Group Project: RSSI Localization via ESP32

Chan Yin Kei
u3578749@connect.hku.hk
IoT Wizards
Hong Kong

Chiu Hoi Kit Marco
mc0123@connect.hku.hk
IoT_Wizards
Hong Kong

Jawwad Muhammad Ghassan
ghassan@connect.hku.hk
IoT_Wizards
Hong Kong

Nip Hok Leung
niphokleung@gmail.com
IoT_Wizards
Hong Kong

1 SYSTEM OVERVIEW

The system has components as follows:

esp32publish: Uploads a simple program to the ESP32. The program allows the ESP32 to connect wirelessly with the host machine (my laptop), and will continuously publish information on detected WiFi networks at regular intervals to the "RSSI_Measurement" topic. The published information includes the WiFi SSID, RSSI, and Mac address. ScanInterval can be adjusted, and scans can be set to target only a WiFi network with a specific name.

mqtt_subscribe: A python client that subscribes to the "RSSI_Measurement" topic on the broker to receive the measurements, and saves the results in a file "payload.csv".

engine: Contains all the code for the algorithm used to predict the client location. Is also responsible for interfacing with the user and taking inputs for AP locations and calibration labels, and reading values from the "payload.csv" file output by mqqt_subscribe.py

The usage is as follows:

Preparation: Get a list of APs and write down coordinates for their location. Make sure to know their mac addresses so they can be identified.

1. Run **mqtt_subscribe** to subscribe to the channel to listen for RSSI measurements.
2. Run **esp32publish** to get RSSI measurements from surrounding APs.
3. Wait for the first subscription messages to arrive. This will generate the file "payload.csv".
4. Run **engine** and look through the SSID/mac addresses in payload.csv and identify the known AP locations and input their coordinates.
5. On **engine** calibrate the location by providing the location of the client at some timestamps. Recommended not to type 'l', and to type 's' to use the Hata-Okumara model with shared parameters for each AP.
6. Choose timestamps to predict the client location on.

2 ALGORITHM DESIGN

Initial analysis on the provided data shows that, even without using the provided locations of the APs, a random forest model trained on the data is able to provide accurate prediction of the true location of the clients.

The strong performance of the random forest model suggests that signal strength from the 6 AP locations is sufficient to provide an accurate prediction, however, as we will not be given labelled data and not be able to train our model in advance on thousands of data points for a new environment, we will need to engineer features, likely using information on the provided locations of the APs.

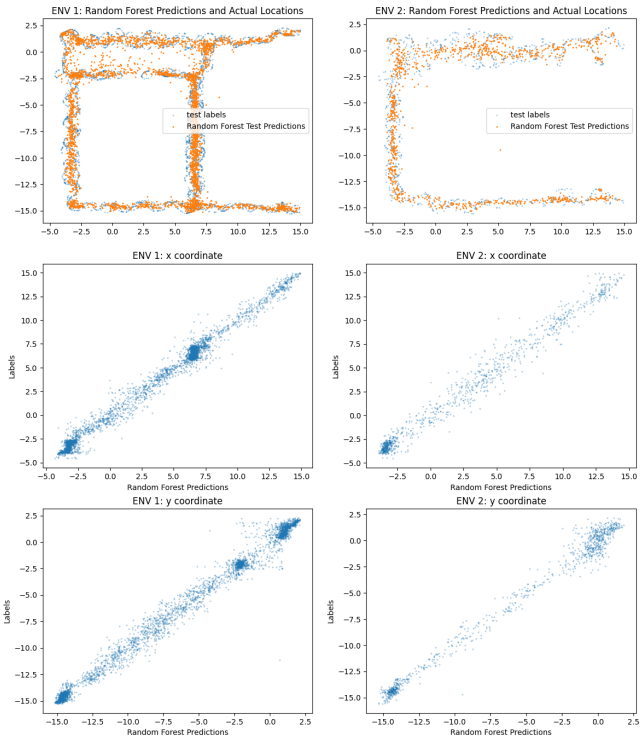

Figure 1: Results of random forest model

Plotting RSSI signal strength against distance from the GT label, it appears that there is a noisy linear relationship between them which does not vary hugely between APs.

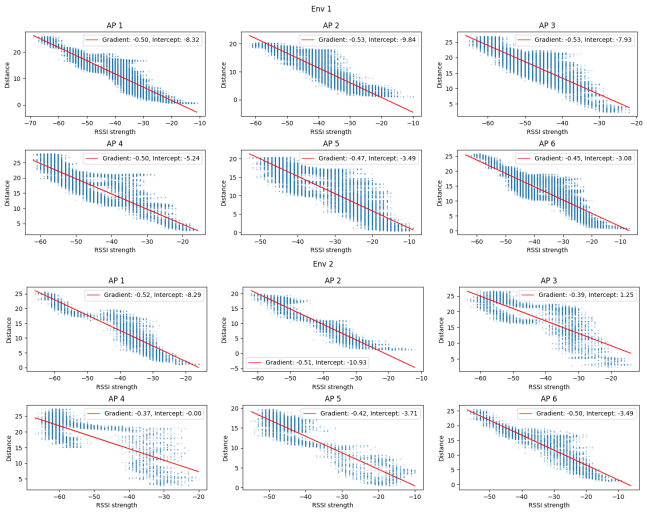

Figure 2: Linear relationship of RSSI and distance for each AP

If we attempt to look for a 'one-size-fits-all' relationship for all points in the dataset, a linear relationship still seems somewhat reasonable.

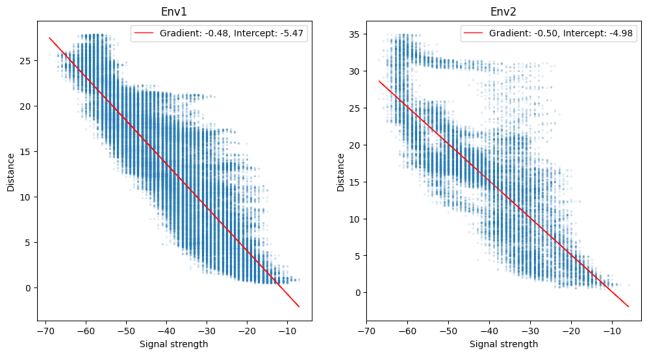

Figure 3: All AP datapoints aggregated

An alternative is to estimate the distance using the *Hata-Okumara* model[2], where the distance is described by:

$$d = \exp \frac{1}{10n}(M - P_{RX} - X_\alpha + 20 \log \lambda - 20 \log 4\pi)$$

Here, $M = G_{TX} - G_{RX} + P_{TX}$, where G_{TX}, G_{RX} are the attenae gain of the transmitter and receiver respectively, in dBi, and P_{TX} is the transmitted power level in dBm.

P_{RX} is the power level measured at the receiver, X_α is a normal random variable with a standard deviation of α, λ is the wavelength of the signal in meters (assumed to be 0.12m), and n is a coefficient that measures the influence of obstacles on the signal. For free space, $n \approx 2$, and for obstructed environments, $n \approx 5$ are reasonable initial estimates. [1]

This model suggests that the logarithm of the distance is a linear function of signal strength. Plotting this shows a slightly worse fit for the given data than a linear model, but this may be an artifact of the specific datasets provided.

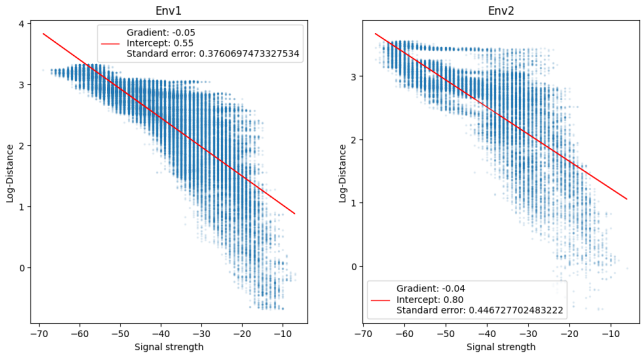

Figure 4: Loglinear relationship

To apply the model requires finding values of M, n and α. Based on the standard errors from the regression line of the loglinear relation shown above, we estimated $\alpha/n = 0.45$.

We created two models based on different starting assumptions assumptions. The first assumes that values of M, n are the same for all access points, and the second assumes that each access point can have distinct values of M, n. Both models are calibrated by being given samples of client locations and corresponding RSSI values of the AP locations. The pair of M, n values corresponding to the maximum likelihood (fixing α/n to be 0.45 as stated earlier) are then found. The optimization was done via the library Optuna, which uses a Bayesian framework to search the parameter space.

Given values M, n, α and P_{RX}, we can compute the log-likelihood of the client being at a particular distance from any AP using the Hata-Okumara model. By summing the log-likelihoods over all APs, we get the log-likelihood corresponding to a client location at a pair of x,y coordinates. We tested two methods to use the likelihood estimates to return the location of the client: The first once again uses maximum likelihood estimation, and searches the area for the coordinates with the greatest likelihood, returning the corresponding pair of coordinates. The second method partitions the space into a regularly spaced grid, evaluates the likelihood at the coordinates of the grid and uses them as weights, and returns the centre of mass of the weighted coordinates. Initial testing revealed better performance for the second method, so we decided to move forward with it for further analysis and evaluation.

3 EVALUATION

3.1 Hata-Okumara model

We perform several experiments to test our model using the algorithm described in part 2, to get a general sense of the accuracy of the algorithm, and how many data points are necessary for calibration.

We test the algorithm as follows:

1. For a calibration size c, we calibrate M, n on a random sample from the data of size c.

2. Using the calibration parameters, we estimate the coordinates of 5 other points, and record the distance at which we are off by.

3. We repeat the experiment 20 times and record the errors. (Ideally, we would repeat the experiment many more times, but each run takes around half a minute.)

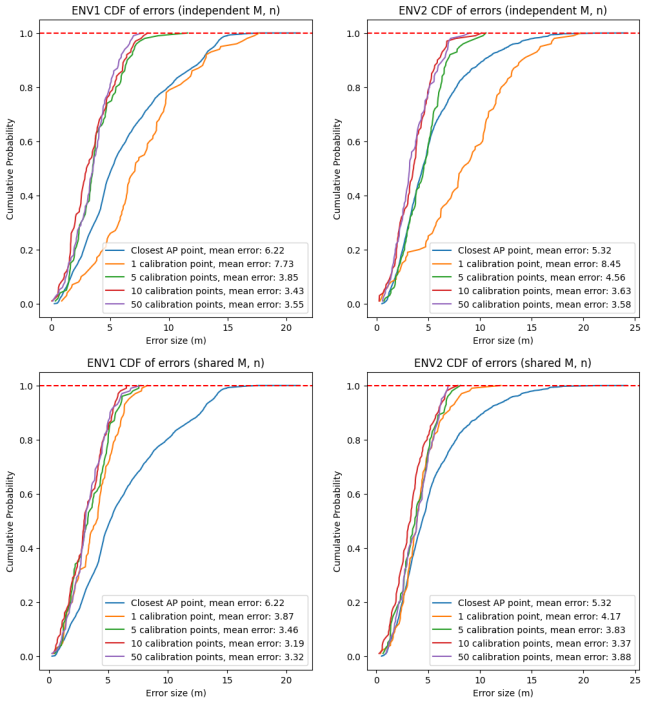

Figure 5: CDF of errors

As a baseline for performance, we compared our errors to that of simply taking the coordinates of the AP point with the smallest RSSI as a best guess. This yielded a mean error of 5.5m and 5.3m in ENV1 and ENV2 respectively.

From the results, we can see that using independent M, n values performed poorly when the size of the calibration set was small, but was similar to that of using shared M, n values when the size of the calibration set was larger. When using shared M, n values, a very small sample is required for calibration.

Except in the case of using 1 sample only with independent M, n values, our algorithm showed substantial improvement over the baseline, but was not able to come remotely close to the performance of random-forest fingerprinting.

In order to further improve performance, we would need to learn and use more information about the environment the calibration stage. Our model relies on the assumption of fixed values of M, n, α, which may not hold. Further potential areas for improvement of the model are:

1. Modelling α independently for each AP: Doing this will allow more reliable APs (smaller αs) to be weighted more heavily in likelihood estimates.

2. Revising our model for n to vary based on client location. This is challenging and likely only possible if calibration is done comprehensively over the space. A possible way to do this is to model n as a Gaussian process.

Parameters could also be adjusted for speed/accuracy tradeoffs. Examples would be the density of points on the grid (for finding the weighted mean), and number of trials (guesses) done during parameter optimization for M, n.

3.2 Linear model

I was curious to see the potential of a linear model, as it appeared to fit the data better. I implemented a linear model based on the assumption that for each AP

$$d = mP_{RX} + c + X_\alpha$$

where m, c are coefficients for the slope and intercept of a linear regression, and X_α is a normally distributed random variable of standard deviation α. Parameters were fit independently for each AP, and α was estimated using the mean leave-one-out absolute error divided by $\sqrt{\frac{2}{\pi}}$ [?]. The coordinates returned are the maximum likelihood estimate.

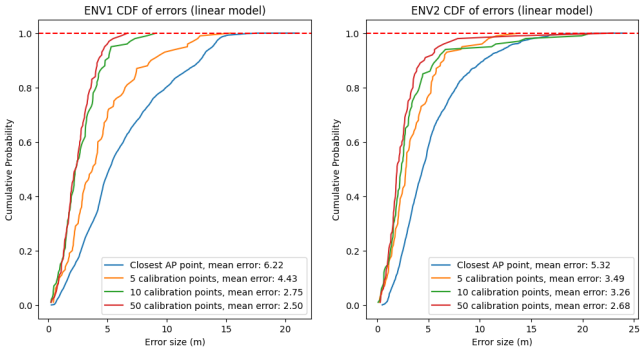

Figure 6: CDF of errors (linear model)

The linear model outperformed the Hata-Okumara model on the dataset, provided there were enough calibration points, which is unsurprising as it did fit the data better.

4 REAL-WORLD EXPERIMENTS

4.1 War-Driving

The goal of our war-driving exercise was to identify and map the locations of five different APs broadcasting the same SSID, "HKU" on Level-1 (CPG-1) of the Chi Wah Learning Commons. To determine the AP locations, we targeted MAC addresses exhibiting an RSSI greater than -40 dBm. As higher RSSI values indicate stronger signal strength, suggesting closer proximity to the respective AP.

```
Scan Done
SSID: HKU, MAC: 30:C5:0F:68:8E:20, RSSI: -30
SSID: Wi-Fi.HK via HKU, MAC: 30:C5:0F:68:8E:23, RSSI: -30
SSID: Y5ZONE, MAC: 30:C5:0F:68:8E:26, RSSI: -30
SSID: eduroam, MAC: 30:C5:0F:68:8E:22, RSSI: -30
SSID: CSL Wi-Fi Roam, MAC: 30:C5:0F:68:8E:24, RSSI: -30
SSID: CSL, MAC: 30:C5:0F:68:8E:25, RSSI: -30
SSID: eduroam, MAC: 30:C5:0F:68:9E:A2, RSSI: -54
SSID: Y5ZONE, MAC: 30:C5:0F:68:9E:A6, RSSI: -54
SSID: HKU, MAC: 30:C5:0F:68:9E:A0, RSSI: -55
SSID: Wi-Fi.HK via HKU, MAC: 30:C5:0F:68:9E:A3, RSSI: -55
SSID: CSL Wi-Fi Roam, MAC: 30:C5:0F:68:9E:A4, RSSI: -55
SSID: IIOPGH, MAC: 82:50:87:F4:51:CB, RSSI: -58
SSID: CSL, MAC: 30:C5:0F:68:8E:E5, RSSI: -60
SSID: Wi-Fi.HK via HKU, MAC: 30:C5:0F:68:9E:63, RSSI: -60
SSID: eduroam, MAC: 30:C5:0F:68:9E:62, RSSI: -61
SSID: Y5ZONE, MAC: 30:C5:0F:68:9E:66, RSSI: -61
SSID: HKU, MAC: 30:C5:0F:68:8E:E0, RSSI: -62
SSID: eduroam, MAC: 30:C5:0F:68:8E:E2, RSSI: -62
```

Figure 7: Example of an RSSI trace

The selection rationale of specific AP locations was informed by marked disparities in RSSI values among various MAC addresses, as detected through comprehensive Wi-Fi tracing activities. For instance, as depicted in Figure 7, there is a notable RSSI difference of at least 25 dBm between the primary AP and the other APs broadcasting SSID "HKU". This substantial variance in RSSI values is indicative of distinct physical separations between the APs. These observations led us to pinpoint the following locations in Figure 8.

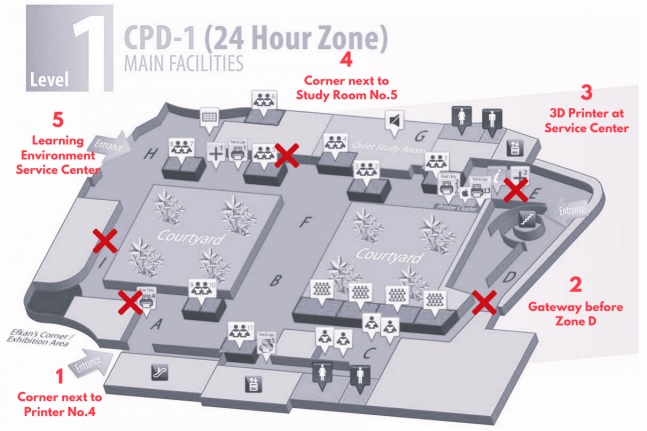

Figure 8: Floor plan of Chi Wah with locations marked

The detailed AP Locations are as follows:

1. **MAC: 30:C5:0F:68:89:60, SSID: HKU** located at a secluded corner before Zone A, an area not depicted in the floor plan. This specific location is characterized by its proximity to Printer No.4 and a dividing wall.

2. **MAC: 30:C5:0F:68:95:60, SSID: HKU** situated at the gateway just prior to entering Zone D from Zone C.

3. **MAC: 30:C5:0F:68:8D:60, SSID: HKU** positioned adjacent to the 3D printer located at the service counter at Zone E. The spot is distant from the stairs and closer to the lifts.

4. **MAC: 30:C5:0F:68:8E:E0, SSID: HKU** found outside Study Room No.5 in Zone F, near a corner that is separated by a wall which includes a door.

5. **MAC: 30:C5:0F:68:82:20, SSID: HKU** positioned outside, directly in the middle of the Learning Environment Service Centre in Zone I, surrounded by tables and chairs which make it a study area for students.

For all these locations, we have recorded an RSSI value of around -32 on average for several tracings. This maintains a clear disparity in RSSI values compared to other areas, suggesting less interference and stronger connectivity. Therefore, it is believed that we have successfully identified five distinct AP locations.

5 CONTRIBUTION STATEMENT

Below are the specific roles assigned during the project:

1. Nip Hok Leung is responsible for setting up the subscribe/publish infrastructure and engine filesystem, preliminary data analysis, design, implementation and evaluation of the client localization algorithm.

2. Chan Yin Kei is taking part in the initialization phase and is responsible for conducting War-Driving activities.

3. Chiu Hoi Kit Marco is taking part in the initialization phase and is responsible for conducting War-Driving activities.

4. Jawwad Muhammad Ghassan discussed the client localization algorithm, tested the system and proofread the report.

REFERENCES

[1] Atreyi Bose and Chuan Heng Foh. 2007. A practical path loss model for indoor WiFi positioning enhancement. In *2007 6th International Conference on Information, Communications Signal Processing*. 1–5. https://doi.org/10.1109/ICICS.2007.4449717

[2] M. Hata. 1980. Empirical formula for propagation loss in land mobile radio services. *IEEE Transactions on Vehicular Technology* 29, 3 (1980), 317–325. https://doi.org/10.1109/T-VT.1980.23859

[3]]1850703 user46234. [n. d.]. Mean Absolute Deviation of normal distribution. Mathematics Stack Exchange. arXiv:https://math.stackexchange.com/q/1850703 https://math.stackexchange.com/q/1850703 URL:https://math.stackexchange.com/q/1850703 (version: 2016-07-06).