

Report for Urban Computing project1

Xiang Xinye
SCSE

Nanyang Technological University
Singapore
XI0001YE@e.ntu.edu.sg

Tan Jie Heng Alfred
SCSE

Nanyang Technological University
Singapore
TANJ0307@e.ntu.edu.sg

Yin Wenqi
SCSE

Nanyang Technological University
Singapore
S220157@e.ntu.edu.sg

Abstract—In the rapidly evolving landscape of urban computing, the challenge of indoor localization using smartphones stands out as a critical research area. This paper documents our experience with AI6128 Urban Computing Course Project 1, which delves into the practical implementation of indoor localization using a dataset from Microsoft Indoor Location Competition 2.0. By harnessing diverse smartphone sensors such as WiFi, geomagnetic, and inertial sensors, we explored various methodologies to preprocess, visualize, and analyze spatio-temporal data. Essential tasks covered include the visualization of ground-truth waypoints, geomagnetic heatmaps, and WiFi RSS heat maps for three access points. Beyond these core objectives, we embarked on advanced explorations, notably developing a deep learning-based location fingerprint model. Our work not only reinforces understanding of sensor modalities but also illuminates the challenges and potential solutions for indoor localization. Our findings have practical implications for various applications, ranging from emergency services to entertainment.

Index Terms—Indoor Localization, Smartphone Sensors, Geomagnetic Heatmap, Data Preprocessing, Deep Learning-Based Localization

I. BACKGROUND

A. Introduction

Localization is an important aspect of urban computing. The two types of localization are outdoor localization and indoor localization. The former can be addressed by the well-known Global Positioning System (GPS). However, the latter cannot be well-addressed by the same system due to the low energy capacity of satellites. Instead, there are generally two other schemes of wireless indoor localization: device-based and device-free localization.

For the former, the target needs to have a device on them, and their location is calculated based on the interaction between the target's device and the wireless devices deployed within the building. As for device-free localization, the target is not required to possess any device. Instead, the location is determined based on the target's (indirect) interaction with the deployed wireless sensors. The aim of these two schemes is the same – to accurately locate the target through wireless means. The location returned by these two schemes could take different forms, such as the symbolic location and absolute coordinates of the target's physical location.

B. Objectives

In this project, our team will analyze various signals from smartphones to accurately determine the absolute coordinate of a target's physical location. Following this, we will visualize the target's waypoint, the geomagnetic heatmap in different locations, and the WiFi RSS heatmap. With the fingerprints, we were able to create a deep learning model to predict the location of the target using the input features more accurately.

II. DATASET

A. Data Information

The sample dataset used in this project was a subset of the dataset used in Microsoft Indoor Location Competition 2.0 [0]. The original, full dataset contains various signals such as WiFi signatures, geomagnetic field data, and iBeacons data. These, together with the ground truth way point data, were collected from 2,718 floors of 300 buildings in 3 Chinese cities. The sub-dataset used in this project contains only the data from two of these buildings, denoted as site1 and site2 respectively. These data are collected by an Android smartphone, which was attached to a site-surveyor, as the site-surveyor walks around the site building.

The dataset used in this project contain trace files, which are saved and grouped according to the site they were collected, and further grouped into the floor which they were collected from. Together with these trace files are other metadata such as the size information (height and width).

Each of the trace files are text files. The data are formatted as TimeData TypeValue, where Time values are in Unix Time (milliseconds), Data Type values are the type of data collected (e.g., accelerometer data), and Value values describe the associated data type. The table below shows all the data type and the format of their corresponding values.

B. Dataset Analysis

Prior to working with the data, our team performed a brief exploratory data analysis. This helps us better understand the dataset before processing them. We detail some of the findings we had from this analysis.

Firstly, we noticed that the number of ground truth waypoints (i.e., *TYPE WAYPOINT*) is significantly fewer than the other data collected from the smartphone. For example, in one of the trace text files obtained from site1, B1, there are 5,604 geomagnetic and accelerometer data logged, as compared to

TABLE I: Data type and values

| S/N | Data Type | Values (Space delimited) |
|-----|----------------------------------|--|
| 1 | TYPE_WAYPOINT | P_x, P_y |
| 2 | TYPE_ACCELEROMETER | X, Y, Z, accuracy |
| 3 | TYPE_GYROSCOPE | X, Y, Z, accuracy |
| 4 | TYPE_MAGNETIC_FIELD | X, Y, Z, accuracy |
| 5 | TYPE_ROTATION_VECTOR | X, Y, Z, accuracy |
| 6 | TYPE_ACCELEROMETER_UNCALIBRATED | $X_b, Y_b, Z_b, X_a, Y_a, Z_a$, accuracy |
| 7 | TYPE_GYROSCOPE_UNCALIBRATED | $X_b, Y_b, Z_b, X_a, Y_a, Z_a$, accuracy |
| 8 | TYPE_MAGNETIC_FIELD_UNCALIBRATED | $X_b, Y_b, Z_b, X_a, Y_a, Z_a$, accuracy |
| 9 | TYPE_WIFI | ssid, bbsid, RSSI, frequency, lastseen |
| 10 | TYPE_BEACON | UUID, MajorID, MinorID, Tx Power, RSSI, Distance, MAC, Unix Time |

15 for the ground truth data. This disparity results from the ground truth waypoint being manually collected by the site-surveyor, as opposed to the other data types which were automatically logged by the smartphone's sensors. As a result, the waypoint visualization with only the ground truth data would be very sparse. In order to obtain a denser waypoint visualization, some data augmentation would be required.

Another observation we made was the fact that several of the sensor readings seem to share the same timestamps. For example, the accelerometer, gyroscope, rotation vector, and geomagnetic field data seem to share the same time stamps. This includes all the uncalibrated counterparts. This means that these readings are logged at the same time. On the other hand, the WiFi data are logged at different time, as suggested by the different timestamps. In fact, the number of WiFi data is different from the number of the other data types. This is because of the presence of various WiFi APs, corresponding to different SSIDs. Each of these WiFi APs are recorded as separate entries, with their corresponding RSSI and SSID. Lastly, it is important to note that the timestamps of the various sensor readings do not match the timestamp of the recorded waypoints directly. This suggests that certain approximations and assumptions have to be made when pre-processing the various data.

C. Data Preprocessing

As mentioned in the preceding section, the sparsity of the ground truth data warrants for the need to augment the data. Intuitively, we are trying to interpolate between successive waypoint data, but using other sensor modalities. As discussed above, the data logged by the accelerometer is dense relative to the waypoint data, we will use them to estimate the positions between successive waypoint data.

To do so, we first predict when steps are taken by the surveyor, using the acceleration data (i.e., *TYPE_ACCELEROMETER*). The algorithm to do so is

described in [0]. Firstly, the acceleration magnitude is calculated from *TYPE_ACCELEROMETER*, using L2-norm. Next, a low-pass filter (i.e., Butterworth Filter) is applied to remove high-frequency noise and smooth the acceleration magnitude. With the smoothed acceleration magnitude, a peak detection algorithm is used to identify the local maxima and local minima from this filtered signal. Each local maxima is associated to a step taken by the surveyor. Then, we can calculate the stride length between two steps is calculated using the parameters found in [0], while also accounting for the peak-to-peak amplitude of the acceleration magnitude.

Next, we use the values from *TYPE_ROTATION_VECTOR* to find the orientation of the device (i.e., surveyor) in the 2D plane. This tells us the direction which the strides are taken. This, together with the calculated stride length, allows us to determine the relative positions of the surveyor, with respect to the previous position. In other words, we could determine the position of the surveyor at time t , denoted by P_t , in relation to the surveyor's position at time $(t - 1)$ as follows:

$$P_t = P_{t-1} + S_t O_t$$

where S_t is the stride length and O_t is the orientation vector, both at time t . At this point, we will store only $S_t O_t$ as an array, together with the timestamp. We are able to do this because both *TYPE_GYROSCOPE* and *TYPE_ACCELERATION* share the same timestamp as noted previously.

Lastly, we will use the ground truth waypoints to determine the surveyor's absolute position. To do so, let WP_T and WP_{T+1} be two successive waypoint data. Then, let $\{RP_j^{(T)}\}_{j=1}^R$ be the set of $S_t O_t$ whose timestamps are between the timestamps of WP_T and WP_{T+1} , inclusive. The terms in $\{RP_j^{(T)}\}_{j=1}^R$ are ordered in increasing timestamp. Then, we can 'interpolate' the positions, denoted as $\{AP_i^{(T)}\}_i$, as follows,

$$[AP_i^{(T)}] = \begin{cases} WP_T + RP_1, & \text{if } i = 1, \\ AP_{i-1}^{(T)} + RP_i, & \text{if } 1 < i \leq R. \end{cases} \quad (1)$$

Then, to ensure a smoother 'interpolation', we will add an offset to each $AP_i^{(T)}$, while considering that the final position should be WP_{T+1} . These offset $AP_i^{(T)}$ will be our augmented waypoints between timestamps of WP_T and WP_{T+1} .

III. ESSENTIAL TASKS

A. Visualization of waypoints

In this visualization process, we introduced two types of visualization of ground truth locations. One is using normal waypoints while the other is the augmented version of step positions which is used in model training.

For visualizing normal way points, we do the following: Firstly append all the waypoints into an array, and paint them on the floor plan picture according to the coordinates, with reference to the height and width of the map given by the json file. Figure 1 shows the result from the visualization.

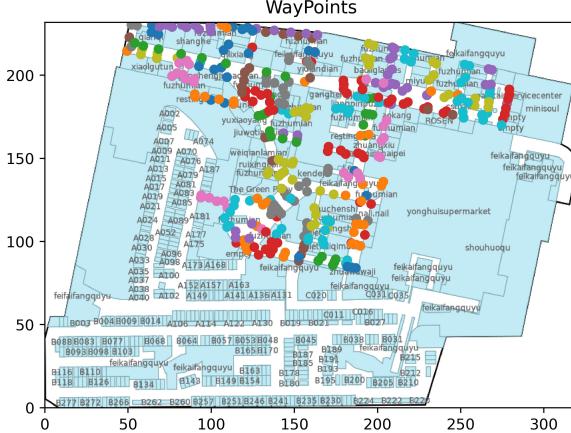


Fig. 1: Way points visualization

As can be seen from Figure 1, the waypoints are too sparse. To have a better visualization and denser labels, we will have to perform augmentation on the waypoint data as explained in Section I(C).

Upon implementing the augmentation, we have a denser visualization as seen in Figure 2.

The augmented waypoints visualization is more difficult. The waypoints are too sparse to use as train data. So it is necessary to use other information collected to detect more coordinates pairing with timestamps that we can use. We use acceleration data collected by accelerometer to do it. Firstly, we calculate the magnitude of acceleration to decide if there can be a step. After that, we choose reasonable step criterion and check if there is a valid step. If there is, then save the coordinates and the time stamp information. Then, using the data to compute stride length by computing step period before times the acceleration magnitude, which can lead us to real positions. Then we combine them with waypoints collected by the phone, and we have our augmented waypoints data. 2 is how it looks like in a map:

B. Geomagnetic Heat Map

The geomagnetic readings are logged more frequently than the manually collected waypoint readings. This is likely because the geomagnetic sensor of the phone continuously collects the readings in the background, whereas the waypoint data are collected by the user.

Despite being denser, the geomagnetic readings do not explicitly capture the location of the user. Therefore, there is a need to calibrate the geomagnetic readings with the waypoint data to localize the user. To this end, we first split the geomagnetic data in a file, into non-overlapping subarrays, where each subarray contains geomagnetic data with the same timestamp. Let the j^{th} subarray be $MG_j = \{(t_j^{mag}, x_{j,i}^{mag}, y_{j,i}^{mag}, z_{j,i}^{mag})\}$, for $1 \leq i \leq |MG_j|$. We can then estimate the indoor position, WP_j^{mag} , of each MG_j by comparing it to the augmented

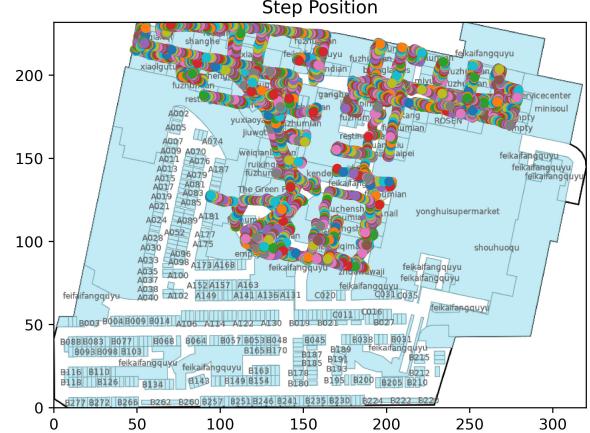


Fig. 2: Augmented waypoints visualization

waypoint data, $WP_s = (t_s^{WP}, x_s^{WP}, y_s^{WP})$. We do this by comparing their timestamps:

$WP_j^{mag} = \{x_k^{WP}, y_k^{WP} : k = \arg \min |t_s^{WP} - t_j^{mag}|\}$, where j is again fixed. This means that we take WP_j^{mag} to be the position of WP_k , where the difference between the timestamp of MG_j and the timestamp of WP_k is the smallest, among all the waypoint data.

Note that there is a possibility that $|MG_j| > 1$, for some j , or $WP_a^{mag} = WP_b^{mag}$ for some a, b such that $a \neq b$. In other words, there could be scenarios where each indoor position has more than one geomagnetic reading associated to it. To visualize a geomagnetic heat map, where each position is associated to one geomagnetic strength, we will have to aggregate all these geomagnetic data for each of the position. We will do this by using the mean L2-norm.

Suppose for some k , the indoor position associated to each $MG_{r_1}, MG_{r_2}, \dots, MG_{r_q}$, for some r_1, r_2, \dots, r_q , is (x_k^{WP}, y_k^{WP}) (i.e., $WP_{r_1}^{mag} = \dots = WP_{r_q}^{mag} = \{(x_k^{WP}, y_k^{WP})\}$). Then, the magnetic strength associated to (x_k^{WP}, y_k^{WP}) , define it by MS_k , is given to be:

$$MS_k = \frac{1}{\sum_{s=1}^q |MG_{r_s}|} \sum_{s=1}^q \sum_{i=1}^{|MG_{r_s}|} \sqrt{(x_{r_s,i}^{mag})^2 + (y_{r_s,i}^{mag})^2 + (z_{r_s,i}^{mag})^2} \quad (2)$$

In other words, we take the L2 norm of every geomagnetic data with indoor location (x_k^{WP}, y_k^{WP}) , then find the mean of all the L2 norm.

Each MS_k is an aggregation of the geomagnetic strength from each of the geomagnetic data associated to (x_k^{WP}, y_k^{WP}) . With these, we can create a heatmap, with different hue indicating the strength of the geomagnetic intensity (red hue shows an area with highest geomagnetic strength, while violet hue shows an area with the weakest geomagnetic strength). Figure 3 shows one such geomagnetic heat map for level 3 of site1.

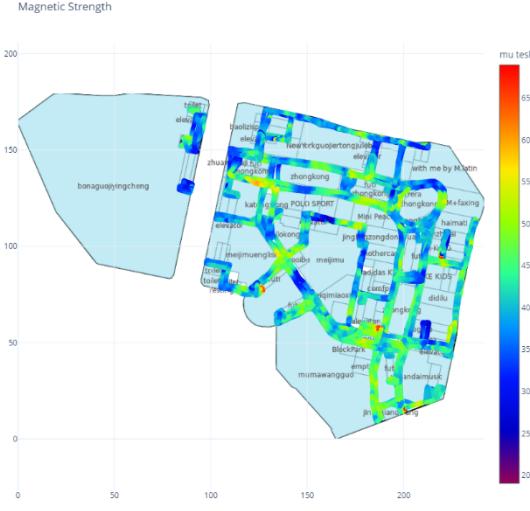


Fig. 3: Geomagnetic heat map for site1, Floor 3, mapped to augmented way points.

To obtain the collection of $MG_{r_1}, MG_{r_2}, \dots, MG_{r_q}$ during the data-processing step, we maintain a dictionary of key (x_k^{WP}, y_k^{WP}) and append $MG_{r_1}, MG_{r_2}, \dots, MG_{r_q}$ as the values. This is done concurrently with finding WP_j^{mag} .

C. Visualize Wi-Fi RSS heat maps of 3 Wi-Fi Ap

1) *Approach Description:* Similarly, we will visualize the RSS (Received Signal Strength) of Wi-Fi APs, as a heatmap. The goal is to map each Wi-Fi instance, denoted as $WIFI_i = (bssid_i, t_i, r_i)$, to its corresponding location on the floor plan. Here:

- r_i represents the Wi-Fi signal strength.
- $bssid_i$ denotes the BSSID of the AP.
- t_i signifies the timestamp when this instance was recorded.

To associate each instance with a physical location, we localize the $WIFI_i$ instance by comparing its timestamp t_i with all timestamps t_{WP_j} from the ground truth waypoints. This is handled the same way as the calibration of the geomagnetic data describe in Section II(B).

We assign the waypoint position (P_x, P_y) whose timestamp is the closest to t_i . Mathematically, this is represented as:

$$WIFI_{bk} = \{(r_i, t_i, bssid_i) \mid k = \arg \min_j |t_{WP_j} - t_i|\} \quad (3)$$

After determining the positions of all $WIFI_i$ instances, we may find multiple RSSI records r_i for a unique AP with BSSID b at a specific position. To consolidate these readings, we compute the average RSSI for that location. The formula for computing this average is:

$$WP_{rssk}^b = \frac{1}{N} \sum_{WIFI_{bk}} r_i \quad (4)$$

where WP_{rssk}^b represents the average RSS of an AP with BSSID b at an augmented waypoint during timestamp k , and N is the total number of $WIFI_i$ instances recorded at timestamp k for BSSID b .

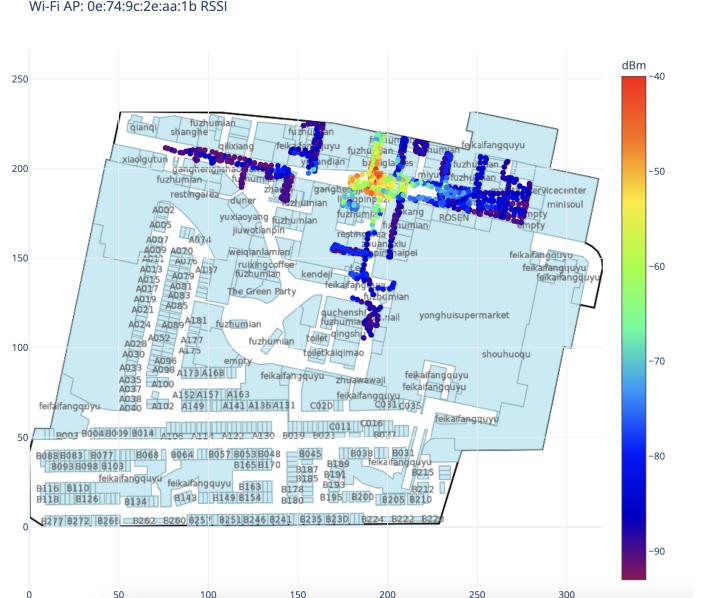


Fig. 4: Wi-Fi heat map for site1-B1

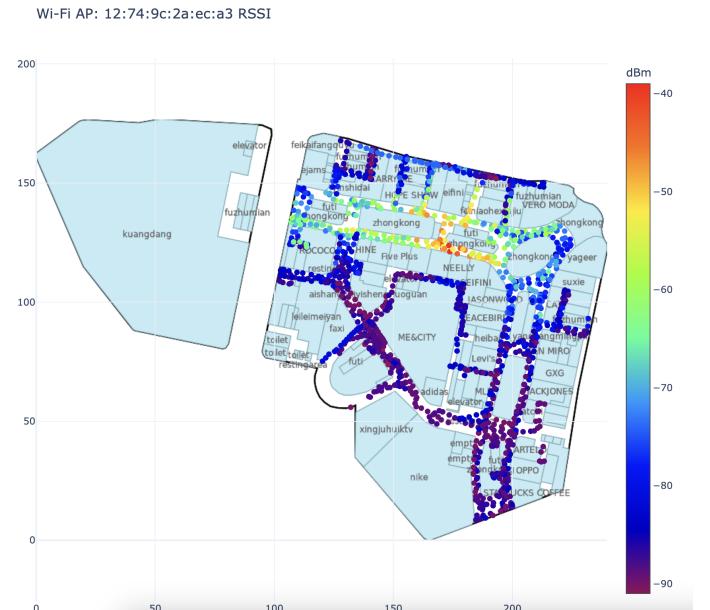


Fig. 5: Wi-Fi heat map for site1-F2

2) Result Presentation:

a) *Description of the Heat maps:* In these heat maps, locations where Wi-Fi signals were recorded are represented as dots. The color of each dot indicates the absolute value of the Wi-Fi signal strength at that location. Lighter shades, such as red or light green, represent strong signals, while darker shades, like deep green, represent weaker signals.

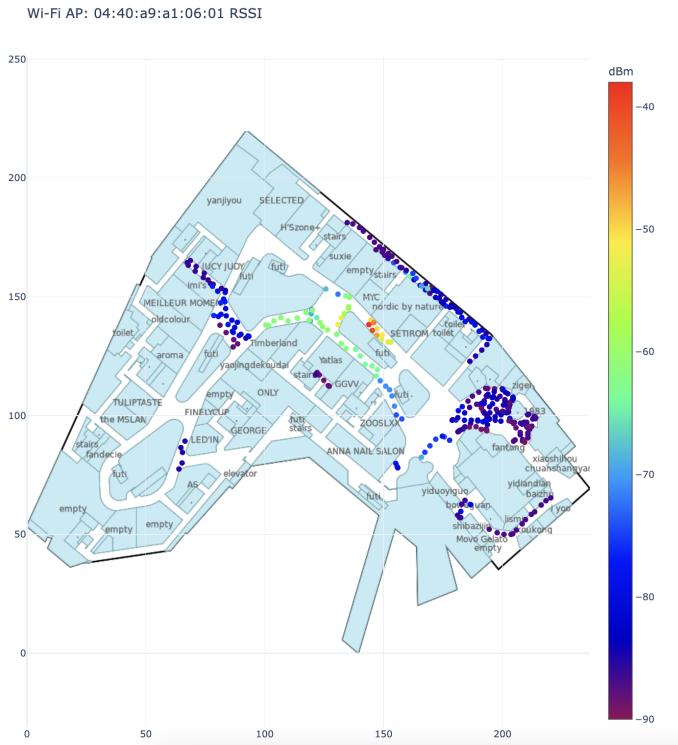


Fig. 6: Wi-Fi heat map in site2-F4

The distribution of colors across the heat map provides insights into the signal distribution of the Wi-Fi APs. Areas with lighter shades are likely closer to the Wi-Fi APs or have fewer obstructions, leading to stronger signals. Conversely, darker areas might be farther from the APs or have obstructions, resulting in weaker signals.

This visualization technique is valuable as it not only helps in identifying potential dead zones but also aids in optimizing the placement of new APs for better coverage.

IV. BONUS TASKS

A. Build a deep learning-based location fingerprint model

In the bonus task, we attempt to use VGG16 to refine the localization prediction using the inputs from the various sensor modalities. VGG is a well-known convolution-based architecture that has demonstrated strong performance in a variety of image classification tasks. Although indoor localization is not an image classification task, we can still use CNN-based models for this task, as demonstrated in [0] and [0].

In this project, we will transfer a pre-trained VGG16 model to predict the position coordinates from the average Wi-Fi RSSI value and geomagnetic strength, found in Section III(B,C). We also included iBeacon RSSI as our input to provide finer details for the model to learn from. This is therefore a regression problem and the mean-squared error (MSE) is used as our loss function.

The pre-trained VGG16 is implemented as a PyTorch network class. The model consists of two parts: A features extractor followed by a fully connected part. The former aims

to learn useful features from the input data, while the latter combines these learned features to predict the location. For our model, we initialized the weights of the feature extractor from the pre-trained model, which consists of convolutional and pooling layers. We then replaced the fully connected part of the pre-trained model with several fully connected layers using ReLU as our intermediate activation functions. The final output layer is linear without ReLU, with user-specified output size. In our case, since we are predicting the indoor position at a particular site and floor, (P_x, P_y) , we will have an output size of 2.

To obtain the training data, we append the geomagnetic strength, Wi-Fi RSSI value, and iBeacon RSSI value, all sharing the same associated augmented waypoint. These form the input data, and the label for each datapoint will be the augmented waypoint. Note that these information are obtained from previous sections. Again, we used the augmented waypoint as our labels because of the sparsity of the ground truth waypoint. Since we are training a supervised model, more datapoints are required for a more robust model.

After obtaining these datapoints, we split the dataset using a 8:2 train-test split ratio. Therefore, 80% of the original dataset will be used for training of the model while 20% will be for evaluating the model. For the training of our model, we ran it for 50 epochs, with stochastic gradient descent as our optimizer, initial learning rate of 0.01 and momentum of 0.9. We also adopted the StepLR as our learning rate schedule with gamma of 0.5 and step size of 3.

After training 50 epochs, we ran the model that was trained on our input data on a holdout test set. The model achieves a test loss of 0.024 under MSE.

As an illustration, we visualized the error heat map as shown in Figure 7. The error heat map gives a proxy of how accurate our model is at predicting the indoor location; the lower the loss, the more accurate it is at predicting the augmented waypoints. Interestingly, the area with the lowest loss is near the centre of the floor level, whereas the extremities have higher losses. A potential reason for this is the higher degree of attenuation and noisiness of some signals, like the Wi-Fi signal, across further distance. The extremities of the floor are likely the furthest from the Wi-Fi APs, hence the Wi-Fi signal would be greatly attenuated and noisy, resulting in a noisier input data, therefore poorer prediction by the model.

B. Future work

Our group used VGG16 as our model of choice in this prediction task. Although the network proves to be effective in predicting indoor location using readings from various sensor modalities, there are improvements that could be made. For instance, the input data could have been pre-processed differently. Using the geomagnetic strength as an example, we aggregated all the magnetic strength (as L2 norm of the geomagnetic reading) with the same augmented waypoints as a pre-processing step. However, this would result in some information loss, as very different geomagnetic readings (with different (x, y, z) coordinates) could have similar geomagnetic

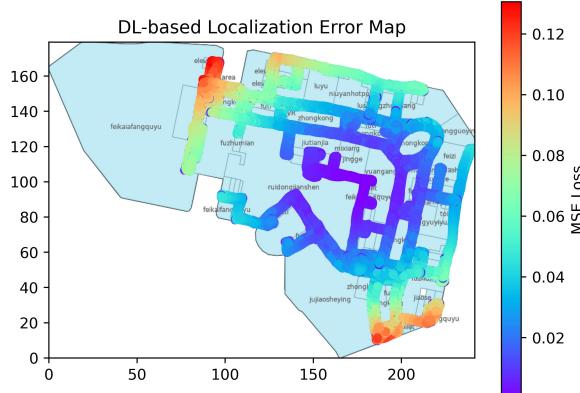


Fig. 7: Geomagnetic heat map mapped to augmented way points.

strength (similar L2 norm). Therefore, instead of manually pre-processing these raw information, we could get the model to represent this using an autoencoder. Here, we include an autoencoder before the convolution step to get the model to learn a representation of the raw data in a more effective manner for the indoor localization task. This type of model is similar to the one seen in [0].

V. CONCLUSION

Having worked with the Microsoft Indoor Localization dataset, our team now has a better understanding on how to make sense of different sensor data, the limitations each of the data type have, and how to pre-process these data to produce a fingerprinting model. We visualized these as heatmaps in the report. Finally, we used these pre-processed data, aggregate them as input data, and trained a deep-learning, convolution-based, VGG16 model for predicting the indoor location of a device. However, we also note that there are areas that can be improved on.

VI. GROUP MEMBER CONTRIBUTIONS

A. Contributions by Xiang Xinye

I contribute to this assignment by writing the code to visualize waypoints and augmented way points onto the map as well as the related part of the report. Moreover, I set up the network used to train the finger print model, with the part of the report.

B. Contributions by Yin Wenqi

First, I analyzed the data, and completed to visualize 3 wifi heatmaps. Next, in bonus tasks, I processed all the raw data, and merged it to fit the deep learning-based model using PyTorch.

C. Contributions by Tan Jie Heng Alfred

I worked on the initial exploration of the dataset, and noting of the sparsity of some of the data types. Then, I worked on the visualization of the geomagnetic data, including the working of the augmented way points. Lastly, I worked on the bonus task section of the report together with Xinye.

APPENDIX

A. Geomagnetic Heatmaps

The geomagnetic heatmaps for all the floors of site1 and B1 of site2 can be found in Figure 8, while the geomagnetic heatmaps for all the floors, except B1, of site2 can be found in Figure 9.

B. Ground Truth Visualization

The waypoint visualization for all the floors of site1 and B1 of site2 can be found in Figure 10, while the waypoint visualization for all the floors, except B1, of site2 can be found in Figure 11.

The step position visualization for all the floors of site1 and B1 of site2 can be found in Figure 12, while the step position visualization for all the floors, except B1, of site2 can be found in Figure 13.

C. WiFi RSSI Visualization

The WiFi RSSI visualization for all the floors of site1 and B1 of site 2 are found in Figure 14, whereas the WiFi RSSI visualization for all the floors, except B1, of site2 can be found in Figure 15.

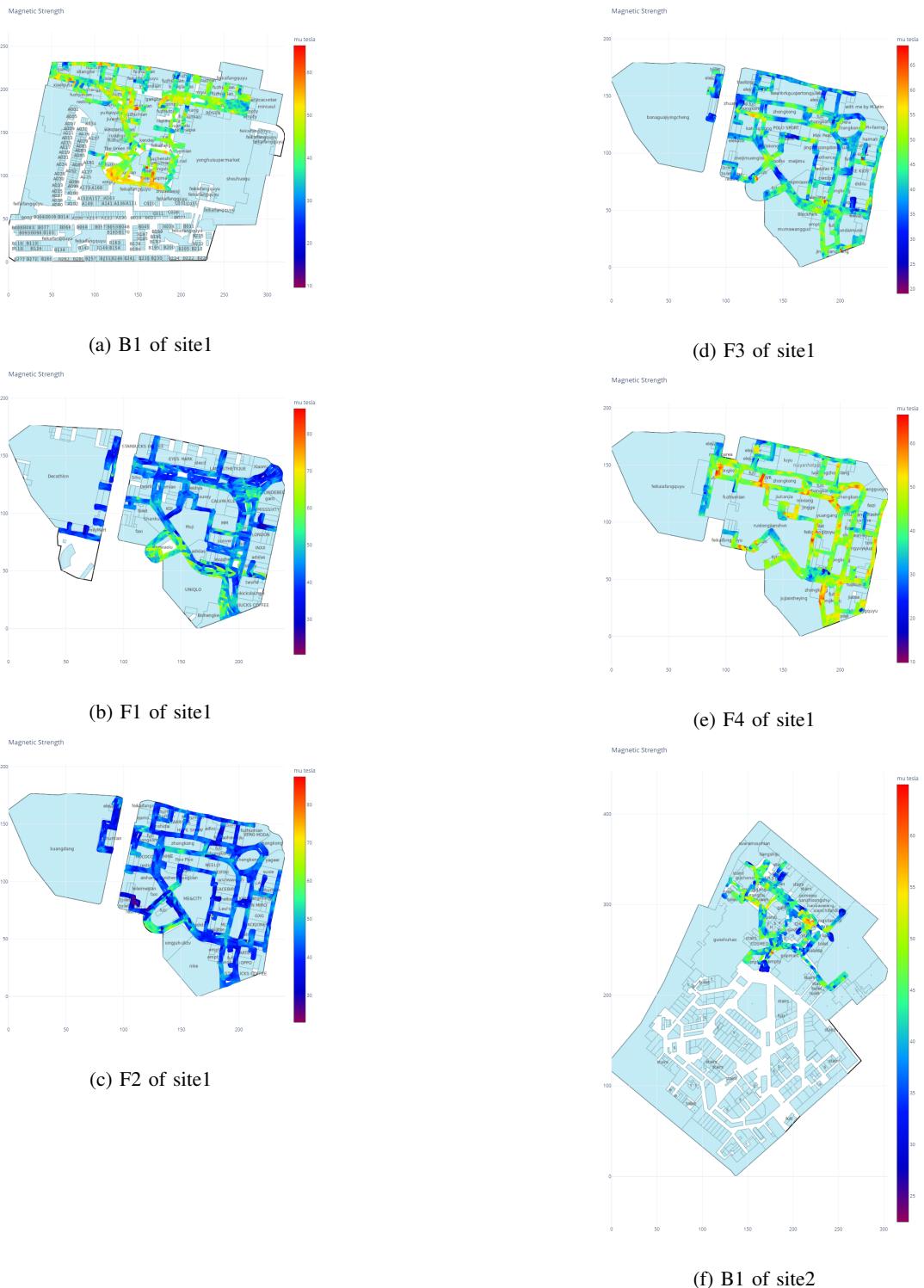
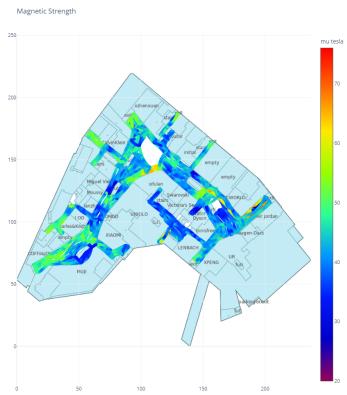


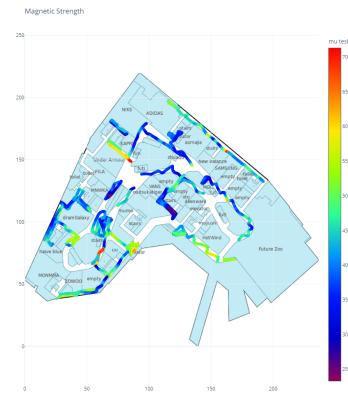
Fig. 8: Geomagnetic heat maps for all floors in site1 and B1 of site2

VII. REFERENCES

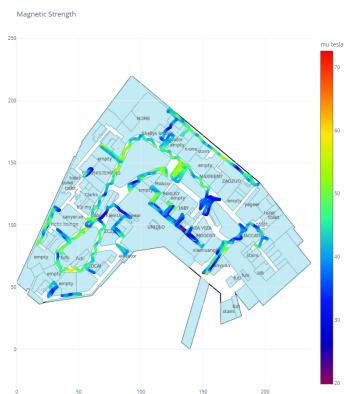
- X. Song, X. Fan, X. He, C. Xiang, Q. Ye, X. Huang, G. Fang, L. L. Chen, J. Qin, Z. Wang. (2019). CNNLoc: Deep-Learning Based Indoor Localization with WiFi FingerPrinting. IEEE. 589 – 595. doi:10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCL2019.00139.
- X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, G. Fang. (2019). A Novel Convolutional Neural Network Based Indoor Localization Framework With WiFi Fingerprinting. IEEE Access, 7, 110698-110709, doi:10.1109/ACCESS.2019.2933921.
- M. Research, “Indoor location competition 2.0 (sample data and code),” (2020). Available: <https://github.com/locationcompetition/indoor-location-competition-20>
- A. Abadleh, E. Al-Hawari, E. Alkafaween, and H. Al-Sawalqah, “Step detection algorithm for accurate distance estimation using dynamic step length,” in 2017 18th IEEE International Conference on Mobile Data Management (MDM). IEEE, 2017, pp. 324–327.



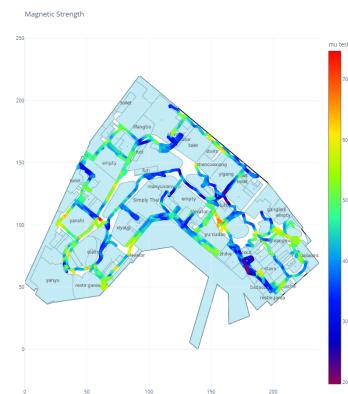
(a) F1 of site2



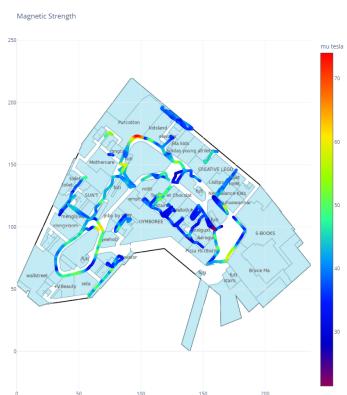
(e) F5 of site2



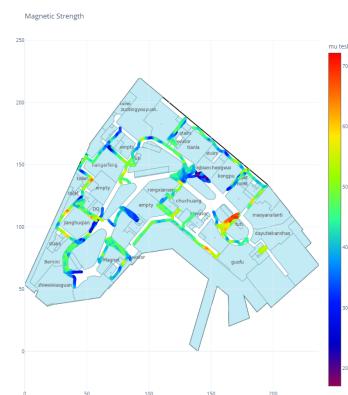
(b) F2 of site2



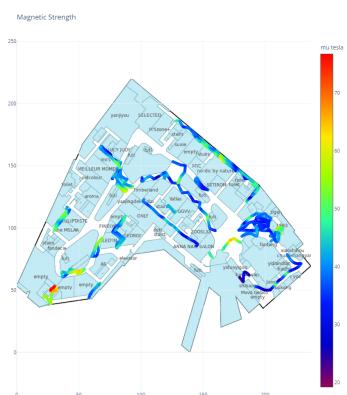
(f) F6 of site2



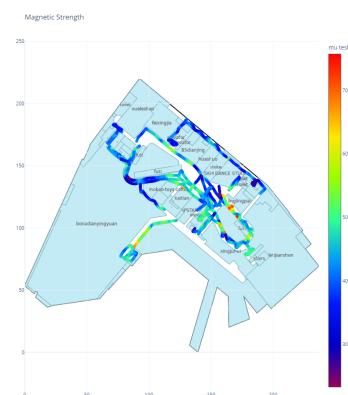
(c) F3 of site2



(g) F7 of site2



(d) F4 of site2



(h) F8 of site2

Fig. 9: Geomagnetic heat maps all floors except B1 in site2

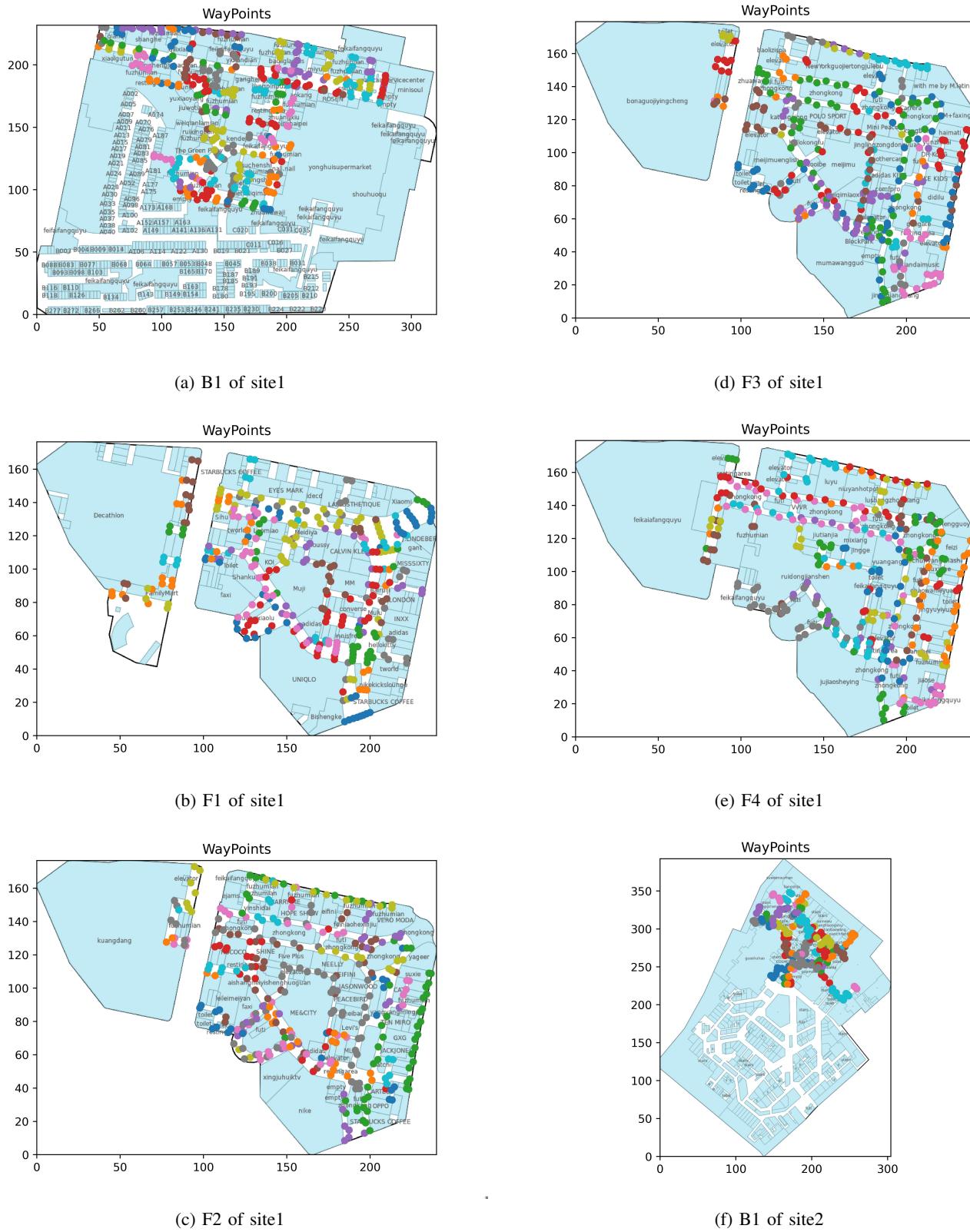


Fig. 10: Waypoints maps for all floors in site1 and B1 of site2

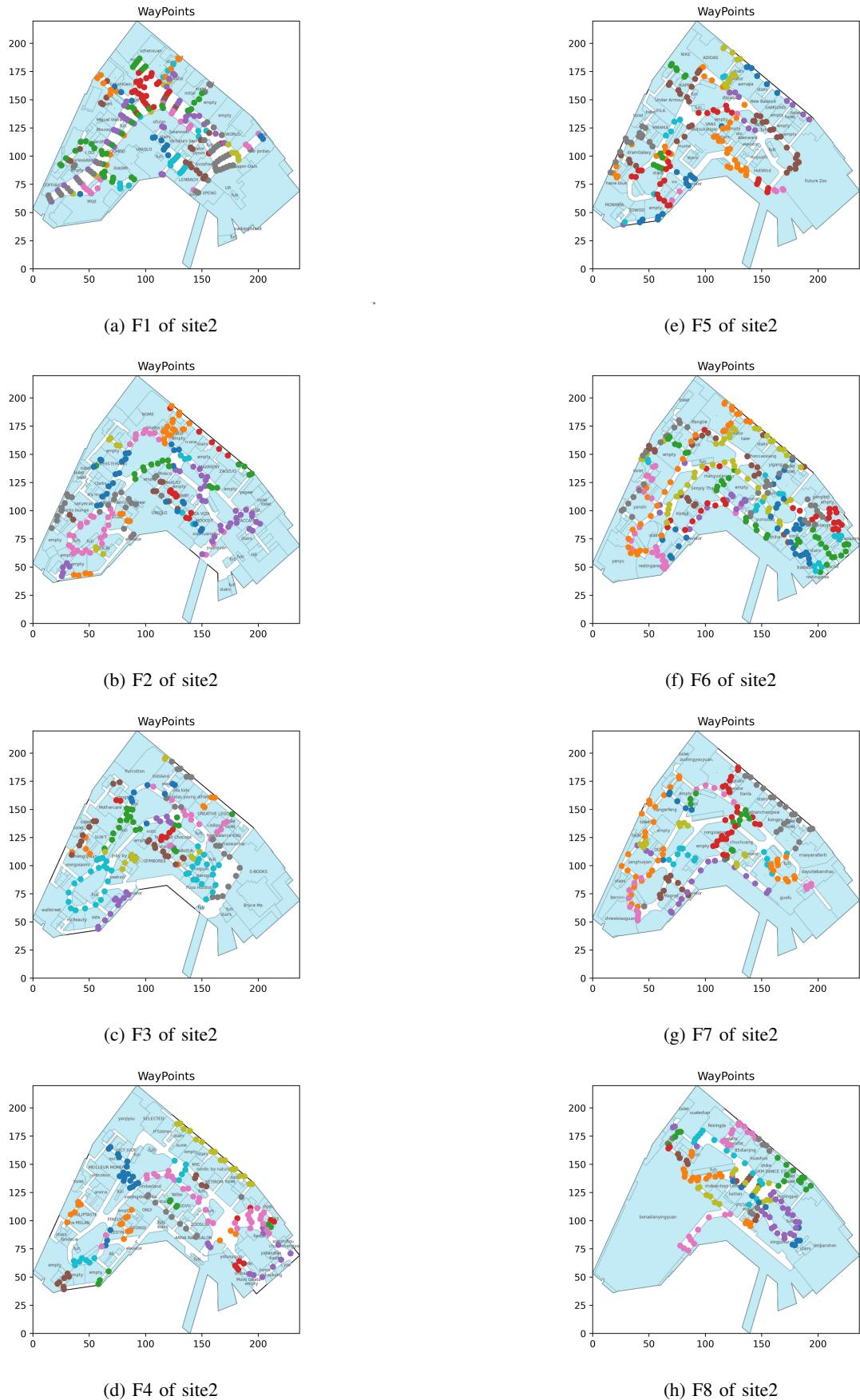


Fig. 11: Waypoint maps all floors except B1 in site2

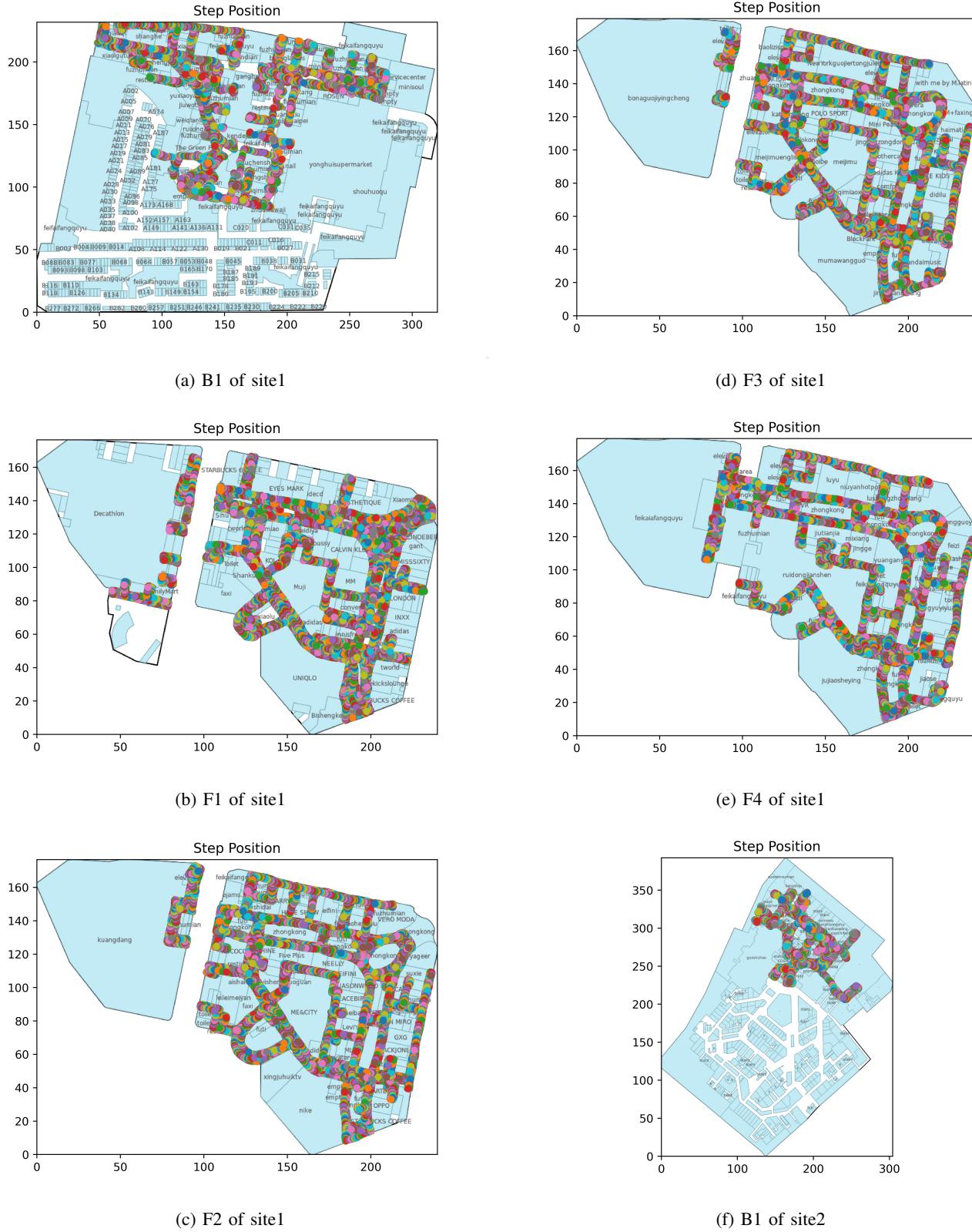
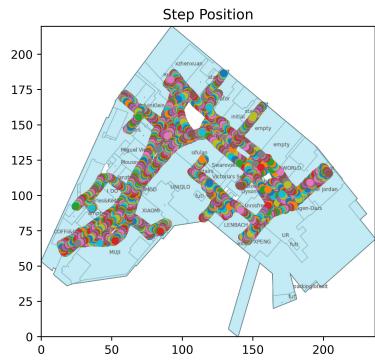
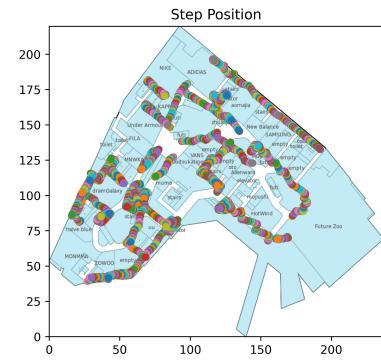


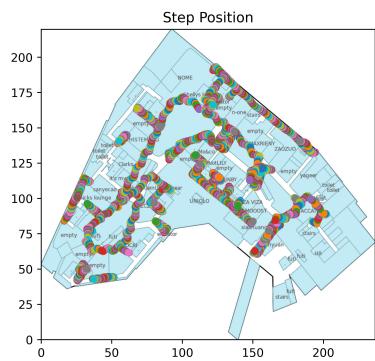
Fig. 12: Step position maps for all floors in site1 and B1 of site2



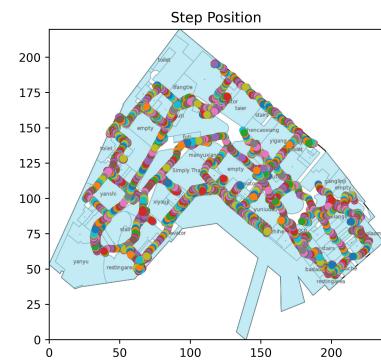
(a) F1 of site2



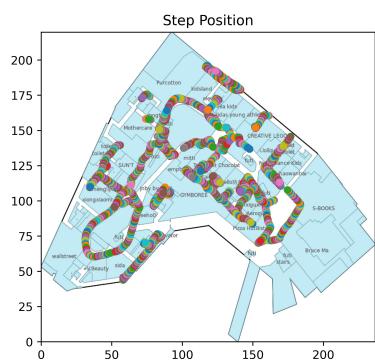
(e) F5 of site2



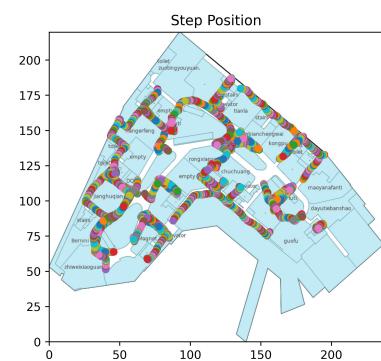
(b) F2 of site2



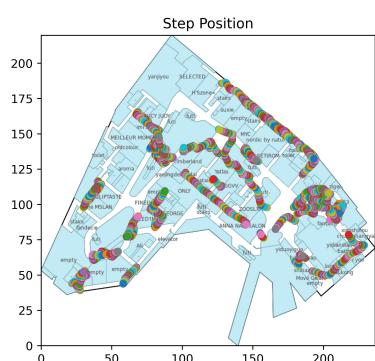
(f) F6 of site2



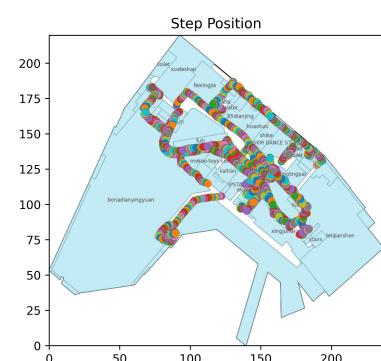
(c) F3 of site2



(g) F7 of site2



(d) F4 of site2



(h) F8 of site2

Fig. 13: Step position maps all floors except B1 in site2

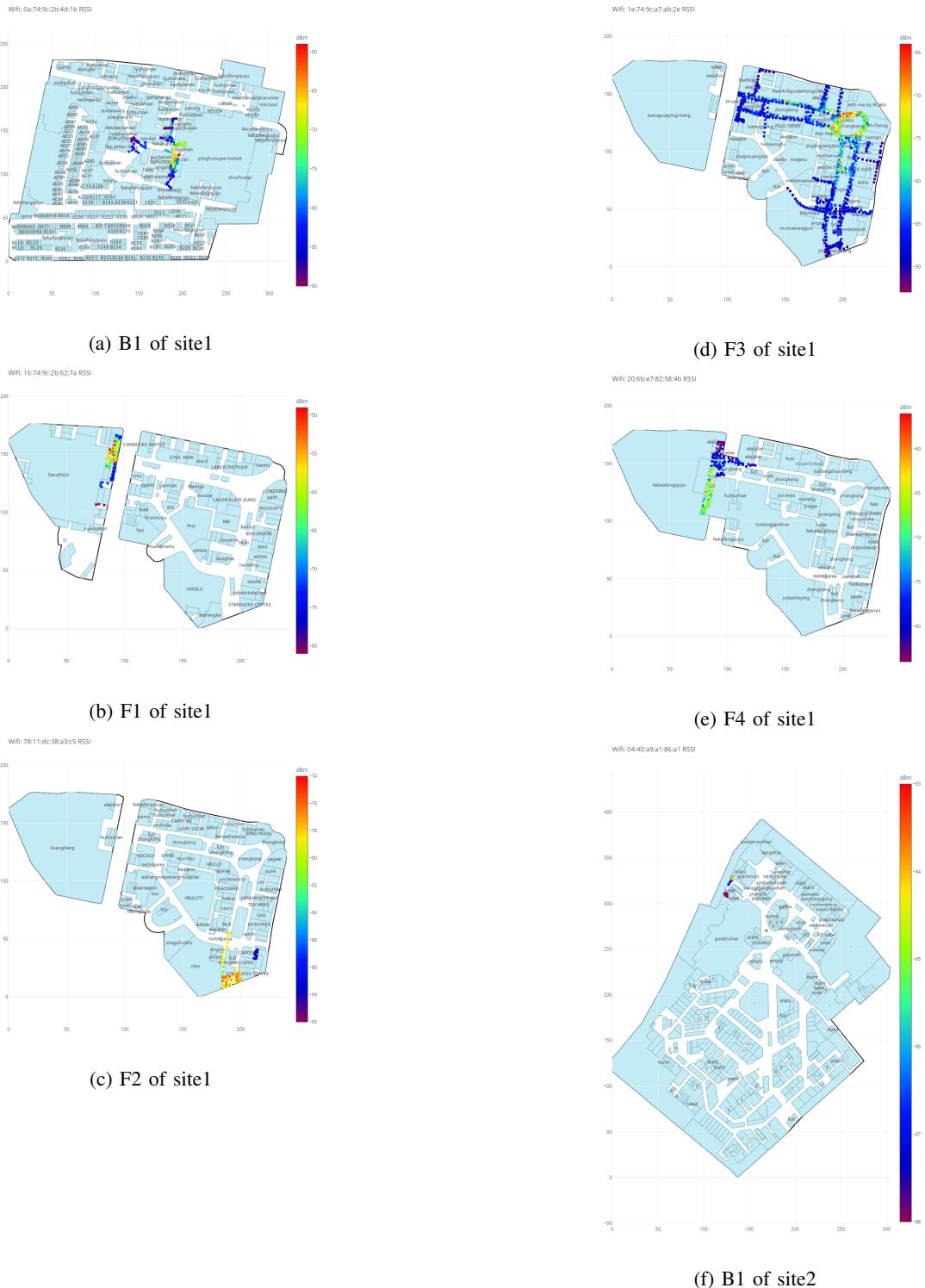
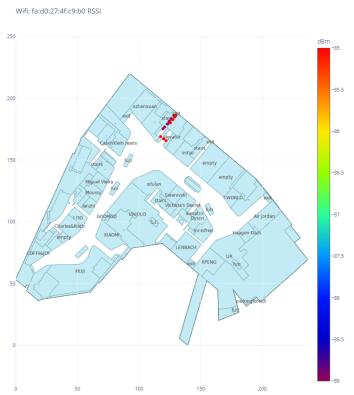
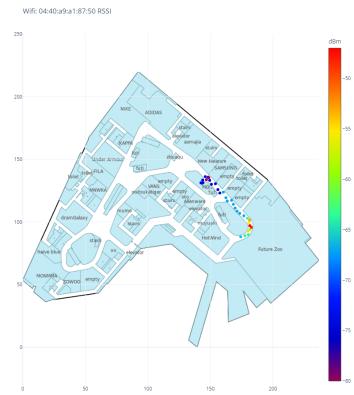


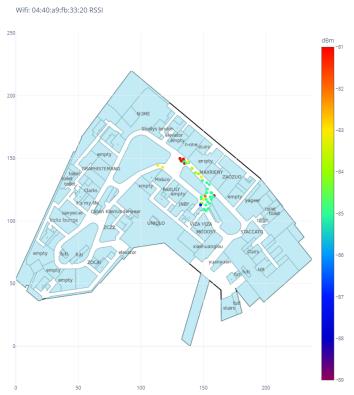
Fig. 14: WiFi RSSI for random WiFi APs from all floors of site1 and B1 of site2



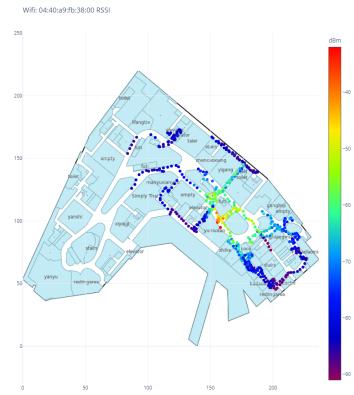
(a) F1 of site2



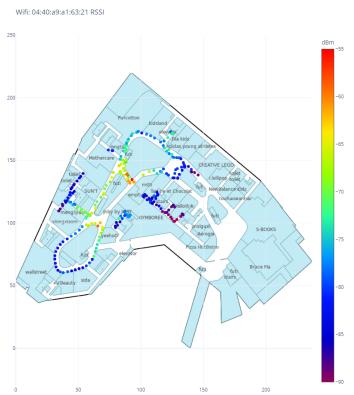
(e) F5 of site2



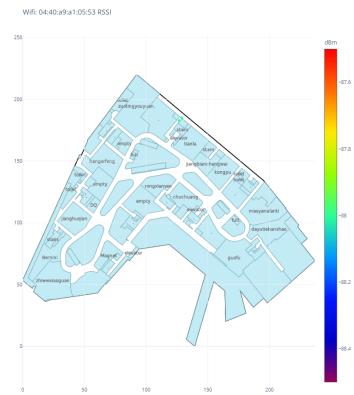
(b) F2 of site2



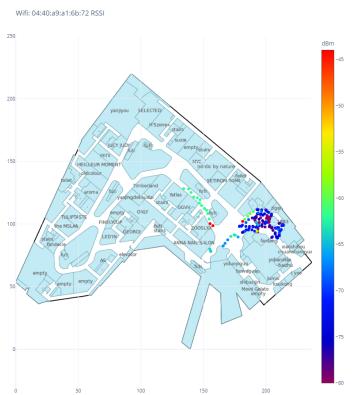
(f) F6 of site2



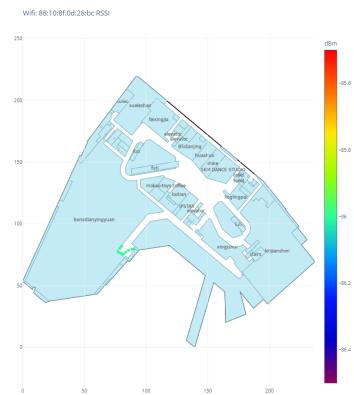
(c) F3 of site2



(g) F7 of site2



(d) F4 of site2



(h) F8 of site2

Fig. 15: WiFi RSSI for random WiFi APs from all floors, except for B1, of site2