Smart Phone Sensing - Report 2

Group: 7, SmartPhone: OnePlus 6T, Android Ver: 10, Git Branch: main

Minas Melas
NetID: mmelas
Student No.: 5397324
Email: M.Melas@student.tudelft.nl

Nishad Mandlik
NetID: nmandlik
Student No.: 5242541
Email: N.G.Mandlik@student.tudelft.nl

***Abstract*—One of the techniques that have been developed for localization is the analysis of different RSSI signals gathered from nearby WiFi Access Points using Bayes' theorem. There are multiple methods of using Bayes' theorem such as parallel or iterative and many different variables and filtering methods. We used a composite of the serial and parallel approaches for calculating the position of the user. We used some filtering techniques such as picking only WiFi from well known sources and considering only the nearest access points. Finally, we computed the confusion matrices and the accuracies of different tests.**

Division of Labour:
Minas: Code for Data Collection, Cross Validation and RSSI Interpolation.
Nishad: UI Design and Code for Inference and Data Filtering.

I. Data Collection

During our initial study, it was observed that for some APs, on moving between neighbouring cells, there was negligible change in the RSSI values. Upon studying the strength of the mobile network, we noticed that the values decreased considerably upon moving away from the window, towards the interior of the room. Furthermore, upon moving around the floor, we found that the phone connects to different base stations at different locations in the building. Thus, we decided to use the network strengths of the mobile base stations to aid the WiFi RSSI data. Using a dual-sim phone, we are able to collect data for two mobile operators.

Every time the user presses any button that will start a scanning process, the app introduces a delay of 5 seconds, and performs some scans whose results are not utilised. This is done to tackle the limitation put by some smartphones on the frequency of updating the scan results. After the scan, we disable scanning for 15 seconds so that old results do not occur after moving to the next cell.

The data is collected by performing 100 iterations at each cell. In each iteration, a set of 3 WiFi scans and a set of 5 mobile network scans are performed (explained in the next section). The delay between successive WiFi scans is 200ms. In case of mobile networks, the signal strength is updated internally by the OS, which means that the app cannot initiate a mobile network scan. Since these updates are not performed at a very high frequency, the delay between collection of consecutive samples is set to 500ms, instead of 200ms as in the case of WiFi. Thus, the total time needed per cell for the training process corresponds to roughly 6 minutes. In addition to the training data, we collected some extra samples, using the 'SAMPLES' button in the app. These were used for cross-validation, which is explained in the next section. When gathering the training scans that resulted in the confusion matrix II, we remained static in the center of the cells and did not rotate at all while we had the phone in our hands.

II. Data Processing

As mentioned earlier, collection of WiFi data is performed in sets of 3 scans. In each set, the average RSSI for each AP is computed, so as to reduce the variation in the RSSI values. In order to ensure that the training data does not expire over time, APs with randomized MAC addresses are removed from the collected data. These MAC addresses have their second most significant nibble equal to 2, 6, A or E [1]. Apart from these, APs created by mobile hotspots may also disappear over time. Secondly, smart devices like Google Assistant, network cameras, home-automation devices also operate as an AP during their configuration phase. Once configured, they switch to Station mode and cease to exist as an Access Point. Most of such APs are filtered out by using MAC address vendor-lookup. Only the APs from popular router manufacturers such as *Netgear*, *TP-Link*, *D-Link*, *Belkin*, etc. are retained. Furthermore, Access Points which have a low beacon frequency or those whose signal strength is low in a particular cell, may lead to inconsistencies during testing. Thus, we consider only those Access Points which have appeared in more than 10% of the total scans performed in that cell.

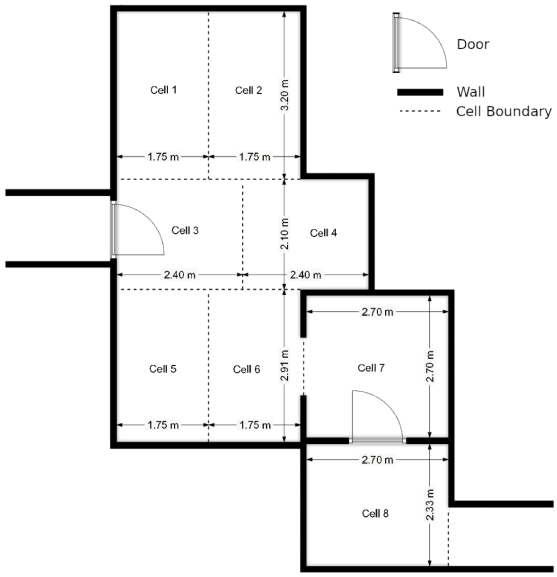

Fig. 1: Home layout

If only the APs with strongest signal strengths are selected during training, and one (or more) of them are not available during testing, the model is left with lesser data-points for arriving at an inference. Thus, in order to make the training data as complete as possible, we perform RSSI-based filtering during the testing phase instead. After scanning the Access Points during testing, 5 APs which have the least RSSI magnitudes, and which are present in the training data, are selected for the Bayes' calculation.

It is possible, that for a particular AP, the RSSI value which has been detected during testing, has never occurred during training, but values in its neighbourhood might have occurred. To tackle this issue, we calculate the probabilities as follows:

$$p(RSSI) = \frac{\sum_{i=RSSI-m}^{RSSI+m} p(i)}{2m+1}$$

$$p(RSSI|cell) = \frac{\sum_{i=RSSI-m}^{RSSI+m} p(i|cell)}{2m+1}$$

where m is a 'slack' given to the detected RSSI value. The additional samples which we had gathered during data collection were used for computing the best value for k, using cross-validation. This resulted in $k = 3$ for maximum accuracy.

We observed that the signal strength of mobile network shows greater fluctuations as compared to WiFi RSSI values. Thus, we selected 5 samples for averaging, as opposed to the 3 in the case of WiFi. We also observed that multiple 'network cells', were detected by Android, however, the data is consistent, only for the cells which are currently serving the phone. Thus, among the detected network cells, only two serving cells are selected (one for each SIM card). With this, the volume of data gathered is quite low, and no further filtering is performed. The computation of probabilities is similar to that used for WiFi data.

III. Sample Radio Maps

Figure 2 shows the similar (for Cells 4 and 6), and different (for Cells 7 and 6) PMF distributions for two Access Points. Cells 4 and 6 are situated in an open area and are in line of sight. Thus, the loss/gain in signal strength is lesser while moving between these cells. If the AP is situated very close to one of these cells, considerable decrease in signal strength may be observed while moving to the other. However, in this case, from the low RSSI values, it is evident that the AP is not situated close to the cells. Thus, it can be said that the distance between the cells is insignificant to cause any difference in the network strength. In the case of the different PMF distributions, the AP is most likely closer to Cell 7 (due to the stronger signal). Since Cell 7 is enclosed by walls on all sides, the radio waves have to penetrate a dense medium while travelling from Cell 7 to Cell 6, which explains the difference in signal strengths for these two cells.

For mobile network strengths, there are windows at the top of Cell 1 and Cell 2, and at the bottom of Cell 5 and Cell 6 , in the sense of Figure 1. Hence, Cells 1 and 2 receive a similar signal strength and have similar PMFs (Figure 3. In the case of Cell 4 and Cell 6, Cell 4 is more towards the inside, while Cell 6 is closer to the window. This results in a lower signal strength for Cell 4, making is easy to distinguish from Cell 6. It is interesting to note that although the WiFi PMF for Cell 4 and Cell 6 was similar, the mobile network PMF is different. Hence, the mobile network data aids the WiFi data while determining the position.

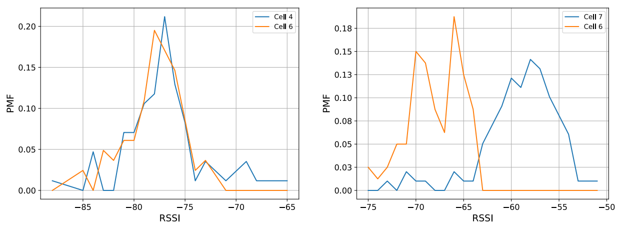

Fig. 2: Similar (left) and different (right) PMF distributions of WiFi RSSI for two cells.

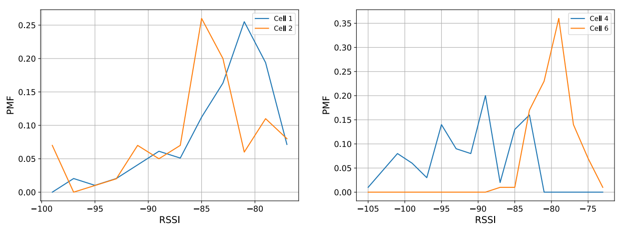

Fig. 3: Similar (left) and different (right) PMF distributions of Mobile Network RSSI for two cells.

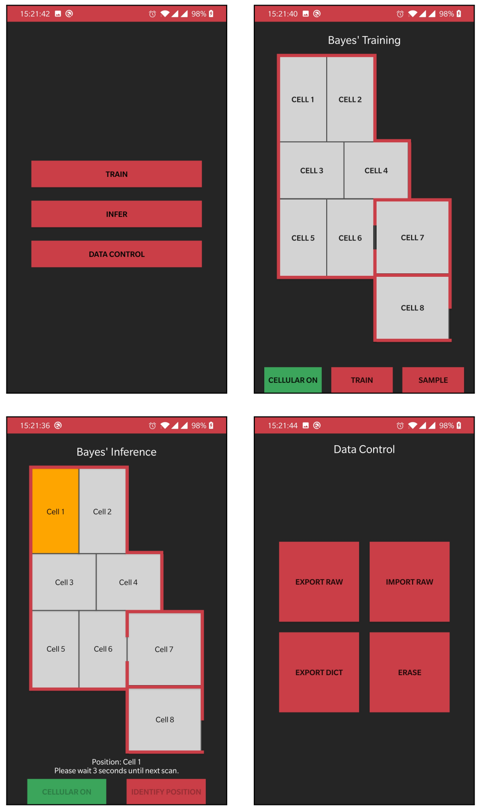

Fig. 4: App home screen (top left), and screens for Training (top right), Inference (bottom left), and for Deleting or Exporting/Importing Trained Data to/from external storage. (bottom right)

TABLE I: Confusion Matrix for inference using only WiFi data.

	Cell1	Cell2	Cell3	Cell4	Cell5	Cell6	Cell7	Cell8
Cell1	6	2	0	0	0	2	0	0
Cell2	1	8	0	0	1	0	0	0
Cell3	2	0	3	1	0	2	1	1
Cell4	0	1	0	2	0	6	1	0
Cell5	0	0	0	0	6	4	0	0
Cell6	0	0	0	0	0	10	0	0
Cell7	0	0	0	0	0	0	9	1
Cell8	0	0	0	0	0	0	4	6

TABLE II: Confusion Matrix for inference using both WiFi and Mobile Network data.

	Cell1	Cell2	Cell3	Cell4	Cell5	Cell6	Cell7	Cell8
Cell1	8	0	1	1	0	0	0	0
Cell2	0	10	0	0	0	0	0	0
Cell3	1	0	6	2	1	0	0	0
Cell4	2	1	3	4	0	0	0	0
Cell5	0	0	0	0	7	3	0	0
Cell6	0	0	0	0	0	10	0	0
Cell7	0	0	0	0	0	0	8	2
Cell8	0	0	0	0	0	0	0	10

IV. Evaluation

We used a combination of the serial and parallel Bayes' method for finding the most probable position when the '`IDENTIFY POSITION`' button is pressed. As stated previously, we use the five APs with the strongest signals that appeared in more than 10% of the training scans. For each of these Access Points, we compute the posteriors for each cell. Thus, in the first stage, we have 5 sets of posterior values. Each set among these, is used as the set of priors for each mobile network base-station. Hence, in the second stage, considering all combinations, we have 10 sets of posteriors. In each set a vote is cast for the cell which has the highest posterior value. The cell receiving the maximum number of votes is elected as the probable cell.

The screenshots of the app are shown in Figure 4

We can see from the confusion matrices that the accuracy for only using Wi-Fi signals is 62.5%, while including both WiFi and mobile networks is 78.75%.

V. Discussion

- (hard) Getting accurate results at various positions in the cell.
- (hard) Getting correct inference while facing a different direction than that during training. WiFi RSSI values do not vary much, but Mobile Network Strength was observed to reduce considerably upon facing away from the window.
- (novel) Inclusion signal strength of mobile network for computing the posteriors.
- (novel) Calculating the best "slack" (k) for averaging the occurrences of RSSI values in the $(-k, +k)$ neighbourhood.
- (novel) Filtering out APs with randomized MAC addresses, and those from unusual vendors.
- (novel) Considering only those access points that have appeared in at least 10% of the scans.
- (novel) Considering the five APs with the lowest RSSI magnitudes.

References

[1] Wes Purvis and Slava Dementyev. *Get to know MAC Address Randomization in 2020*. en-US. Sept. 2020. URL: https://www.mist.com/get-to-know-mac-address-randomization-in-2020/ (visited on 05/26/2021).