COMP9336 – Mobile Data Networking

Project – WIFI Fingerprinting

T2 2022

Name: Yuhua Zhao – **ZID:** z5404443

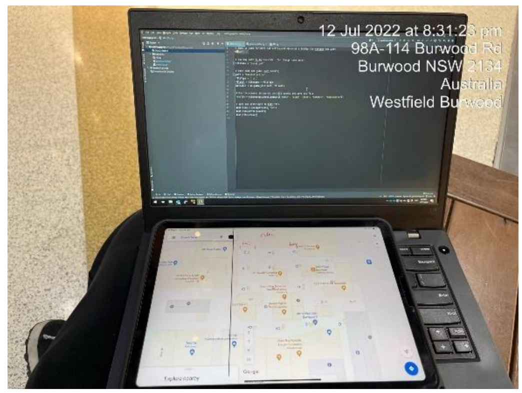
Introduction

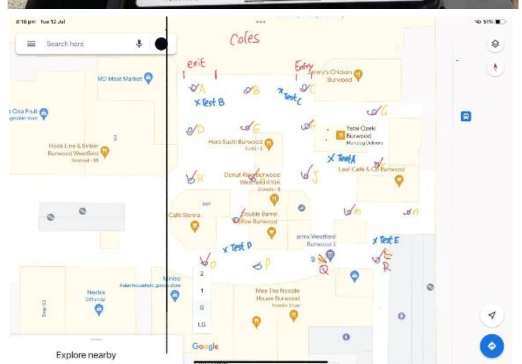
GPS is used for localization purposes, but the indoor area may not take effect due to limitations in radio frequency and travel distances. In the meantime, due to the number of WIFI devices deployed and mobile devices used, WIFI could be a possible solution for localization.

There are two possible solutions that I can think of to use wifi and an algorithm to get the user's positions. Firstly, we can take advantage of wireless access points (WAP), as WAPs are deployed in fixed locations. We can obtain where the Users are associated with which APs to get the appromixly location of Users, but this method is not suitable for this project. The second method use the WIFI Fingerprinting method that uses BSSID and RSS values to calculate the rough Distance between the Detection location and the database detection points. But the WIFI fingerprinting has a number of issues that will be discussed as follows. The fundamental calculation function of this project is using "N-dimensional Distance" which will be discussed in the Algorithm design section.

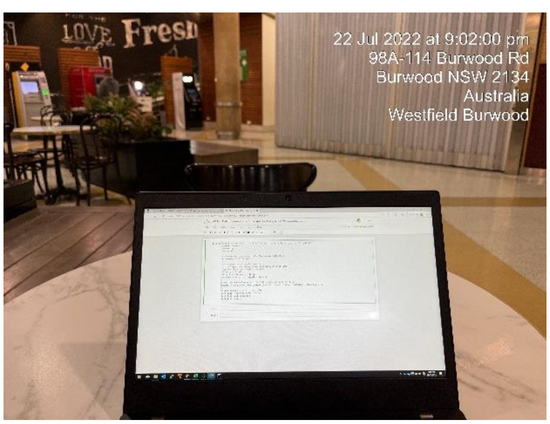
Experiments

My first RSS data collection was completed on 12/07/2022 at Burwood Coles Ground Floor. Each RSS data collection is done by using Window API and stored in the output to the File location. At this time, I collected 18 RSS outputs as detection points and 5 Test points for verification.





For this data collection, I simply walk to the data collection location and run the program to collect data without any wait time. But there I realized one issue even though I ran the data collection at different locations in a short period of time, some of the data sets' data are identical (I also did some test at other locations at different time, but I forgot to take photos). After that I designed "*Experiment 1*".



Experiment 1:

I ran the dataset collection at one location with different waiting times, this is because I realized that if I simply run the Window API to collect data without wait time, the old SSIDs are still stored in the Window OS. So, I ran the data collection at the same location same height but have a time gap between each collection.

Yuhua Zhao
Z5404443

Filename	Waited time (mins)	NO. SSID	NO. BSSID	NO. 2.4GHz	NO. 5GHz
22072022_0.txt	0	53	126	60	66
22072022_1.txt	1	36	61	41	20
22072022_2.txt	2	24	48	32	16
22072022_3.txt	3	24	46	31	15
22072022_4.txt	4	25	44	30	14
22072022_7.txt	7	26	44	31	13

As shown above, the number of SSID and BSSID significantly decreased at this location as the waiting time increased. If I ran the data collection right away when I get to the location, there will be more SSID compared to a longer wait time which potentially is because some SSID stored in Windows OS as caches that have not been clear.

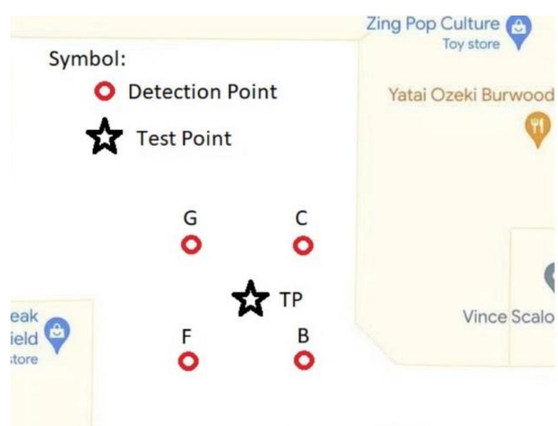
For each one-minute time gap, the SSID and BSSID significantly decrease until the waited time is 2. The result above shows that for each data collection at a different location, I should wait at least 2 minutes to ensure the dataset's accuracy.

After Experiment 1, I can see that retrieving stable RSS data is relatively difficult as users move around in an indoor area which may affect calculating an accurate location of users. Experiment 2 is an upgraded design that uses the "N-dimensional distance" formula to calculate the distance between Detection Point and the Test point. This N-dimensional distance uses SSID and BSSID to identify WAPs as one SSID may have multiple BSSIDs and one BSSID (MAC address of an AP) that may broadcast multiple SSID. But the combination of SSID and BSSID only has one. To get a different perspective of analysis, I also use different frequencies for the distance calculation in order to get a better result.

As we know, 2.4 GHz is relatively stable as the penetration to obstacles is good, the coverage area is large and many legacy WIFI routers support 2.4 GHz only. But the drawback is there are only three non-interference channels and massive radio frequency usage devices that may affect the channels performance as a result of inaccuracy of localization calculation such as Bluetooth and microwave, etc. 5 GHz frequency will not have the migrate disadvantage of 2.4 GHz as it has a maximum of 25 non-interference channels and fewer interferences. But the limitation of 5 GHz radio frequency is the obstacle penetration is weak.

Experiment 2:

At this time, I ran the Detection point data collection with more than 2 minutes of wait time to ensure accuracy. And the difference is the test point data collection. The expectation of WIFI fingerprinting is to allow users to move around and the program can be able to compare the collected data with the database at the background level to identify user locations. The “u” of the file name refers to unstable (without wait time) and the “s” refers to stable that wait more than two minutes before data collection.



Compare Test Dataset by Both Frequency

TestFile	LOC B	LOC C	LOC F	LOC G
Test_s1.txt	369.197	171.967	112.585	111.92
Test_s2.txt	406.604	321.86	276.074	173.153
Test_u1.txt	391.714	198.34	270.694	121.448
Test_u2.txt	401.437	194.697	166.46	182.465

==

Compare Test Dataset by using 2.4GHz Frequency

TestFile	LOC B	LOC C	LOC F	LOC G
Test_s1.txt	34.9374	11.993	64.5378	11.803
Test_s2.txt	11.4091	77.9769	117.315	40.2632
Test_u1.txt	103.735	109.402	254.608	41.6804
Test_u2.txt	151.815	76.6085	140.178	155.223

==

Compare Test Dataset by using 5GHz Frequency

TestFile	LOC B	LOC C	LOC F	LOC G
Test_s1.txt	367.541	171.548	92.2508	111.296
Test_s2.txt	406.444	312.272	249.908	168.407
Test_u1.txt	377.728	165.439	91.9234	114.071
Test_u2.txt	371.623	178.992	89.7713	95.9135

As shown on the left, I provided three tables that compare the distance base on 2.4 GHz frequencies, 5 GHz and use both frequencies to compare distances.

If we look at the 2.4 and 5 GHz frequency table, the distance between Test points to all detection points should not be longer than 5 meters and the Test points to each Detection location should be similar but the results show differently. Both stable and unstable Test Points compare to the Detection point database, and the result distances are much higher than my expectations, especially at the 5 GHz table.

By comparing the stable and unstable test points in 2.4 GHz, the distance value of the unstable is much higher than the non-stable dataset. but the distance to each Detection point is still unpredictable. In the 5 GHz table, the distance to each location is also messy and unpredictable, but not much difference between the stable and unstable test point dataset which may be because the distance of 5GHz is relatively large that cannot tell the difference.

From the experiment above, I believe it's not feasible to estimate where users' locations are, the better solution is to calculate which Test Point is closest to which Detection point and assume where the users are. If the distance between each detection point is not larger than 5 meters, theoretically we can get the user's location accurately. The distance between each detection point cannot be smaller than 2 meters based on my data collection experience, if the Detection points are too close to each other, the RSS data collection may have no difference, but if the distance is too large with RSS accuracy issue, the testing point may identify to another non-neighbor Detection point. So, I believe 5 meters is a reasonable deviation. Another conclusion from the above experiment is that the stable Test Points are better for the localization base on the 2.4 GHz table, but more datasets will be provided below for the user localization.

From my perspective, I don't think maintaining a Detection Point RSS database manually is a suitable solution for calculating the distance. By using Aruba Wireless Access points as an example, the Wireless Access point can be able to dynamically change the Input voltage to increase the transmitter signal gain or external antennas added to an Access Points that also affect the Database's accuracy. The goal of my algorithm is to be able to adjust the Database data automatically if datasets are provided.

Algorithm Design

My algorithm design is relatively simple and theoretically efficient. This is because my algorithm's user localization automatically reads the Detection Points input and compares it to Test Points in result the closest Detection points as the location of users. Since there are not many methods to control data, so the user localization is sensitive to the Input dataset accuracy. In fact, I cannot think of any method to increase the accuracy of user localization, at the end of the day, it is still data analysis that relies on input data.

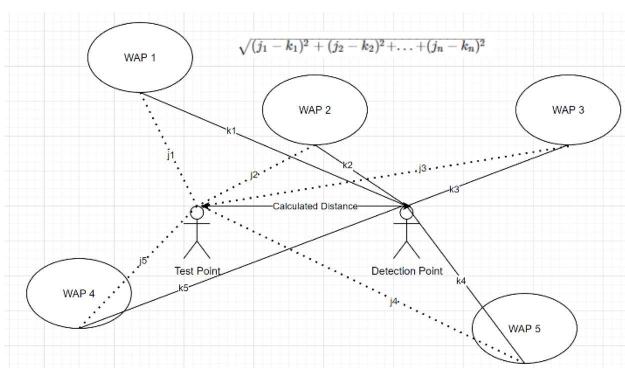
The First step of my WIFI fingerprinting algorithm is to convert the Raw data from Window API to a certain format that allows data comparison and processing. The data structure includes Python Dictionary and Array. Each dictionary store the SSID information with BSSID_Info that store each BSSID with the corresponding channel number and Signal Strength. But the Signal value is present in percentage, so I need to use the Channel number and the Signal Strength value to convert to distance in meters by using the formula "Free Space Loss Path Equation". The function of the description above the name "Pre_Analysis". Another issue is the Radiofrequency didn't specific on the Raw data, but I can determine the frequency type based on the Channel, this is because the legal 2.4 and 5 GHz channels are not overlapped in Australia.

```
# Clean Up the Data And store it
def Pre_Analysis(content):
    overall_dic = []
    Splited_SSID = split_ssids(content)
    for i in range(len(Splited_SSID)):
        SSID_Info = get_SSID_Info(Splited_SSID[i])
        overall_dic.append(SSID_Info)

    table_Array = []
    for i in range(len(overall_dic)):
        ssid = overall_dic[i]["SSID"]
        for j in range(len(overall_dic[i]["SSID_Info"]["BSSID_Info"])):
            if len(overall_dic[i]["SSID_Info"]["BSSID_Info"][j]) > 3:
                BSSID = overall_dic[i]["SSID_Info"]["BSSID_Info"][j][0]
                Signal = overall_dic[i]["SSID_Info"]["BSSID_Info"][j][1]
                Channel = overall_dic[i]["SSID_Info"]["BSSID_Info"][j][3]
                ssid_frequency = get_frequency(Channel)
                ssid_SignalStrength = get_SignalStrength(float(Signal.replace("%", "")))
                est_Distance = get_estDistance(ssid_frequency, Channel, ssid_SignalStrength)
                temp = [ssid, BSSID, ssid_frequency, Channel, est_Distance]
                table_Array.append(temp)

    return table_Array
```

The second step is to read the Test Point data and compare it with the Detection point data and compare to get the lowest value, so we know where users are located. Nothing fancy about the Test Point data, but there is an important point of the Detection Points Dataset is my algorithm can be able to read multiple Detections Points datasets to maintain its Database to compare with the Test Points, but the Detection Points must be at the same location for the data collection in each dataset. The benefit of this is it will eliminate the hotspot WIFI for the calculation to increase the accuracy. But it won't be able to eliminate the interferences that cause by Hotspots. Another benefit is it will get an average distance value to each WAPs that potentially mitigate the interferences caused by other Radiofrequency.



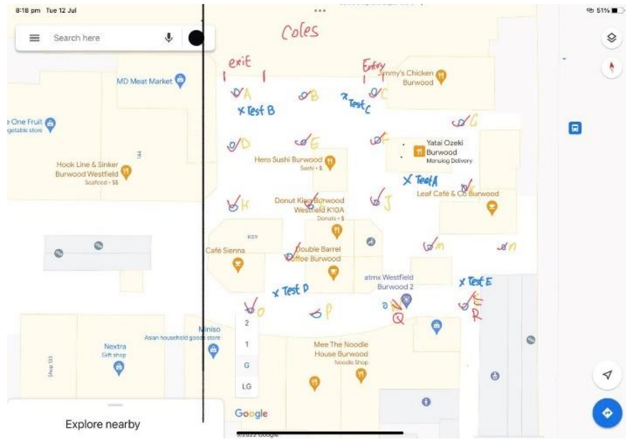
After comparing the distance between the Test Points and Detection Points, it will get the minimum value amount of the result and shows where the Test points are located.

The reason for using the N-dimensional Distance formula is because there are a large number of BSSID's RSS values that can be used to calculate the distance, by comparing the common SSID and BSSID and using each distance to the WAP to be able to calculate the distance between two data collection points.

Yuhua Zhao
Z5404443

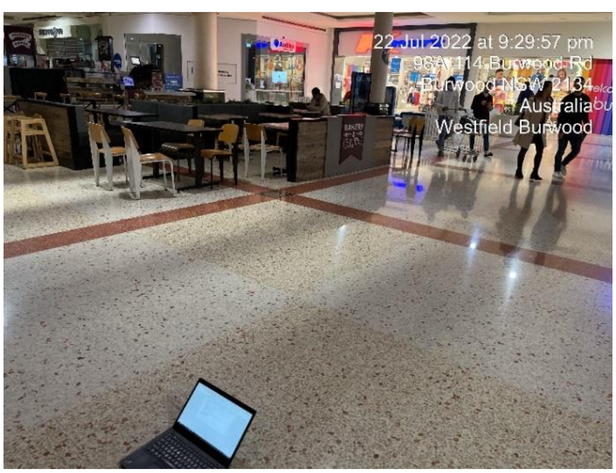
Project Data Collection Experience

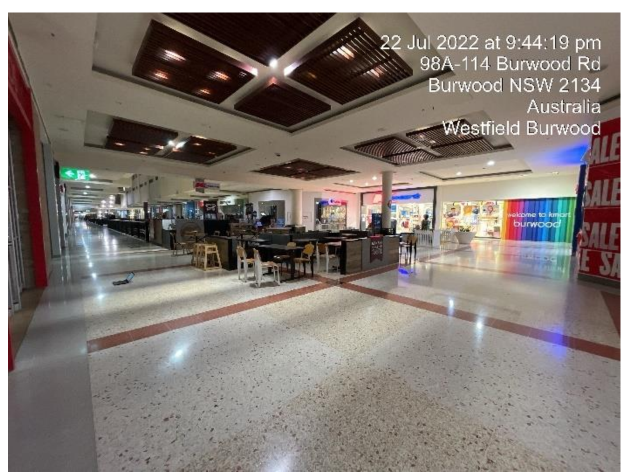
Originally, I selected Burwood Westfield Ground floor Near to Coles area as a location, but due to foot traffic and each dataset collection would take more than 30 minutes of time, I have no choice but to abundance the old dataset and change locations.



The second location that I selected is also Burwood Westfield but in Level 2 near to Kmart area. As mentioned above, I would need to attend the data collection at least twice to mitigate the effect of the hotspots.

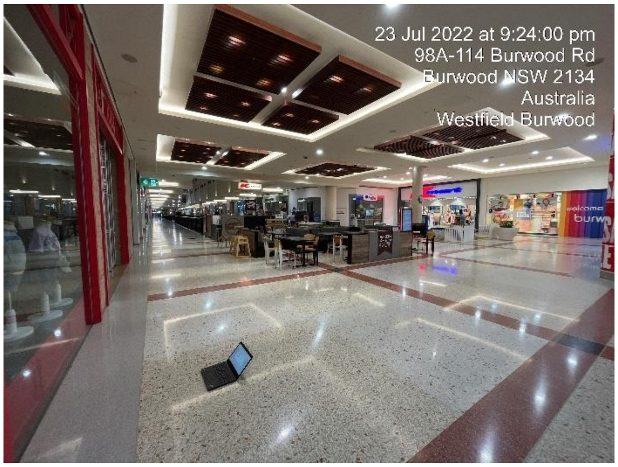
The First Dataset collection was done on 22/07/2022 and it took me more than 1 hour, this is because I collected 11 Detection Points, and for each location, I have to wait a minimum of 3 minutes to mitigate the Window OS cache effect, so the minimum spend time is 33 minutes for one dataset. On the same day, I also design and completed the data collections for Experiment 1 and other associated Test Point datasets. But unfortunately, the Test Point datasets got abandoned after my performance result analysis (Results are not stables). The images that are shown below weren't the start time and the end time, I took the photos in the middle of data collections.

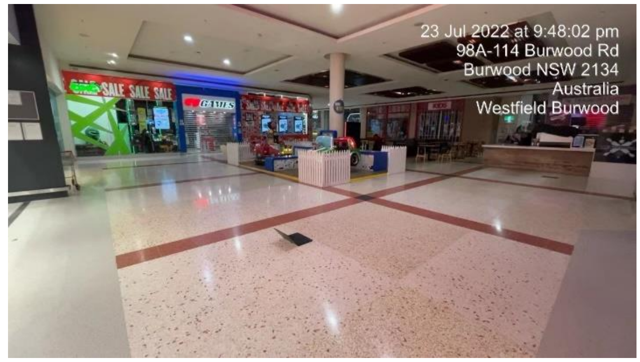




Yuhua Zhao
Z5404443

The Second Dataset collection was done on 23/07/2022 and also took me more than an hour for the data collection. I would need to collect the same Detection Point dataset at each identical location. Getting a stable Result for Each Testing point also took me a minimum of 30 minutes.





To get stable datasets, I also tried to maintain the same parameters for each dataset collection, for example, maintaining the same height, and same positions of the laptop for the Detection point datasets collection. But the Test Point data collection is relatively flexible as we cannot require users to put their handheld devices on the floor to get an accurate location.

Yuhua Zhao
Z5404443

Project Dataset Analysis

Dataset 1 - Information Summary:

Filename	NO. SSID	NO. BSSID	NO. 2.4GHz	NO. 5GHz
LocationA.txt	46	132	42	90
LocationB.txt	44	122	53	69
LocationC.txt	43	108	37	71
LocationD.txt	41	105	41	64
LocationE.txt	44	125	39	86
LocationF.txt	40	108	43	65
LocationG.txt	34	99	35	64
LocationH.txt	39	139	43	96
LocationI.txt	54	131	45	86
LocationJ.txt	53	125	37	88
LocationK.txt	43	126	59	67

Dataset 1 - 2.4 Frequency Channel Count:

Location	CH 1	CH 2	CH 2	CH 4	CH 5	CH 6	CH 7	CH 8	CH 9	CH 10	CH 11	CH 12	CH 13
LocationA	16	1	0	1	0	9	0	2	2	1	7	0	3
LocationB	13	1	1	0	0	13	4	1	1	2	15	0	2
LocationC	9	0	1	1	0	12	3	1	1	2	5	0	2
LocationD	7	0	1	0	1	12	4	1	1	2	10	0	2
LocationE	7	0	0	3	0	15	0	2	1	2	8	0	1
LocationF	8	0	0	1	0	13	3	0	1	2	13	0	2
LocationG	10	0	0	1	0	10	0	2	1	1	10	0	0
LocationH	10	0	1	1	1	11	3	2	1	1	11	0	1
LocationI	11	0	0	2	0	10	3	2	0	2	12	0	3
LocationJ	8	0	0	2	0	13	2	0	2	2	5	0	3
LocationK	14	0	0	2	1	17	3	2	1	1	15	0	3

Dataset 2 - Information Summary:

Filename	NO. SSID	NO. BSSID	NO. 2.4GHz	NO. 5GHz
LocationA.txt	50	139	37	102
LocationB.txt	42	109	35	74
LocationC.txt	40	127	33	94
LocationD.txt	31	103	42	61
LocationE.txt	33	105	24	81
LocationF.txt	42	131	31	100
LocationG.txt	35	95	32	63
LocationH.txt	34	107	37	70
LocationI.txt	44	117	37	80
LocationJ.txt	37	100	31	69
LocationK.txt	38	109	36	73

Dataset 2 - 2.4 Frequency Channel Count:

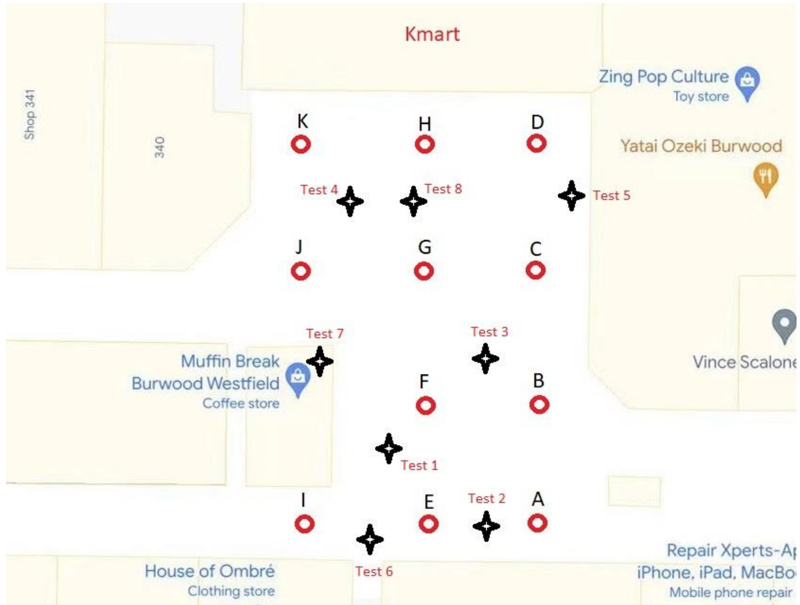
Location	CH 1	CH 2	CH 2	CH 4	CH 5	CH 6	CH 7	CH 8	CH 9	CH 10	CH 11	CH 12	CH 13
LocationA	9	0	0	1	1	9	0	0	0	3	12	0	2
LocationB	12	0	0	1	0	10	0	0	1	1	8	0	2
LocationC	12	0	0	2	1	9	3	0	0	1	5	0	0
LocationD	7	0	0	0	0	15	3	0	1	2	13	0	1
LocationE	5	0	0	2	0	10	0	0	0	1	4	0	2
LocationF	8	0	0	2	0	9	1	0	1	2	7	0	1
LocationG	5	0	0	1	0	11	1	1	1	0	11	0	1
LocationH	6	0	0	1	0	12	3	0	1	1	12	0	1
LocationI	8	0	0	3	1	12	0	0	1	3	8	0	1
LocationJ	5	0	0	2	0	14	0	1	1	1	6	0	1
LocationK	7	0	0	2	0	9	3	1	1	2	10	0	1

As shown on the left-hand side, by comparing both detection datasets, we can see that the number of SSID, BSSID, and 2.4 GHz and 5 GHz are similar. The reason why the numbers are different potentially because of three reasons. Firstly is because of Hotspots Wifi added or removed. The second reason is because of the Radio Frequency interference that causes some of the SSID or BSSID cannot be discovered. The third reason also mentioned above, for the High-end WAP, the transmitter signal strength changes automatically based on WIFI coverage and the interferences that affect the data collections.

Due to there being a large number of 5 GHz frequency channels and the 5 GHz channel can use the Channel bonding method that is hard to identify which exact channel it occupies. So we mainly look at the 2.4 GHz channel usage. As we can see Channels 1, 6, and 11 are mainly occupied by the WIFI router due to pre-set from the manufacturer. This means that the 2.4 GHz datasets that I collected for the Detection Point database are under very large frequency interference.

Project Performance Analysis

As the screenshot is shown on the right-hand side, the red circle is the detection point that are used to identify users' location. The black stars are the test points.



This Result get 2.4 GHz Frequency:

File	LOC A	LOC B	LOC C	LOC D	LOC E	LOC F	LOC G	LOC H	LOC I	LOC J	LOC K	Result
Test1.txt	156.27	229.357	126.562	259.3	252.965	169.718	64.5499	419.655	274.737	111.226	384.966	LOC G
Test2.txt	122.537	34.1409	57.844	184.712	269.462	50.2259	75.1358	298.372	275.26	158.288	273.142	LOC B
Test3.txt	42.5519	16.4084	8.95912	184.629	83.9593	13.8992	6.75701	55.7516	15.393	138.147	226.526	LOC G
Test4.txt	36.1569	177.579	126.78	275.907	246.783	179.872	55.5976	430.49	268.713	133.73	248.038	LOC A
Test5.txt	29.9107	37.2319	50.7851	311.055	193.764	156.6	39.4238	360.193	191.925	179.349	216.529	LOC A
Test6.txt	41.5849	182.126	123.334	211.442	131.584	50.7333	13.1047	120.004	93.624	166.926	161.002	LOC G
Test7.txt	136.99	222.176	136.159	159.563	17.476	63.7203	83.4651	176.967	37.6264	63.0784	141.096	LOC E
Test8.txt	206.616	169.27	90.5129	257.868	120.591	157.713	82.2253	310.368	132.046	159.292	101.944	LOC G

As shown on the screenshot above, the Test Point that successfully identifies to the neighbour Detection only have 37.5% which is Test 2, 3, 8 that can be able to neighbour Detection points.

This Result get 5 GHz Frequency:

File	LOC A	LOC B	LOC C	LOC D	LOC E	LOC F	LOC G	LOC H	LOC I	LOC J	LOC K	Result
Test1.txt	97.6488	215.245	131.924	90.8767	38.2384	42.6247	63.2036	86.4214	45.8992	57.6524	54.161	LOC E
Test2.txt	69.1132	186.824	100.57	70.6041	64.3258	44.5428	58.0993	70.3306	48.9104	50.1222	67.5943	LOC F
Test3.txt	63.2595	177.731	95.075	49.0144	53.6754	32.2935	49.1312	54.8078	31.9565	40.4156	48.6835	LOC I
Test4.txt	80.8267	93.5287	55.1077	78.7018	49.8971	46.9065	44.1155	93.8978	34.7874	49.6354	40.1266	LOC I
Test5.txt	47.6928	95.2891	67.4513	87.0783	16.1386	26.9798	52.0526	102.708	31.0687	41.3768	43.5791	LOC E
Test6.txt	113.222	232.614	173.004	100.031	78.8717	51.9657	91.5817	116.262	64.1292	75.0154	60.7338	LOC F
Test7.txt	90.332	220.51	155.196	70.4763	53.5736	45.3697	80.2975	75.6196	44.2067	42.0778	54.3157	LOC J
Test8.txt	88.6137	204.194	122.889	69.4901	50.0763	29.3244	50.1921	63.8154	37.2756	36.3919	48.6964	LOC F

As shown on the screenshot above, the Test Point that successfully identifies to the neighbour Detection only have 50% which is Test 1, 2, 6, 7 that can be able to neighbour Detection points.

Yuhua Zhao
Z5404443

This Result get 2.4 and 5GHz Result:

File	LOC A	LOC B	LOC C	LOC D	LOC E	LOC F	LOC G	LOC H	LOC I	LOC J	LOC K	Result
Test1.txt	184.271	314.54	182.817	274.763	255.839	174.989	90.3404	428.461	278.545	125.28	388.757	LOC G
Test2.txt	140.684	189.918	116.018	197.746	277.034	67.1319	94.9786	306.549	279.572	166.034	281.382	LOC F
Test3.txt	76.2393	178.487	95.4962	191.024	99.6505	35.1576	49.5937	78.1801	35.4706	143.937	231.698	LOC F
Test4.txt	88.5454	200.703	138.239	286.912	251.777	185.887	70.9737	440.612	270.956	142.644	251.263	LOC G
Test5.txt	56.2961	102.305	84.4323	323.014	194.435	158.907	65.2971	374.55	194.423	184.06	220.871	LOC A
Test6.txt	120.618	295.43	212.466	233.91	153.411	72.6244	92.5146	167.086	113.481	183.007	172.076	LOC F
Test7.txt	164.092	313.028	206.458	174.434	56.3519	78.222	115.819	192.446	58.0516	75.8249	151.19	LOC E
Test8.txt	224.817	265.232	152.624	267.067	130.575	160.416	96.334	316.861	137.207	163.396	112.977	LOC G

As shown on the screenshot above, the Test Point that successfully identifies to the neighbour Detection only have 62.5% which is Test 2, 3, 5, 6, 8 that can be able to neighbour Detection points.

The performance calculation above is not strict, as long the Test Point match with either one of the four Neighbour Detection points considers correct. And the combination of 2.4 and 5 GHz gets the best accuracy. The result is not surprising as WIFI Fingerprinting is hard to achieve high accuracy due to the limitations of WIFI performance. The accuracy extremely relies on the Data that feed to the algorithm, the more accurate result that is provided, the more accurate the user localization you can have.

Conclusion

WIFI Fingerprinting used as user localization is possible according to the experiments that I have done above, but accuracy would be a drawback due to unstable RSS data collection. According to the Project Performance analysis, I can see that in use both frequencies for this location is better than simply using 2.4 GHz or 5 GHz due to the limitation of both frequencies. In this scenario, using both frequencies can mitigate the 2.4 GHz signal interferences as well as the 5 GHz bad obstacle penetration. But from my experience, the usage of frequency to user localization relies on the indoor environment, for example in an open area without too many obstacles like an indoor office, 5 GHz will be a better solution but in a Warehouse that may have a lot of stocks, 2.4 GHz will be better solutions. But Westfield shopping centers have number of glass and other interferences, 2.4 GHz and 5 GHz would be better solutions.

Getting a larger dataset may in result a higher accuracy but as mentioned above, each RSS data collection requires a minimum of 3 minutes of waiting time in a public area which is difficult due to the high foot traffic in shopping centres. I understand the volume of datasets may not be enough, but the quality of the datasets is high. During this project, I learned that data collection is not a simple thing, especially for high-quality data.

Reference:

1) Ho, Q-D, Le-Ngoc, T & Tweed, D 2017, Long Term Evolution in Unlicensed Bands, Springer International Publishing, Cham.
2) Marozzi, M., Mukherjee, A. and Kalina, J., 2020. Interpoint distance tests for high-dimensional comparison studies. *Journal of Applied Statistics*, *47*(4), pp.653-665.