photograph of the shop which the user visits and audio information such as background music of the shop visited by the user, will definitely improve on the accuracy of the model and enhances the model's prediction. Social networks and platform can be analyzed by matching the timestamp of the tagged photograph taken by the user with the timestamp of the wifi detection can greatly improve the performance of the model.

V. CONCLUSION

This project enables the team to study the different types of sensor data available, their characteristic and organized the sensor data for the task of indoor localization.

The sensor data from the dataset provided by the Indoor-location-competition-21 included accelerometer, gyroscope, rotation vector, magnetic field, gyroscope, wifi, beacon and waypoint. We are able to pre-process the data and use Matplotlib to visualize the data.

We are able to achieve the basic tasks of visualizing the way points (ground -truth locations), visualizing the geomagnetic heat map and the RSS heat map of more than 3 Wi-Fi APS

For Task 1 where we visualized the way points, we observed that the way points are recorded less frequently than the readings for magnetic field and Wi-Fi strength. That leads to the problem of not having enough ground truth data to align to, which causes inaccuracies when generating heat maps. This issue was tackled by computing step positions.

For Task 2, there is the problem of one step positions having more than one magnetic field reading aligned to it. This problem was tackled by using Equation 1 to find the mean magnetic strength for that step position.

For Task 3, We have produced the RSS heat map of 5 maps each floors. From this heatmaps, we can observe the RSSI strength of the location. It can be inferred that those RSSI strength of near to -30dBm show a strong indication of next to AP locations.[4]. E.g. at Site1-B1, BSSID:12:74:9c:2e:92:fe, at x=122.3857,y=2124127 with RSSI -44dBm,at Site2-F8, BSSID:04:40:a9:52:6a:72, at x=159.8725,y=125.655 with RSSI of -22dBM,Site2-F3, BSSID 04:40:a9:a1:95:93, at x=121.4555, y=117.3479 and RSSI=-40dBm.

A deep learning location fingerprint model is developed. Data from wifi, geogmagnetic, beacon sensor and the combined wifi and geomagnetic information are used to train the model. Model trained by the combined wifi and geomagnetic data produced the least error and loss. The trained model is used to predict the waypoint position based on a randomly selected sensor data. However the prediction is not accurate due to the error caused by normalization and fitting of raw data.

VI FUTURE WORK

More work is required to improve on the mode. Effort is required to investigate the margin of error caused by normalization and fitting. Fixing this issue will result in an accurate prediction of the waypoints based on any sensor data.

Another area of research is the usage of non-fully connected neural network. Currently every neuron in one layer is connected to every other neuron in the next layer. This requires considerate computational resources. A non-fully connected network where a neuron in a layer does not need to be connected to all other neurons in the next layer saves significant computation resources required for training.

The other area of research is incorporating other modal information such as visual or audio information. These new information will be able to enhance the model by providing more data for training.

GROUP MEMBER CONTRIBUTUIOONS

The data analysis and pre-processing steps are done collaboratively as a team. Each of the essential tasks are distributed equally among the group members. Brandon Chua Shaojie completed the first essential task, Chang Fuh Yong performed the second essential task and Cheong Kok Yuet did the third essential task. All three members worked on the Deep-learning Location Fingerprinting Model together.

REFERENCES

[1] Biswal, M. (2021, August 13). indoor-location-compeition-20. Retrieved from github: https://github.com/location-competition/indoor-location-competition-20

[2] Michał Nowicki, Jan Wietrzykowski, "Low-effort place recognition with WiFi fingerprints using deep learning," arXiv:1611.02049.

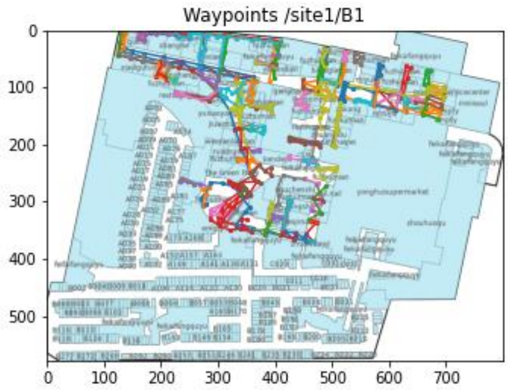
[3] F. Dümbgen et al., "Multi-Modal Probabilistic Indoor Localization on a Smartphone," 2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2019, pp. 1-8, doi: 10.1109/IPIN.2019.8911765.

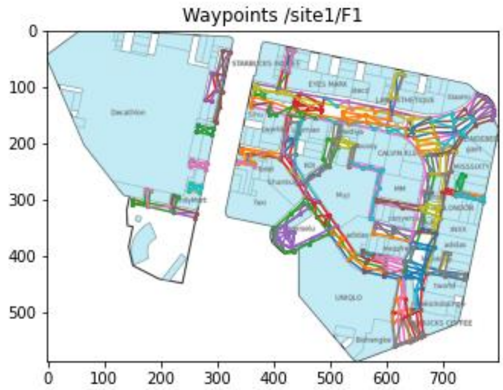
[4] Eyrnetwork: Wi-Fi Signal Strength: What is a Good Signal And How Do You Measure It, https://eyenetworks.no/en/wifi-signal-strength/

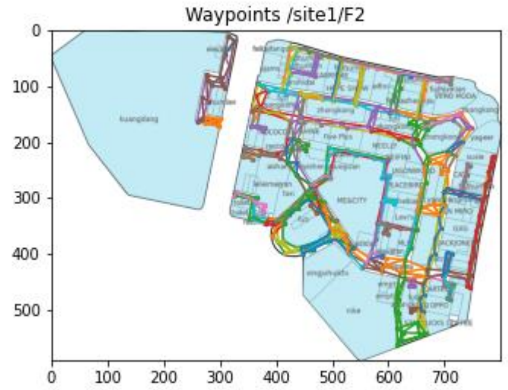
[5] Microsoft Research. (2021, May). Indoor Location & Navigation Retrieved from kaggle: https://www.kaggle.com/c/indoor-location-navigation

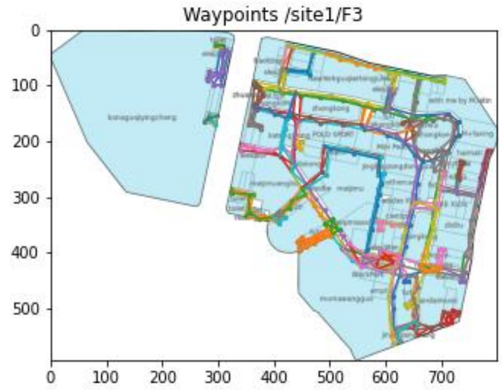
APPENDIX

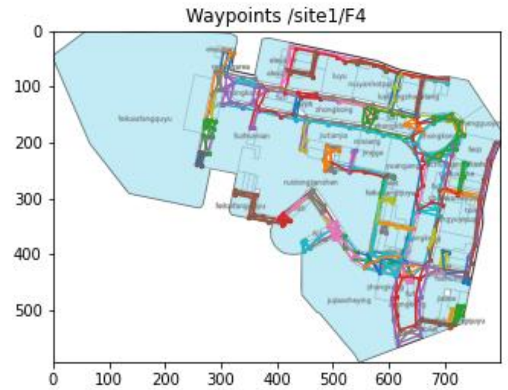
A) *Way points for Site 1*



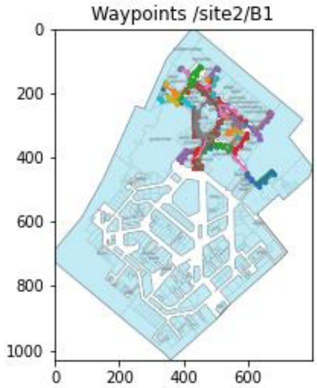


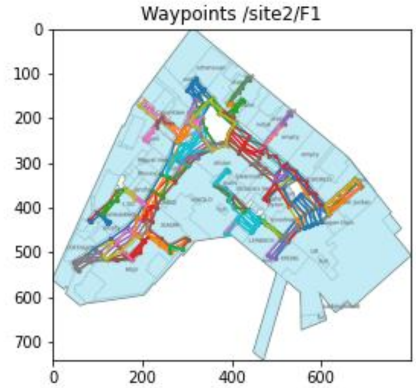


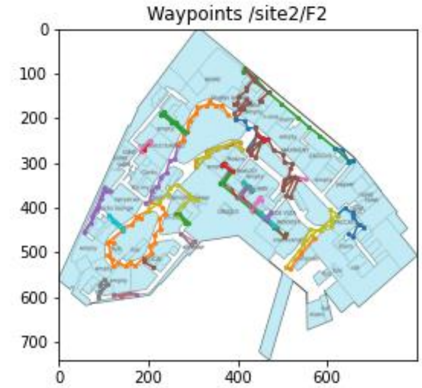


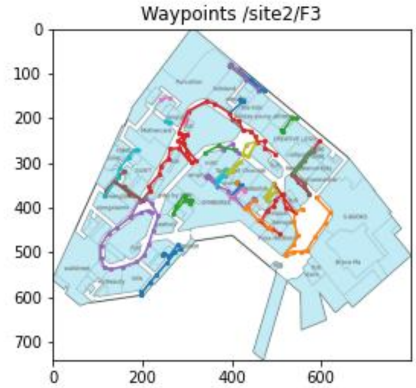


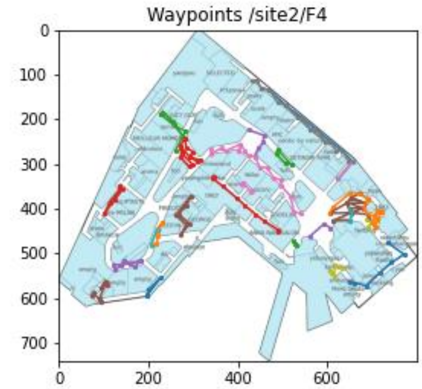
B) Way points for Site 2

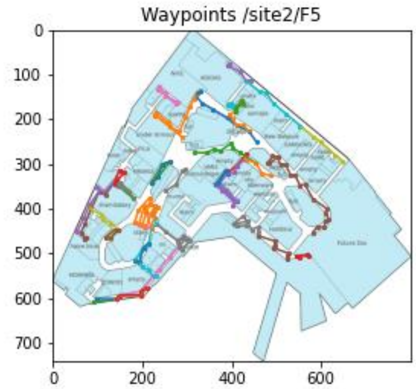


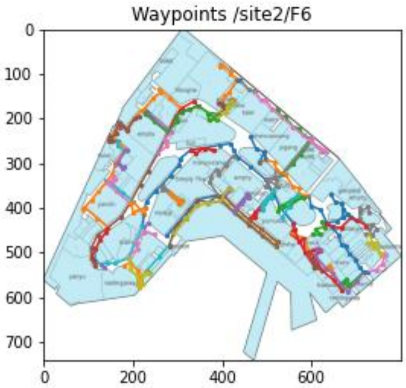


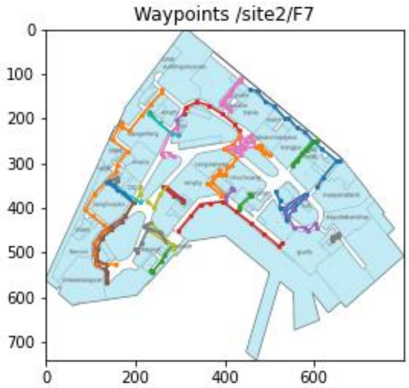


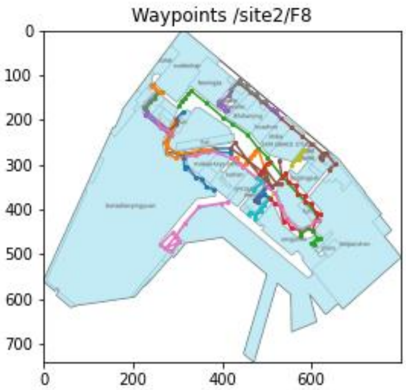




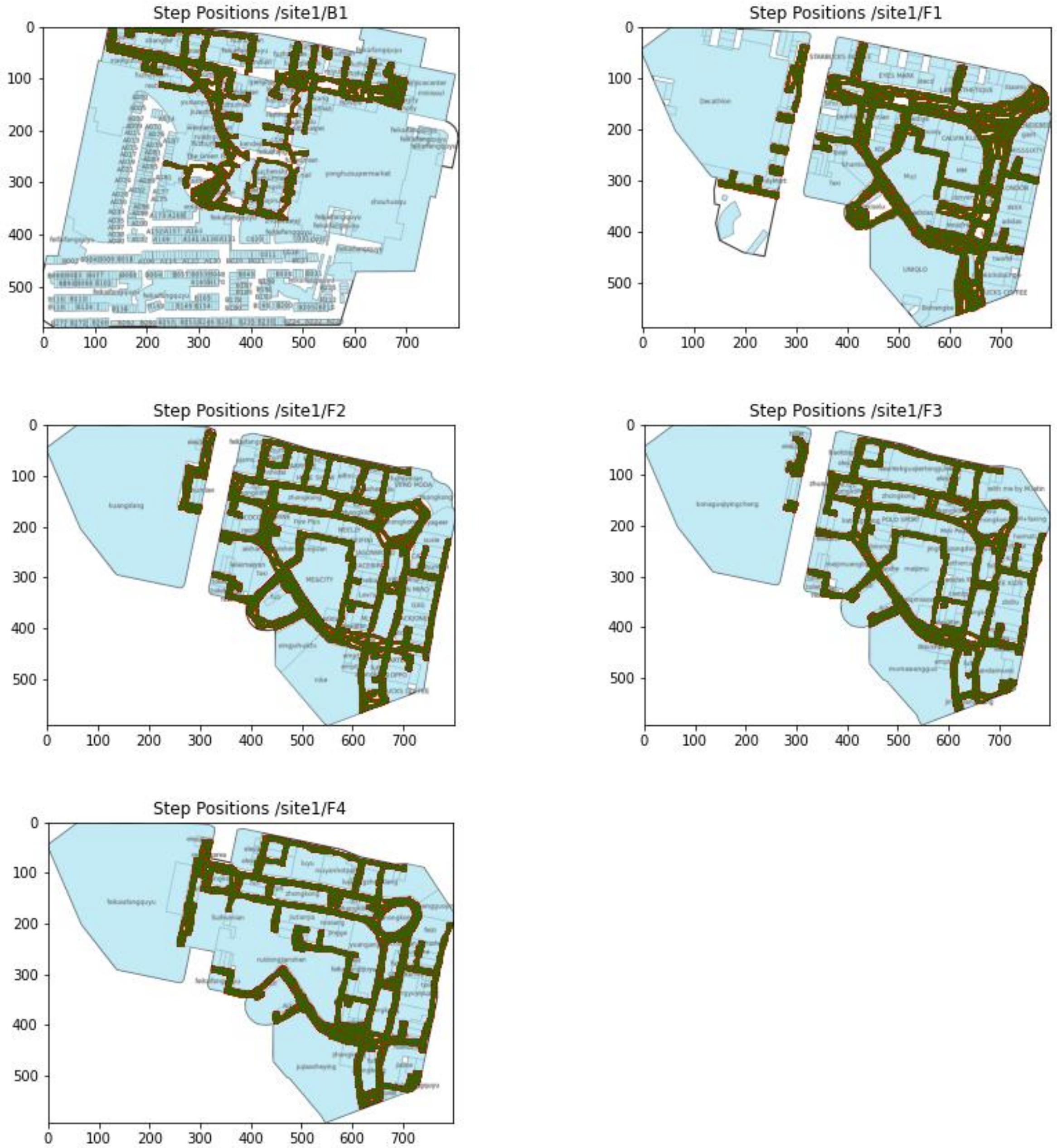



Waypoints /site2/F6
0
100
200
300
400
500
600
700
0
200
400
600

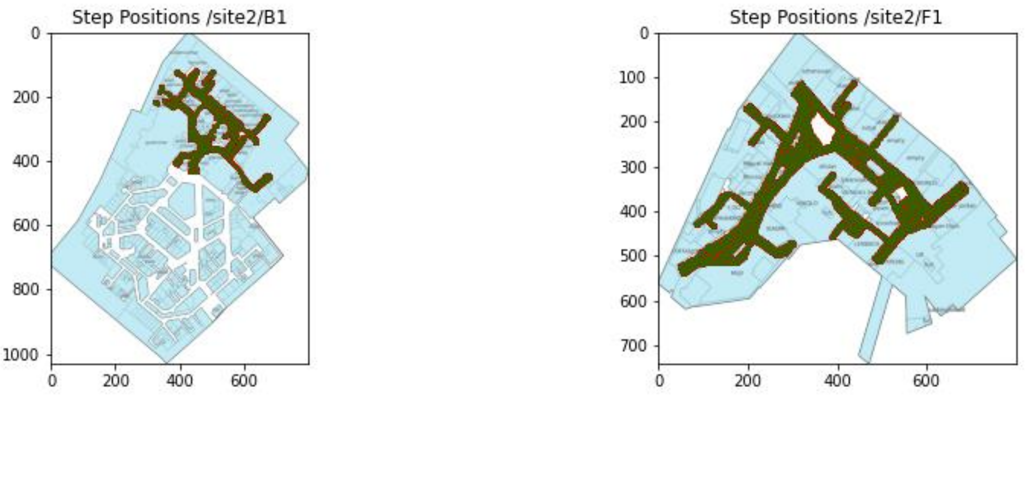

Waypoints /site2/F7
0
100
200
300
400
500
600
700
0
200
400
600


Waypoints /site2/F8
0
100
200
300
400
500
600
700
0
200
400
600

C) Step positions for Site 1

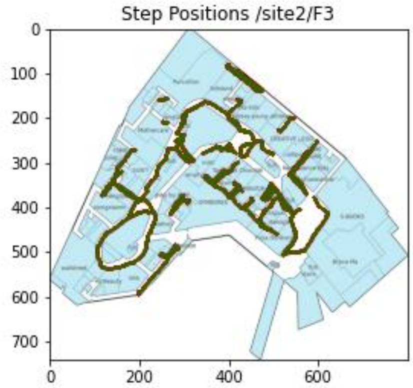


D) Step positions for Site 2

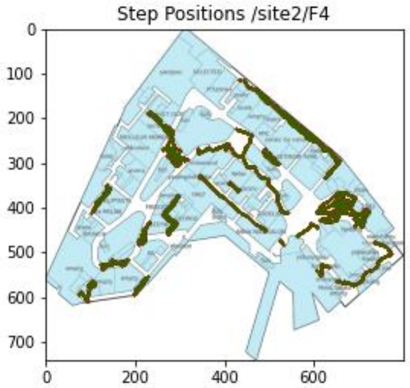


Step Positions /site2/F2

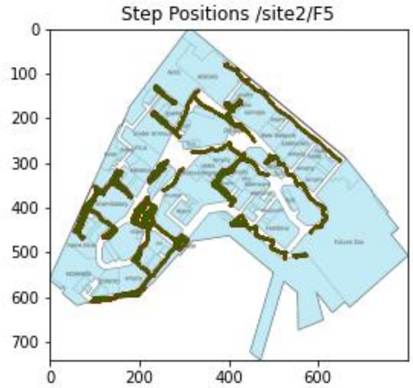
Step Positions /site2/F3



Step Positions /site2/F4

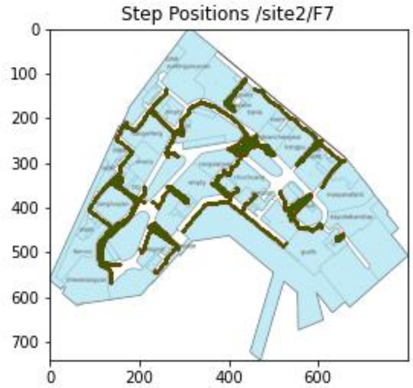


Step Positions /site2/F5

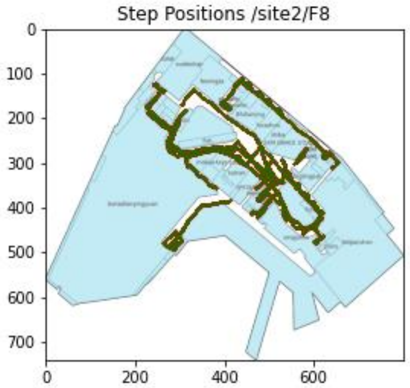


Step Positions /site2/F6

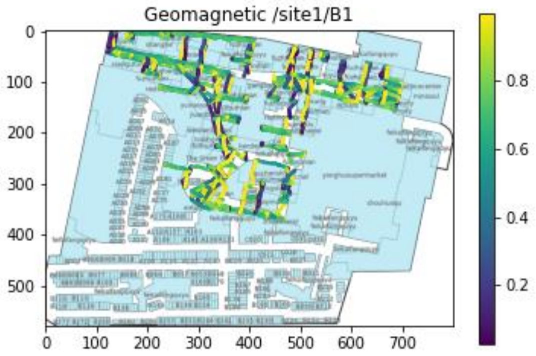
Step Positions /site2/F7

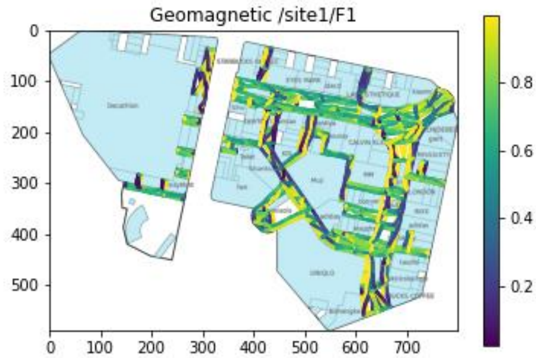


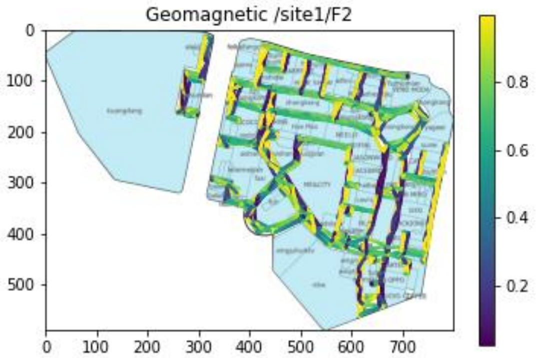
Step Positions /site2/F8

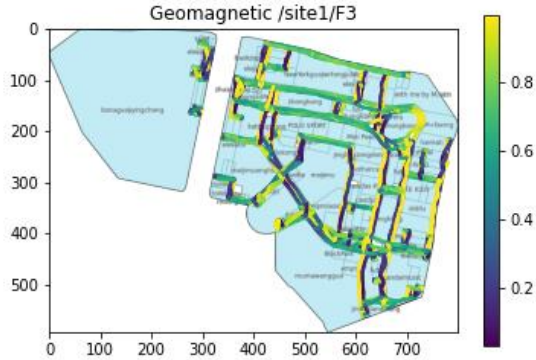


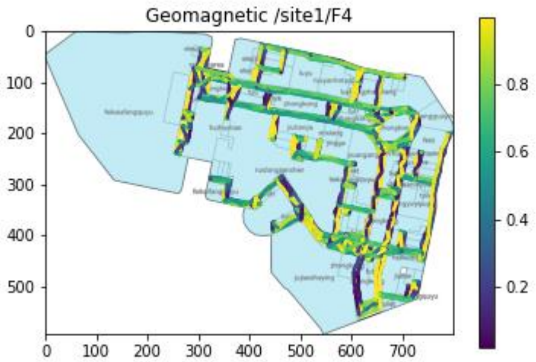
E) Geomagnetic heat maps for Site 1



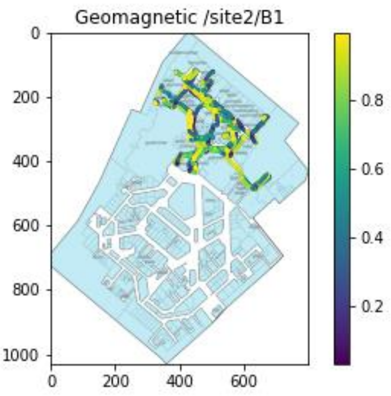


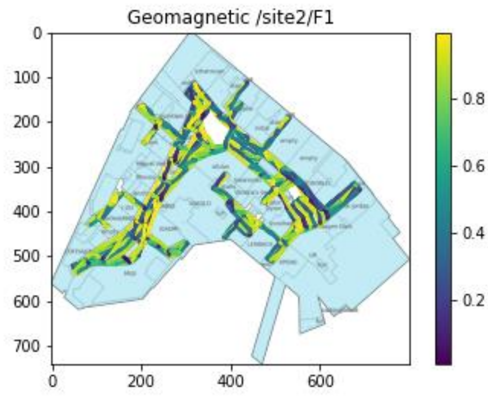




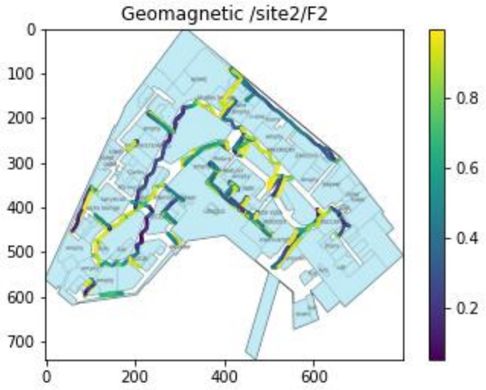


F) Geomagnetic heat maps for Site 2

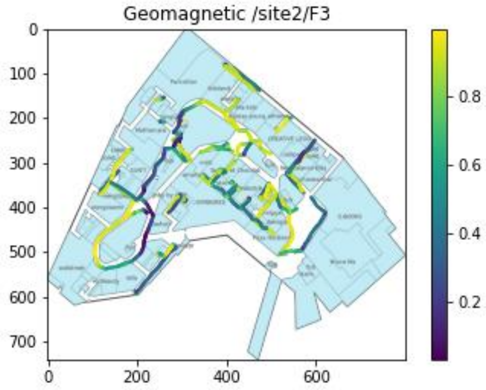




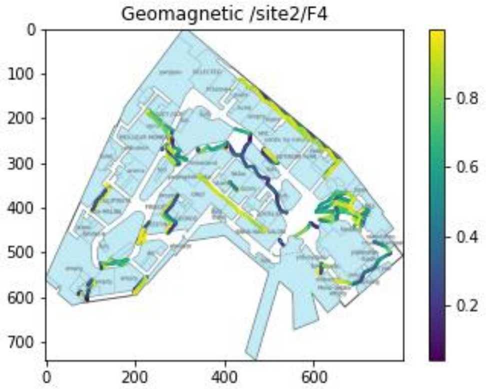
Geomagnetic /site2/F2



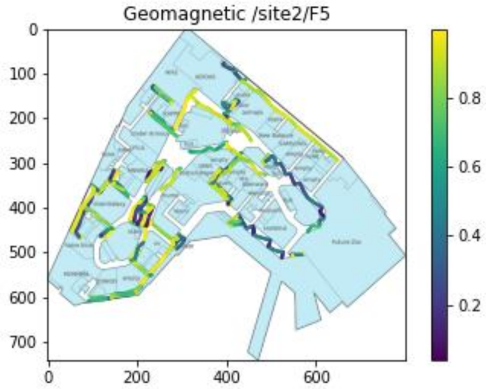
Geomagnetic /site2/F3



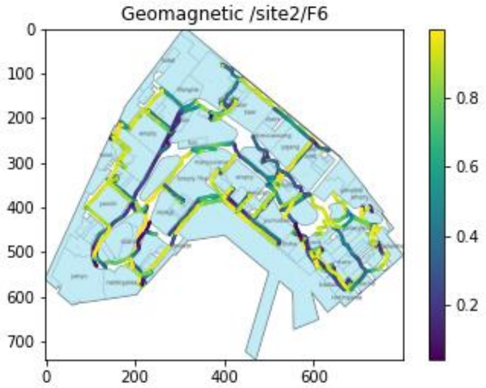
Geomagnetic /site2/F4



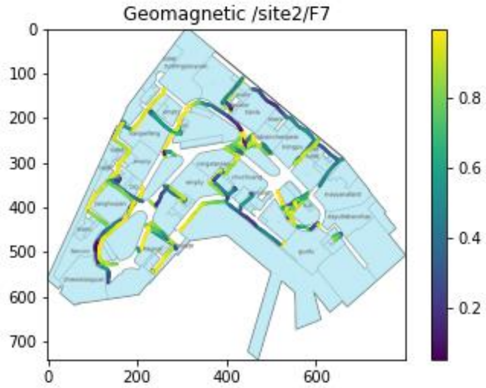
Geomagnetic /site2/F5



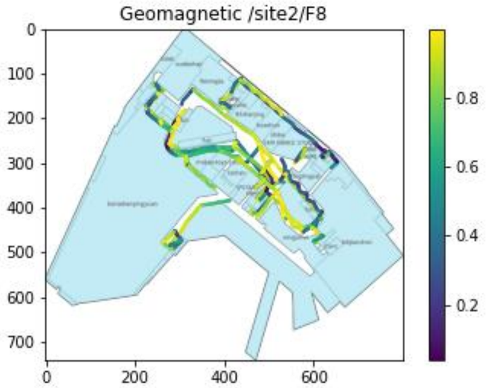
Geomagnetic /site2/F6



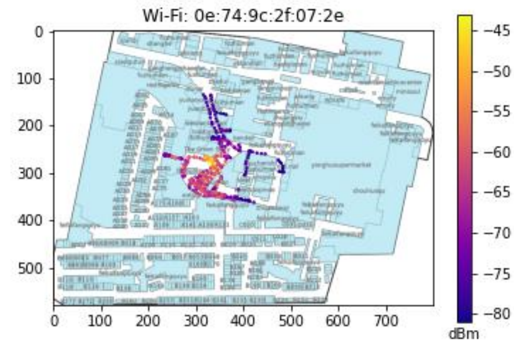
Geomagnetic /site2/F7



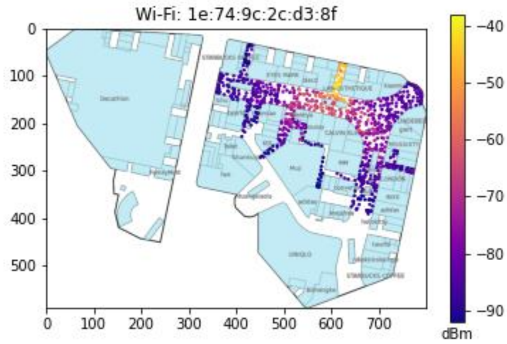
Geomagnetic /site2/F8



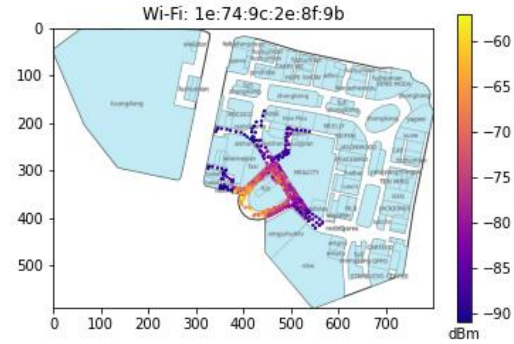
G) Wi-Fi RSS heat map of an access point from each floor for Site 1



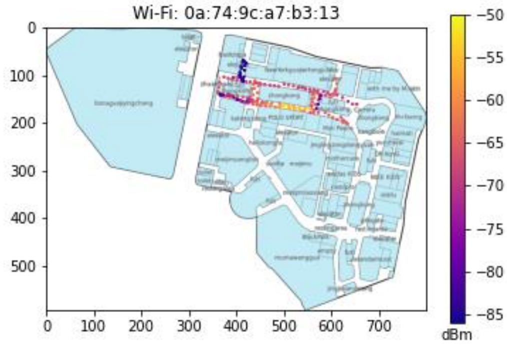
Basement 1



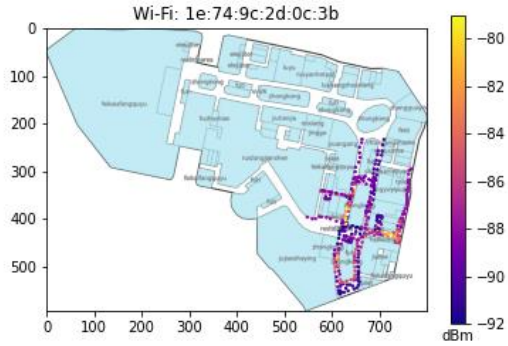
Floor 1



Floor 2

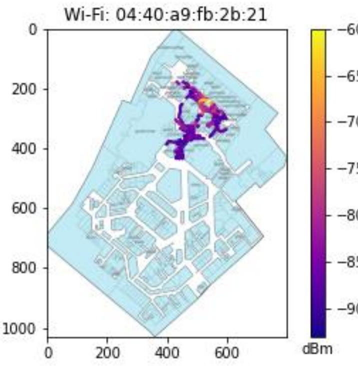


Floor 3

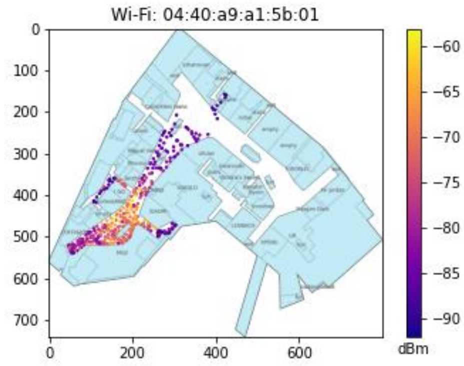


Floor 4

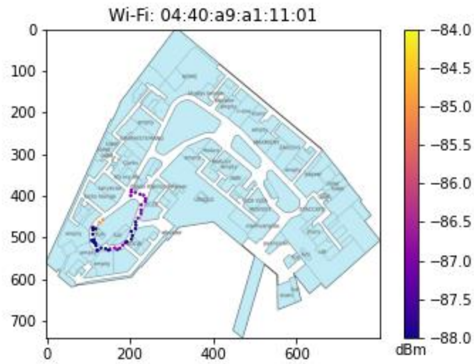
H) Wi-Fi RSS heat map of an access point from each floor for Site 2



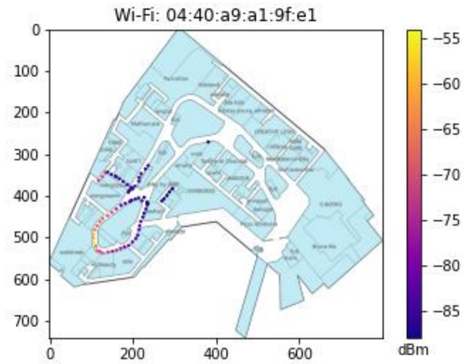
Basement 1



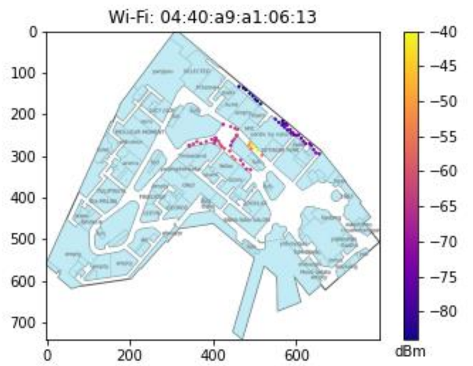
Floor 1



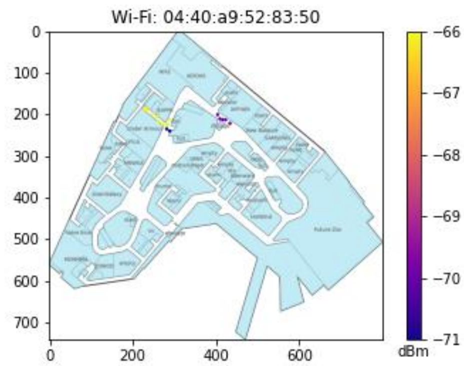
Floor 2



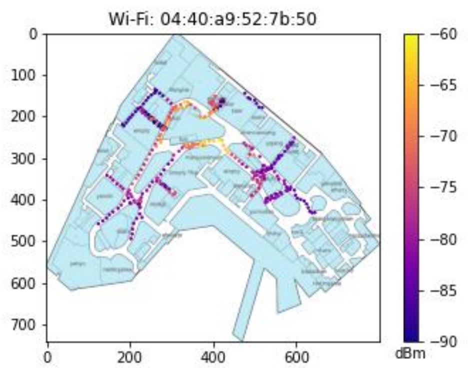
Floor 3



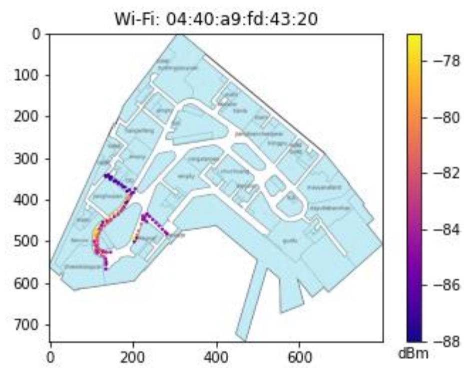
Floor 4



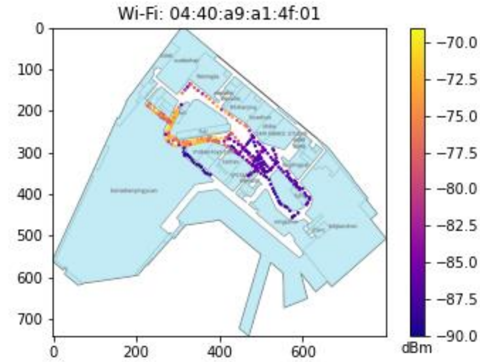
Floor 5



Floor 6



Floor 7


Wi-Fi: 04:40:a9:a1:4f:01
0
100
200
300
400
500
600
700
0
200
400
600
−70.0
−72.5
−75.0
−77.5
−80.0
−82.5
−85.0
−87.5
−90.0
dBm

Floor 8