SMART-TRACE: A COMBINED RFID DATABASING AND WIFI-RSSI FINGERPRINTING INDOOR POSITIONING FOR DISASTER MANAGEMENT USING SMARTPHONE

An Undergraduate Thesis

Presented to

The Faculty of the Electronics Engineering Program
College of Engineering Education
University of Mindanao

In Partial Fulfillment
of the Requirements for the Degree
Bachelor of Science in Electronics Engineering

By

Edwin G. Ligan Jr.
Reyden Jenn R. Cagata
Shen Wa S. Lai

August 2024

APPROVAL SHEET

This undergraduate thesis entitled, " **SMART TRACE: A COMBINED RFID DATABASING AND WIFI-RSSI FINGERPRINTING INDOOR POSITIONING FOR DISASTER MANAGEMENT USING SMARTPHONE**" prepared and submitted by Reyden Jenn Cagata, Edwin Ligan Jr., and Shen Wa Lai, has been examined, accepted, and is hereby endorsed.

ENGR. RANDY E. ANGELIA
Thesis Adviser

PANEL OF EXAMINERS

Approved by the Committee on Oral Examination

ENGR. JETRON J. ADTOON
Committee Chair

ENGR. ELENA MATILLANO -PEREZ
Member

Accepted in partial fulfillment of the requirements for the Degree of **Bachelor of Science in Electronics Engineering.**

DR. CHARLITO L. CAÑESARES, PME
Dean, College of Engineering
University of Mindanao

ACKNOWLEDGEMENT

First and foremost, we are deeply grateful to our families, whose unwavering financial and emotional support provided the foundation for our success.

We are also profoundly thankful to our research adviser, Engr. Randy E. Angelia, for his invaluable guidance and mentorship. Our sincere thanks go to our research statistician, Engr. Rolieven P. Cañizares, for his crucial assistance with the statistical aspects of our research. We would also like to express our deep appreciation to Engr. Jetron Adtoon, our Research Coordinator, and Engr. Elena Matillano-Perez, one of our panelists, for their constructive criticism and insightful feedback. We are equally grateful to Engr. Bernard Camacho, Engr. Paolo Gavino, and Engr. Hannah Angelia, for granting us permission to conduct our experiments and research within their laboratory facility.

We also want to thank Sir Jourdan Salazar, the supervisor of the university's ICTC, for sharing his knowledge on access point technology and explaining it in a way that was easy for us to understand. Additionally, we are grateful to Sir Candido O. Suralta, the chief of the Security and Safety Office (SSO), for providing valuable information on the university's safety measures. Also, we want to express our gratitude to Engr. Chardie Muaña, for also sharing his expertise as a Cisco Engineer and Mikrotik Engineer, which really answered our questions regarding the technicalities around access point routers. Also, thanks to Engr. Jay Al Gallenero for providing additional references for our thesis.

We also want to thank Robert Paulo Murillo, Kenny Rey Guillen, Francis Cofreros, Justin Mark Morales, and Lady J-Anne Loyola for lending us their devices, which were fundamental in making our data gathering successful.

We especially want to extend our heartfelt gratitude to our senpai, John Carlo Lisondato, for his support by providing insightful feedback and technical assistance.

Lastly, but most importantly, we give our deepest thanks to God for providing us with the strength, motivation, wisdom, and understanding to overcome the many challenges and obstacles we faced throughout this project. His divine guidance has been our source of inspiration, enabling us to persevere and achieve our goals. To everyone who contributed to this journey, we are sincerely grateful. Your support has been invaluable, and we could not have achieved this milestone without you.

E.L.J

R.J.C

S.W.L

ABSTRACT

As a widely used device, smartphones have never been integral to disaster management. GPS might be a solution, but it is not accurate for indoors. Hence, SMART-TRACE used smartphone mobile hotspots to locate potential victims inside a building using an interactive map through a website. This study combined radio frequency identification (RFID) and WiFi technology to locate positions inside buildings, called the indoor positioning system (IPS). The parameter used in the study to predict the users is the Received Signal Strength Indicator (RSSI), commonly emitted by smartphones. Then, these signal strengths were surveyed ahead of time, called fingerprinting, which provides excellent assistance in predicting smartphone users. Additionally, this study evaluated SMART-TRACE's capability to identify possible victim locations on the map through several machine-learning algorithms. Among other position algorithms used in the study, the k-nearest neighbours (KNN) algorithm prevails in terms of its accuracy of 99.93% with its calculation time of 0.2s. Results also show a 100% accuracy of the UHF RFID reader with an average time reading interval of 0.923s. At the same time, WiFi-RSSI IPS covering three rooms, and two floors has an overall accuracy of 97.22% in locating smartphones on the map with an average position error of 4.88m.

Keywords: Indoor Positioning System (IPS), KNN algorithm, Controlled Access Point system Manager (CAPsMAN), crowdsourcing, multivariate linear regression analysis

SDG Indicator: #9 (Industry, Innovation, and Infrastructure), #11 (Sustainable Cities and Communities)

TABLE OF CONTENTS

TITLE PAGE i

APPROVAL SHEET ii

ACKNOWLEDGEMENT iii

ABSTRACT iv

TABLE OF CONTENTS v

LIST OF TABLES vi

LIST OF FIGURES vii

Article: SMART-TRACE: A Combined RFID Databasing and WIFI-RSSI Fingerprinting Indoor Positioning for Disaster Management Using Smartphone

I. Introduction 1

II. Materials and Methods 2

III. Results and Discussions 5

IV. Conclusions and Future Works 8

References 8

APPENDICES

Appendix A: Trade-off Analysis 10

Appendix B: Data Sheets 13

Appendix C: Data Gathering and Analysis 19

Appendix D: RSS Data Summary 28

Appendix E: Long-Term Signal Analysis 32

Appendix F: Heatmap 37

Appendix G: Calibration Coefficient and Linear Regression Analysis 62

Appendix H: Source Code 72

Appendix I: User Manual 99

Appendix J: Informed Consent 110

Appendix K: Gantt Chart 113

Appendix L: Production Cost Summary 116

Appendix M: Research Documents 118

LIST OF TABLES

Table 1. Time Reading Interval of the UHF RFID Reader 5
Table 2. Device ID and SSID 5
Table 3. Number of Distinct RP and RSS Data 6
Table 4. Relative and Position Error of the WiFi-RSSI IPS 7
Table 5. Confusion Matrix of the WiFi-RSSI IPS 7
Table 6. Confusion Matrix of Six Simultaneous Devices 7

LIST OF FIGURES

Fig. 1. Conceptual Framework 2

Fig. 2. Functional Block Diagram of the SMART-TRACE System 2

Fig. 3. Location of the Access Points 3

Fig. 4. Proposed RFID Reader Locations 3

Fig. 5. Plotted Reference Points of the room of the BE Building on the Map 3

Fig. 6. Flowchart of the Offline Training Phase 4

Fig. 7. Flowchart of the Online Determination Phase 4

Fig. 8. RS232/RS485/WG26/34 UHF RFID Reader 5

Fig. 9. Position of the Reference Points (RPs) 5

Fig. 10. Reference Points Vs. Number of RSS Data 6

Fig. 11. Heatmap 6

Fig. 12. Long-Term Signal Analysis and the average signal strength from the Six Smartphones 6

Fig. 13. Signal Strength Count from Different Smartphones 7

Fig. 14. Calibration Coefficients from Device ID 1,2,3, and 5 7

Fig. 15. Graphical User Interface of SMART-TRACE System 8

SMART-TRACE: A Combined RFID Databasing and WiFi-RSSI Fingerprinting Indoor Positioning for Disaster Management using Smartphone

Reyden Jenn Cagata
Electronics Engineering Program
College of Engineering Education
University of Mindanao, Davao City, Philippines
reydencagata@gmail.com

Shen Wa Lai
Electronics Engineering Program
College of Engineering Education
University of Mindanao, Davao City, Philippines
shennylie0919@gmail.com

Edwin Ligan Jr.
Electronics Engineering Program
College of Engineering Education
University of Mindanao, Davao City, Philippines
edgallanoligan@gmail.com

Abstract—**As a widely used device, smartphones have never been integral to disaster management. GPS might be a solution, but it is not accurate for indoors. Hence, SMART-TRACE used smartphone mobile hotspots to locate potential victims inside a building using an interactive map through a website. This study combined radio frequency identification (RFID) and WiFi technology to locate positions inside buildings, called the indoor positioning system (IPS). The parameter used in the study to predict the users is the Received Signal Strength Indicator (RSSI), commonly emitted by smartphones. Then, these signal strengths were surveyed ahead of time, called fingerprinting, which provides excellent assistance in predicting smartphone users. Additionally, this study evaluated SMART-TRACE's capability to identify possible victim locations on the map through several machine-learning algorithms. Among other position algorithms used in the study, the k-nearest neighbours (KNN) algorithm prevails in terms of its accuracy of 99.93% with its calculation time of 0.2s. Results also show a 100% accuracy of the UHF RFID reader with an average time reading interval of 0.923s. At the same time, WiFi-RSSI IPS covering three rooms and two floors has an overall accuracy of 97.22% in locating smartphones on the map with an average position error of 4.88m.**

Keywords*—Indoor Positioning System (IPS), KNN algorithm, Controlled Access Point system Manager (CAPsMAN), crowdsourcing, multivariate linear regression analysis

I. Introduction

Knowing the whereabouts of every person inside a premises is essential during an emergency. GPS might be a solution; however, it is primarily applicable to outdoor environments as its signals cannot penetrate buildings due to their low signal strength and accuracy [1]. Also, rescuers must locate potential casualties before it is too late to provide an effective, quick emergency response. Hence, the researcher designed an indoor positioning system (IPS) using smartphone WiFi to address this discrepancy.

IPS is an approach that can locate positions inside a building. According to [2], there are several types of IPS technology in the radio frequency spectrum, such as WiFi, Bluetooth Low Energy (BLE), Radio Frequency Identification (RFID), and Ultrawideband (UWB). Based on the study of [3], UHF RFID has been reported to track users with low power consumption easily. However, RFID and BLE used for IPS have a short range, which is unsuitable for long-range applications [2, 4]. Also, UWB has a broader range; however, it requires higher costs due to its complex circuitry and availability [2, 5]. Among these, WiFi prevails in terms of its low cost, more comprehensive range than RFID and BLE, and the most accessible technology among other types of IPS [2, 6]. Furthermore, most of the research that studied IPS [7, 9, 11] utilizes Received Signal Strength Indication (RSSI) as the representative range-based positioning because of its simplicity and effectiveness. Other indicators used for IPS are time of arrival (TOA) [8], time difference of arrival (TDOA) [9], and angle of arrival (AOA) [10].

Consequently, three primary WLAN RSS techniques for locating clients are the Strongest Base Station, Propagation Model, and Fingerprinting-based method. The Strongest Base Station is the simplest method that estimates the user's location nearest to the Access Points (AP) [11]. Also, the Propagation Model, which includes trilateration and triangulation methods, requires at least three APs to locate an object location [2, 11]. This method is low-cost and straightforward; however, it has low accuracy in a large area and proved unreliable and unpredictable [4, 11-12]. In addition, these studies [4, 13-14] find that the fingerprinting-based method has more accurate results as it cannot be affected by path loss due to its ability to record long-term RSS data. In addition, these algorithms use two basic equations: the Bayes Rule [15], which estimates the location through a probabilistic method, and the Euclidean equation. Other relevant studies [6-7, 9, 11] show

that applying matching algorithms increases the accuracy of the IPS such as Binary Relevance [16], Classifier Chain and Label Powerset [17], k-nearest Neighbor (KNN) [6-7, 9, 11], and Multilabel KNN (MLKNN) [18]. IPS can also be improved through a statistical technique such as Principal Component Analysis (PCA) through noise dimension reduction [19].

Despite the growing interest in IPS using WiFi-RSSI and RFID databasing, more research is needed on applying indoor localization in disaster management using WiFi integrated with RFID technology. Most studies only record attendance using RFID and do not account for indoor localization. Also, other studies account for IPS but do not account for users' information.

The study's general objective is to combine RFID databasing and WiFi-RSSI indoor positioning for disaster management using smartphones. Specifically, the objectives of this study are (a) to create an indoor positioning system using fingerprinting-based WiFi-RSSI, (b) to integrate both the RFID database and WiFi-RSSI indoor positioning for disaster management using a smartphone, and (c) to test its accuracy by conducting a functionality test using the appropriate statistical tool.

Moreover, this research study provides valuable data, especially for government agencies specializing in disaster risk reduction and disaster management. Also, private or commercial buildings such as schools, hospitals, or malls will benefit from the technology for enhancing disaster management. Through this, users will gain support to improve their safety during a disaster. Also, future researchers will benefit from the study that the researchers conducted.

The system focuses on a specific area of interest, which covers two floors at the BE building of the University of Mindanao, precisely three rooms: BE111, BE213, and BE216. This study may only cover three rooms; however, the study will also apply to large internet-based buildings as the methods mentioned are similar. Also, the scope of the study records valuable information such as names. More importantly, the study uses smartphone brands, specifically Realme 6 Pro, RedmiNote 11, Techno Camon 20, Vivo Y21T, Xiaomi 10T, and Xiaomi Poco X3.

II. Materials and Methods

A. Conceptual Framework

This study focuses on creating an indoor positioning system. Fig. 1 shows how the system works from input to output. RFID tag and the smartphone act as the system's input. While RFID tracking, fingerprinting, and position algorithms are processes that aid in achieving the final output.

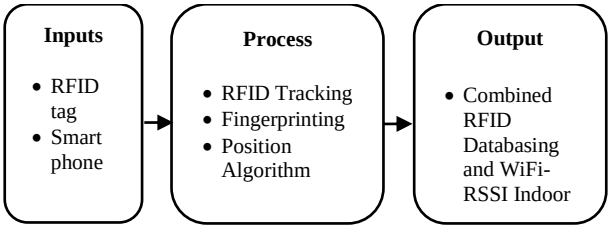


Fig. 1 Conceptual Framework

B. Materials and Resources

Materials used in the study are the; RS232/RS485/WG26/34 UHF RFID Tags/Receivers (902MHz-928MHz), used for identifying IDs; ESP8266 microcontroller, which transfers data from the RFID to the computer through Wi-Fi technology; Then, the computer used is Predator Helios Neo 16, a 13th Gen Intel® Core™ i7-13700HX 2.10 GHz; And Mikrotik hAP Lite Routers as Access Points (APs), which can expand Wi-Fi coverage by being strategically placed in various locations.
Additionally, to administer databases, XAMPP was used to store user data in a MySQL cloud database. Also, the programming languages used are C++ for the ESP8266 microcontroller, Python, which automates data-gathering and implementing offline and online positioning algorithms, and PHP for website creation for the GUI of the SMART-TRACE system. Lastly, the proponents used Leaflet, a JavaScript library, for the interactive map.

C. Methods and Procedures

The skeletal framework of the SMART-TRACE system, as shown in Fig. 2, displays the connection between devices. It begins with a UHF RFID Reader/Tag for tracking and attendance. ESP8266 then receives the data from the UHF RFID Reader and sends it to the computer. Also, the AP can receive information from the smartphone's mobile hotspot through the AP's ability to sniff data packets.

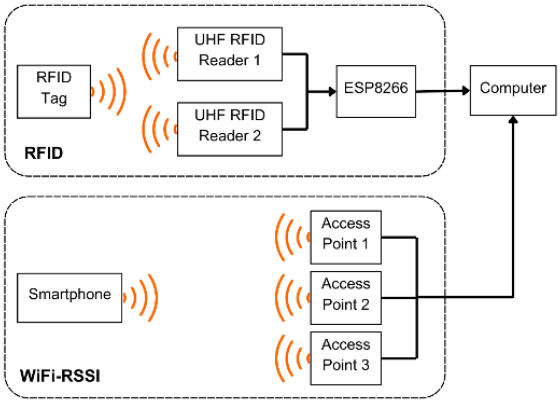


Fig. 2 Functional Block Diagram of the SMART-TRACE System

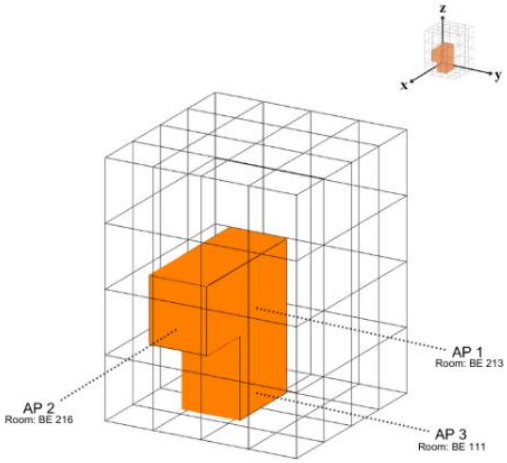


Fig. 3 Location of the Access Points

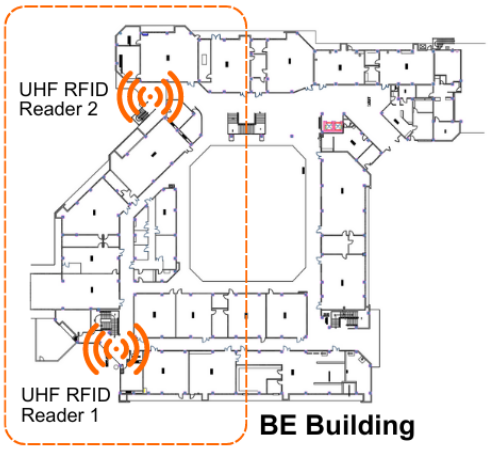


Fig. 4 Proposed RFID Reader Locations

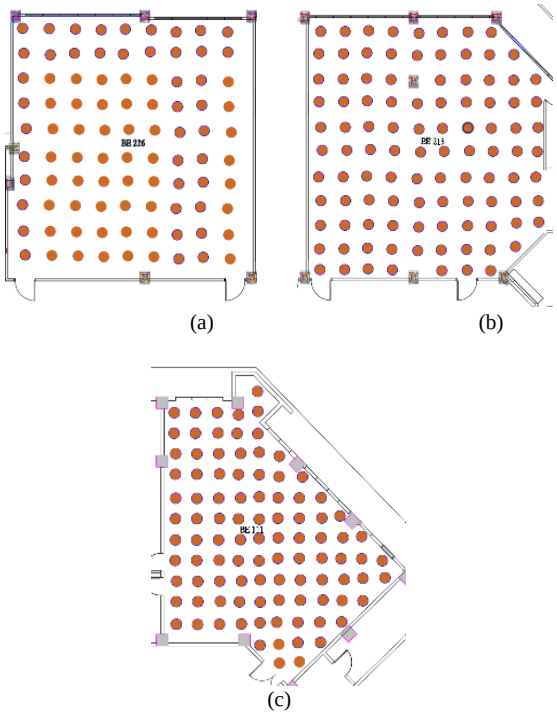


Fig. 5 Plotted Reference Points of the room of the BE Building on the Map: (a) BE 111 (b) BE 216 (c) BE 213

D. Design

A trade-off analysis was conducted by the researchers. It shows that the design provides the most suitable design with factors such as accuracy, convenience, and economic prowess.

Knowing whether people are still inside the building despite not having a device is essential during an emergency. Hence, the researcher placed UHF RFID readers at two entrances of the BE Building, as shown in Fig. 4, as an alternative for smartphones. Then, the proponent collects information through the UHF RFID reader/tag. Before collecting personal information, respondents received an informed consent form regarding information security and data privacy.

The user's smartphone should act as a Wireless Access Point (WAP) or mobile hotspot for the router to receive the user's RSS data [20]. More importantly, Mikrotik routers may publicly see mobile hotspots. However, specific devices are only visible—those registered devices to the system. To obtain RSS data from a mobile hotspot, the researcher used the CAPsMAN (Controlled Access Point system Manager) function of the Mikrotik routers to centralize data management. This way, the researcher can control the system through a single or main router. Also, to automate the system, a Python code commands the router through the router's Secure Socket Shell (SSH).

The APs were placed on two floors. Furthermore, fingerprinting is the collection of RSS data from a Reference Point (RP) within an area of interest. There are two phases for fingerprinting: the offline training phase and the online determination phase.

E. Offline Training Phase

In the offline phase, the first step is to divide the area of interest into sets of points corresponding to a specific Reference Point (RP) represented in a cartesian coordinate form, each representing latitude and longitude, as shown in Fig. 5.

After the data collection, the data undergoes preprocessing by merging similar rows, eliminating extraneous data. Data are processed through filtering, combining, concatenation, Cartesian Product data combination, cleaning, and splitting. Finally, a radio map or the database of RSS readings is produced at the end of the offline phase. The total number of RSS data in the radio map is calculated using the equation (1).

$$R = \sum_{i=1}^{n}\left(\prod_{j=1}^{m} RSS_j\right) \quad (1)$$

$$n = \prod_{j=1}^{m} AP_j \quad (2)$$

Where:
m = *total number of APs*
RSS_j = *total number of RSS value in each channel*
AP_j = *total number of channels in every AP*
n = *total number of combination of channels in each RPs*

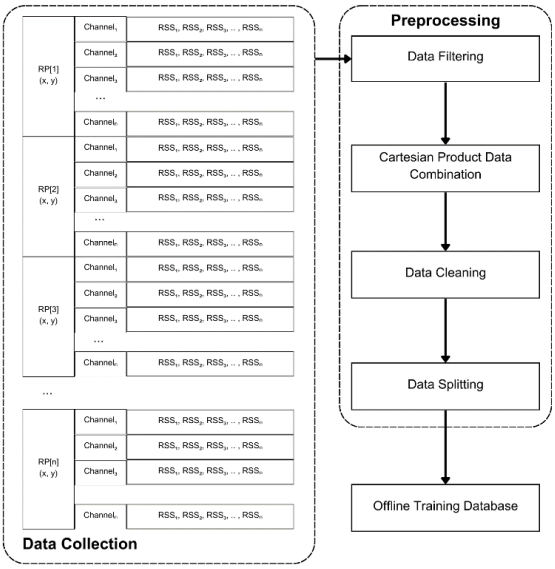


Fig. 6 Flowchart of the Offline Training Phase

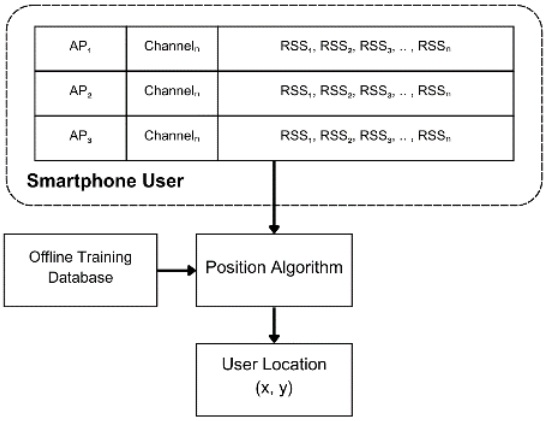


Fig. 7 Flowchart of the Online Determination Phase

To collect the RSS data from each RP, the researcher utilized a laptop that scans the RSS data for two minutes for every RP and proceeds through the outlined process as depicted in Fig. 6.

F. Online Determination Phase

The online phase is a process, as shown in Fig. 7, wherein specific points are measured inside the area of interest. Then, the system used position algorithms to analyze the measured points with the radio map. Finally, the most comparable fingerprint is returned as the estimated position using the Euclidean equation (3).

$$d = \sqrt{\sum_{i=1}^{m}(RSS_i - RSS_i')^2} \quad (3)$$

Where n is the number of APs, RSSi is the ith AP's signal strength received (online), and RSSi' is the average RSS value (offline). The proponents plan to utilize a machine learning localization algorithm, specifically the kNN algorithm, to increase the accuracy of the user's location.

G. Statistical Analysis

First, to test the accuracy of the device's location, different RP sample sizes for the fingerprint were utilized: 100, 200, and 300 RPs. Then, two reference smartphones recorded RSS data from various fingerprint sample sizes.

Second, the researchers conducted a total of 180 trials to find the accuracy of the different smartphones. Then, the proponents compared the estimated distance with the actual distance to find the error level in percentage.

Moreover, several mobile devices were frequently involved wherein different RSS values were measured, which can pose additional issues. Hence, crowdsourcing enabled the system to reduce errors by comparing the varied fingerprints of multiple smartphones [21].

A calibration must be performed prior to the position algorithm. A viable option is to calibrate the device using a data-fitting method that generates a linear transformation from the new device to the reference device. Then, the corrected RSSs merged to form crowdsourcing training data. One approach would be to calculate an offset for each smartphone's average RSSs and then added to the RSS data. A multivariate linear regression estimates the data as a solution. The average value of an AP's measured RSS at each reference point location can then be computed for each smartphone. All device values can be averaged for the regression's reference vector. Only the APs witnessed exclusively with each smartphone are used. The linear regression model can have the following basic form:

$$y = a_S x_S + b_s \quad (4)$$

Where y is the average reference vector calculated from all average RSS values estimated with the linear regression model, and x is the measured RSS values from the smartphone S. At the same time, a_S and b_s are calibration coefficients determined individually for each frequency band. Also, a long-term signal propagation analysis studies the impact of indoor localization systems. The analysis could be analyzed by scanning a single RP and measuring RSS values in an hour.

Position estimation error measures the error between the location of two coordinates using the Haversine formula (7).

$$a = \sin^2\left(\frac{\Delta\varphi}{2}\right) + \cos\varphi_1 + \cos\varphi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \quad (5)$$

$$b = 2 \cdot \mathrm{atan}\left(\sqrt{a}, \sqrt{1-a}\right) \quad (6)$$

$$Position\ Error = R \cdot c \quad (7)$$

Where: $\varphi = \mathrm{latitude}, \lambda = longitude;\ where\ angles\ are\ in\ radians$
$R = earth's\ radius\ (R = 6.371km)$

It shows that RFID tags and smartphones are essential materials for the first part of the process. Finally, the system's performance uses a confusion matrix to define its accuracy. The confusion matrix

in Table V uses equation (8) to determine the system's overall accuracy.

$$Overall\ Accuracy = \frac{(A,A)+(B,B)+(C,C)}{Total\ No.of\ Samples} \quad (8)$$

Where: $A =$ BE 213, $B =$ BE 216, $C =$ BE 111

Also, there are different scenarios to test the accuracy of the WiFi-RSSI IPS. There are seven scenarios, as shown in Table VI. Scenario 1 (S1) shows six simultaneous smartphones that are tested, and then Scenario 2 (S2) has five simultaneous smartphones up to Scenario 7 (S7), which has zero devices.

H. Functionality Test

The researchers collected 300 RSS fingerprints using two different phone models on different floors of the BE building, placed at random locations that the APs cover, to test the functionality of the study. Also, there were three APs located in different rooms. Finally, there was a Graphic User Interface (GUI) for the users' positions.

III. Results and Discussions

A. RFID Reader

The researcher tested the accuracy of the UHF RFID Reader and yielded a 100% success rate of reading the UHF RFID sticker for identifying the user after 50 trials. Also, the time reading interval between five simultaneously tested UHF RFID stickers were tested as shown in Table I. It presents that the first interval provides the least interval time with a dispersion of 0.308 around its mean. Even with a minimum interval of 0.308, the maximum interval is due to the existence of the previous stickers that the UHF Reader is still reading. The setup for the UHF RFID reader is shown in Fig. 8.

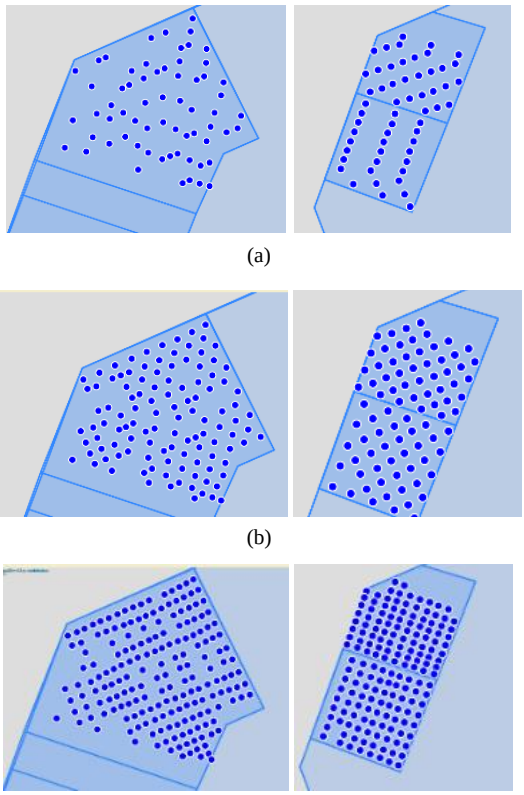
TABLE I. Time Reading Interval of the UHF RFID Reader

	Time Reading Interval (seconds)			
	1	*2*	*3*	*4*
μ	0.923	1.500	1.918	2.680
σ	0.308	0.392	0.307	0.591
Max	1.996	3.591	4.524	4.655
Min	0.308	0.392	0.307	0.591

B. Access Points

The APs' positions were strategically positioned so they could read RSS data within the scope of the study. These particular AP sites were chosen based on their capacity to track and record the signal strength coming from devices in the said locations, ensuring comprehensive coverage throughout the area of interest. These entry points have been placed strategically in the corners to reduce obstruction. In this way, a person can be tracked on different building floors.

Fig. 8 RS232/RS485/WG26/34 UHF RFID Reader



(a)

(b)

(c)

Fig. 9 Position of the Reference Points (RPs)— First floor (left side) and Second floor (right side): (a) 100 RPs, (b) 200 RPs, (c) 300 RPs

C. Offline Training Phase

During the offline phase, the researchers assigned 100, 200, and 300 reference points (RPs) from three rooms using the system's GUI. Also, this study used two reference devices to gather RSS data, mainly Xiaomi Poco X3 and Vivo Y21T. Table II shows the SSID used in the study with its Device ID.

TABLE II. Device ID and SSID

Device ID	SSID
1	Realme_6Pro
2	RedmiNote_11
3	TechnoCamon_20
4	Vivo_Y21T
5	Xiaomi_10T
6	Xiaomipoco_X3

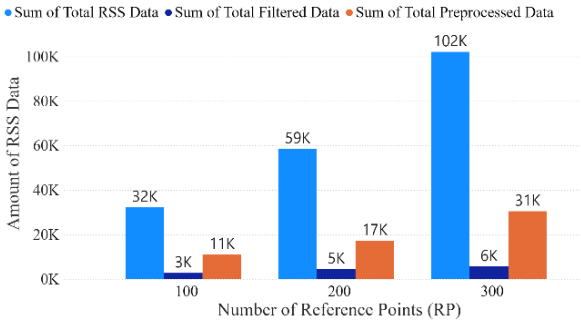


Fig. 10 Reference Points Vs. Number of RSS Data

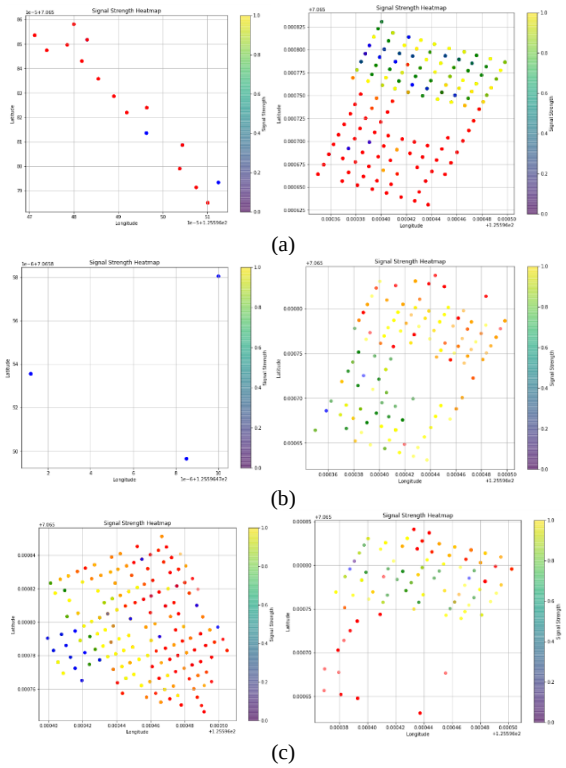


Fig. 11 Heatmap—First floor (left side) and Second floor (right side): (a) AP 1, channel 2452 (right side), channel 2462 (left side), (b) AP 2, channel 2452, (c) AP 3, channel 2427 (right side), channel 2452 (left side)

Next, the proponents proceed with the offline training by following the processes shown in Fig. 6. To prevent losses with its SSID name, the SSID must have no spaces between it. As shown in Fig. 9, these are the positions of the RPs on the map.

TABLE III. NUMBER OF DISTINCT RP AND RSS DATA

Channels	Number of Distinct[a] Reference Point	Number of RSS Data (K[b])
2412	181	14.55
2417	191	12.99
2422	252	6.39
2427	256	9.46
2432	166	4.91
2437	169	7.44
2442	177	7.48
2447	214	4.49
2452	253	6.03
2457	223	8.29
2462	190	20.06

a. Unique, non-duplicate Reference Point
b. Kilo refers to a "thousand"

After gathering the data, the data proceeds with the preprocessing. Fig. 10 presents the impact of the number of reference points on the total RSS data. It is also clear that the total raw RSS data increases as the RP increases, which also increases the preprocessed data. Despite the thousands of RSS data, preprocessing reduced the amount of duplicate data.

Preprocessing further improved the data by using Cartesian Product Data Combination, which combines RSS data from many channels in groups of three while considering the existence of the various channels shown in Table III. Furthermore, as the researcher utilized on using the mobile hotspot of the user, unlike a router that emits a specific frequency band, a smartphone emits different sets of frequencies specifically: 2412, 2417, 2422, 2427, 2432, 2437, 2442, 2447, 2452, 2457, and 2462 as shown in Table III.

Also, Fig. 11 illustrates the heatmap from the three APs from different channels used in the study. It depicts that various frequencies provide different sets of heatmaps. The red marker presents the lowest signal strength, while the blue marker presents the highest signal strength.

Also, the proponent observes that different channels do not contain RSS data that covers all of the RP, as depicted on the heatmap. Even with channel 2427 having the highest number of distinct RPs shown in Table V, many RSS data from various RPs still need to be covered. This means that choosing one channel negatively affects the accuracy. In addition, choosing one specific channel is a disadvantage because scanning for a specific channel requires more time, which is unfavourable for emergency-based scenarios.

D. *Long-Term Signal Analysis*

Subsequently, this study solely focuses on the following brands of smartphones: Realme 6 Pro, Redmi Note 11, Tecno Camon 20, Vivo Y21T, Xiaomi 10T, and Xiaomi Poco X3. In addition, the proponents simultaneously measured the signal strength behaviour of different smartphones for an hour from each room covered in this study. This long-term signal analysis presents cover sets of RSSI with a range unique to their hardware, as shown in Fig. 12, which shows a discrepancy between other devices used in the study.

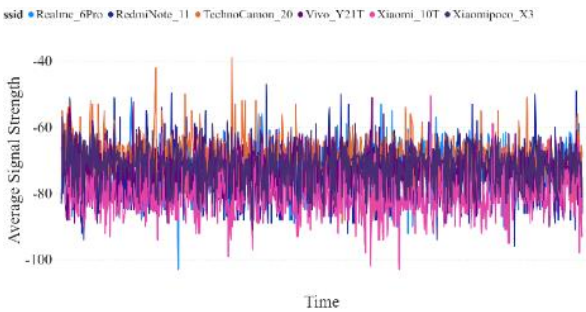


Fig. 12 Long-term signal strength analysis and the average signal strength from the six smartphones

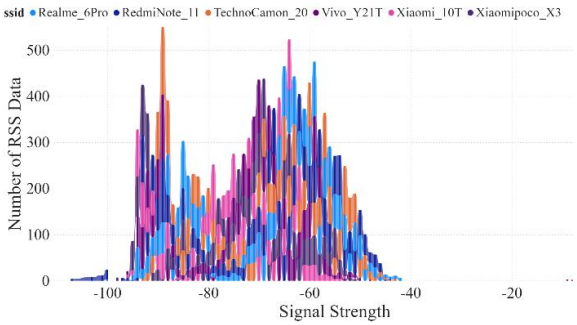


Fig. 13 Signal Strength Count from Different Smartphones

It is also important to note that different smartphones emit various RSS data, as shown in Fig. 13. To aid the problem arising from the RSS reading discrepancies among smartphones. Fig. 14 presents the calibration coefficients using different devices' multivariate linear regression analysis. Additionally, Fig. 14 shows that only some devices use every channel indicated in the report.

As shown in Fig. 11, the total amount of raw RSS is directly proportional to the number of RPs. Moreover, researchers also found a direct relationship between the number of RPs and its run time. Therefore, as the RSS data increases, the run time also increases.

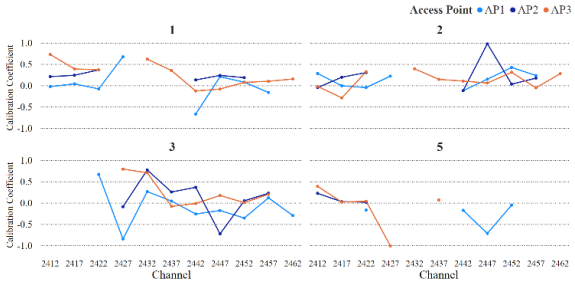
E. Statistical Analysis

The performance overview of the various position algorithms is displayed in Table IV. It demonstrates that, at 0.01 seconds, the Binary Relevance and Classifier chain run the fastest, but both algorithms have the lowest accuracy. Furthermore, with a run duration of only 0.2 seconds, the KNN algorithm has the best accuracy of 99.93%.

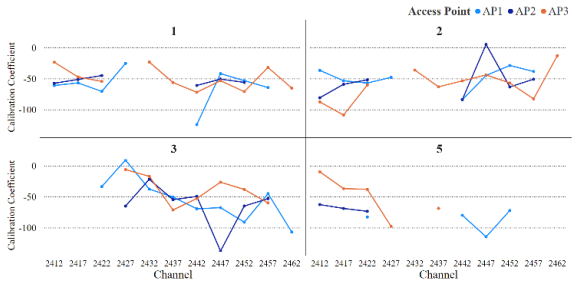
TABLE IV. RELATIVE AND POSITION ERROR OF THE WIFI-RSSI IPS

Device ID	Relative Error		Position Estimation Error (m)
	x	y	
1	0.00252871	0.00310933	4.92
2	0.00265865	0.00213866	4.11
3	0.0022728	0.00351867	5.02
4	0.00246864	0.00281562	4.78
5	0.00205952	0.00315867	4.74
6	0.00280708	0.00368667	5.71
μ	**0.002466**	**0.003071**	**4.88**

Proponents surveyed the position corresponding to the map coordinates from the Leaflet user interface. Table VI presents the relative and position errors of the location of smartphones during the online phase with the application of crowdsourcing. The x and y represent the average relative latitude and longitude error, respectively. Table VII shows the SMART-TRACE system's overall accuracy is 97.22% after 180 trials using a confusion matrix. Lastly, there is a 100% accuracy success rate for detecting six simultaneous devices.



(a)



(b)

Fig. 14 Calibration Coefficients from device ID 1, 2, 3, and 5 (a) a_s (b) b_s

TABLE V. CONFUSION MATRIX OF THE WIFI-RSSI IPS

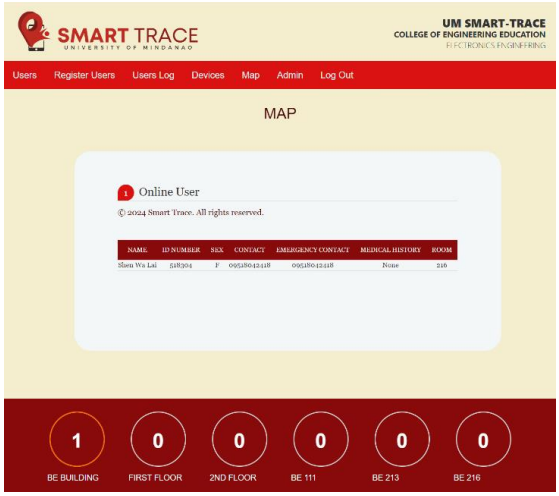
		Predicted Result		
		BE 213	***BE 216***	***BE 111***
Actual Result	***BE 213***	**57**	3	0
	BE 216	1	**59**	0
	BE 111	0	1	**59**
TRUTH OVERALL		58	63	59
ACCURACY		98.28%	93.65%	100%
TOTAL NO. OF SAMPLES		180		
OVERALL ACCURACY		**97.22%**		

TABLE VI. CONFUSION MATRIX OF SIX SIMULTANEOUS DEVICES

		Predicted Result						
		S1	***S2***	***S3***	***S4***	***S5***	***S6***	***S7***
Actual Result	***S1***	**2**	0	0	0	0	0	0
	S2	0	**2**	0	0	0	0	0
	S3	0	0	**2**	0	0	0	0
	S4	0	0	0	**2**	0	0	0
	S5	0	0	0	0	**2**	0	0
	S6	0	0	0	0	0	**2**	0
	S7	0	0	0	0	0	0	**2**
TRUTH OVERALL		2	2	2	2	2	2	2
ACCURACY		100%	100%	100%	100%	100%	100%	100%
TOTAL NO. OF SAMPLES		14						
OVERALL ACCURACY		**100%**						

F. Graphical User Interface

SMART-TRACE system's Graphical User Interface (GUI), as shown in Fig. 15, presents the integration of the RFID database and the WiFi-RSSI Indoor Positioning System (IPS).



(a)

(b)

Fig. 15 Graphical User Interface of SMART-TRACE System: (a) Online User, (b) Map

IV. Conclusions and Future Works

SMART-TRACE has achieved its objectives by integrating RFID and WiFi-RSSI fingerprinting-based IPS technology. This study successfully evaluated SMART-TRACE's capability to identify possible victims' smartphone locations through Leaflet's interactive map. This study also observed the signal strength behaviour of different smartphones through long-term signal analysis. Data shows that smartphones occupy a range of frequencies that differ from the device manufacturer's. Despite differing frequency ranges and only having two reference devices for fingerprinting-based techniques for RSS gathering, crowdsourcing minimizes the error among other devices and expedites data gathering for the offline training database. Also, position algorithms defeated the challenges of the existence of several frequency bands, such as KNN and MLKNN algorithms, with an accuracy of 99.93% and 99.92%, respectively, due to their ability to classify these channels. Also, KNN has the fastest computational calculation time of 0.2 seconds, followed by MLKNN of 0.23 seconds. There is no definitive way to measure the effectiveness of integrating RFID and WiFi-RSS IPS. However, both technologies provide excellent ways of tracking users. With RFID's accuracy of 100% with a mean time reading interval of 0.923 for its first scan and WiFi-RSS IPS' accuracy of 97.22% with an average position error of 4.88m, SMART-TRACE can assist in predicting locations during sudden destructive events.

This study covers only three rooms, and the proponents suggested a broader scope for future studies. Proponents suggest internet-based buildings because they have already established access points, which would be advantageous in locating users. Also, high-end routers would provide a broader range, increasing the study's scope. It is also recommended that other sensors be integrated, particularly in areas with poor WiFi coverage. Disaster management could be improved by providing real-time emergency exit routes on a map for the end users using a mobile application.

References

[1] J. Rutvik, “Indoor Positioning System,” International Journal of Applied Engineering Research, vol. 15, no. 12, pp. 1142–1146, Dec. 2020, doi: https://doi.org/10.37622/ijaer/15.12.2020.1142-1146.

[2] N. Syazwani C. J, A. Wahab, Noorhazirah Sunar, H. Syed, Keng Yinn Wong, and Y. Aun, “Indoor Positioning System: A Review,” International Journal of Advanced Computer Science and Applications, vol. 13, no. 6, Jan. 2022, doi: https://doi.org/10.14569/ijacsa.2022.0130659.

[3] K. Staniec, M. Kowal, S. Kubal, and P. Piotrowski, “TrackMe—A Hybrid Radio-Optical System for Assets Localization in Industry 4.0 Plants,” vol. 69, no. 2, pp. navi.524–navi.524, Jan. 2022, doi: https://doi.org/10.33012/navi.524.

[4] G. Park et al., “An Adaptive Parameter Setting Algorithm to Enhance Performance in Self-Organizing Bluetooth Low Energy Networks,” Wireless personal communications, vol. 87, no. 3, pp. 953–969, May 2015, doi: https://doi.org/10.1007/s11277-015-2619-4.

[5] C. S. Álvarez-Merino, E. J. Khatib, H. Q. Luo-Chen, and R. Barco, “Victim Detection and Localization in Emergencies,” Sensors, vol. 22, no. 21, p. 8433, Jan. 2022, doi: https://doi.org/10.3390/s22218433.

[6] S. Sadowski, P. Spachos, and K. N. Plataniotis, “Memoryless Techniques and Wireless Technologies for Indoor Localization with the Internet of Things,” IEEE Internet of Things Journal, pp. 1–1, 2020, doi: https://doi.org/10.1109/jiot.2020.2992651.

[7] J. Wang and J. Park, “An Enhanced Indoor Positioning Algorithm Based on Fingerprint Using Fine-Grained CSI and RSSI Measurements of IEEE 802.11n WLAN,” Sensors, vol. 21, no. 8, p. 2769, Apr. 2021, doi: https://doi.org/10.3390/s21082769.

[8] G. Wang, A. M.-C. So, and Y. Li, “Robust Convex Approximation Methods for TDOA-Based Localization Under NLOS Conditions,” IEEE Transactions on Signal Processing, vol. 64, no. 13, pp. 3281–3296, Jul. 2016, doi: https://doi.org/10.1109/tsp.2016.2539139.

[9] J. Tiemann and C. Wietfeld, “Scalable and precise multi-UAV indoor navigation using TDOA-based UWB localization,” International Conference on Indoor Positioning and Indoor Navigation, Sep. 2017, doi: https://doi.org/10.1109/ipin.2017.8115937.

[10] Y. Ma, W. Bobo, S. Pei, Y. Zhang, S. Zhang, and J. Yu, “An Indoor Localization Method Based on AOA and PDOA Using Virtual Stations in Multipath and NLOS Environments for Passive UHF RFID,” vol. 6, pp. 31772–31782, Jan. 2018, doi: https://doi.org/10.1109/access.2018.2838590.

[11] S. Tomažič and I. Škrjanc, "An Automated Indoor Localization System for Online Bluetooth Signal Strength Modeling Using Visual-Inertial SLAM," Sensors, vol. 21, no. 8, p. 2857, Apr. 2021, doi: https://doi.org/10.3390/s21082857.

[12] A. Belmonte-Hernandez, G. Hernandez-Penaloza, D. Martin Gutierrez, and F. Alvarez, "SWiBluX: Multi-Sensor Deep Learning Fingerprint for Precise Real-Time Indoor Tracking," IEEE Sensors Journal, vol. 19, no. 9, pp. 3473–3486, May 2019, doi: https://doi.org/10.1109/jsen.2019.2892590.

[13] Mir Yasir Umair, Kopparapu Venkata Ramana, and None Yang Dongkai, "An enhanced K-Nearest Neighbor algorithm for indoor positioning systems in a WLAN," Oct. 2014, doi: https://doi.org/10.1109/comcomap.2014.7017163.

[14] H.-H. Liu, "The Quick Radio Fingerprint Collection Method for a WiFi-Based Indoor Positioning System," Mobile Networks and Applications, vol. 22, no. 1, pp. 61–71, Dec. 2015, doi: https://doi.org/10.1007/s11036-015-0666-4.

[15] D. Yu and C. Li, "An Accurate WiFi Indoor Positioning Algorithm for Complex Pedestrian Environments," IEEE Sensors Journal, vol. 21, no. 21, pp. 24440–24452, Nov. 2021, doi: https://doi.org/10.1109/jsen.2021.3113376.

[16] A. Al-Habashna, G. Wainer, and M. Aloqaily, "Machine Learning-based Indoor Localization and Occupancy Estimation using 5G Ultra-Dense Networks," Simulation Modelling Practice and Theory, p. 102543, Mar. 2022, doi: https://doi.org/10.1016/j.simpat.2022.102543.

[17] C. Katba, "Predicting Indoor Location Using WiFi Fingerprinting – Caroline Katba," Mar. 27, 2019. https://katba-caroline.com/predicting-indoor-location-using-wifi-fingerprinting/ (accessed Jun. 14, 2024).

[18] T. Shen, S. Wang, X. Jiang, G. Peng, and Xianguo Tuo, "An Anisotropic Velocity Model for Microseismic Events Localization in Tunnels," Sensors, vol. 23, no. 10, pp. 4670–4670, May 2023, doi: https://doi.org/10.3390/s23104670.

[19] J. Huang, S. Junginger, H. Liu, and K. Thurow, "Indoor Positioning Systems of Mobile Robots: A Review," Robotics, vol. 12, no. 2, p. 47, Mar. 2023, doi: https://doi.org/10.3390/robotics12020047.

[20] "What Is a Hotspot?," Intel, 2019. https://www.intel.com/content/www/us/en /tech-tips-and-tricks/what-is-a-hotspot.html

[21] G. Retscher, "Fundamental Concepts and Evolution of Wi-Fi User Localization: An Overview Based on Different Case Studies," Sensors, vol. 20, no. 18, p. 5121, Sep. 2020, doi: https://doi.org/10.3390/s20185121.

APPENDIX A

Trade-Off Analysis

TRADE-OFF ANALYSIS

The proponents performed a trade-off analysis to assess and compare the suggested designs. A statistical method called trade-off analysis looks at the advantages and disadvantages of each design by analyzing the computed values. It also gave the researcher the ability to select the best design while considering several variables. The analysis's conclusions direct the choices made by the researchers.

Researchers' Constraints

In order to obtain the best results possible, the following parameters decided by the researchers serve as the study's design.

Accuracy. Disaster management is significant in the proposed design as it rescues potential casualties. Hence, this metric is vital for testing and statistical analysis of the study.

Convenient. This parameter would answer if the proposed design is easy to implement. Also, it entails the simplicity of the proposed design in giving a solution to a problem.

Ideal installation. The researcher considered the scalability and sustainability of the proposed design, which can account for future innovations regarding installation processes.

Economical Value. Efficiency and cost-effectiveness were critical study criteria. The proponents implemented tactics to maximize resource allocation and save costs while maintaining quality standards, as mentioned in SDG 8 regarding economic growth.

Environment. The environmental impact of the suggested design is another topic the researcher focuses on, as mentioned in SDG 15, Life on Land.

Care of Maintenance. Understanding the importance of sustainability and long-term usability, the researcher included elements that made maintenance more manageable.

Manufacturability. As mentioned in SDG 12 regarding responsible consumption, it is vital to enhance the manufacturability of a product. It can result in more environmentally friendly production methods through waste reduction, resource consumption, and environmental effects.

Social Value. Studies need to have a positive impact on society. Thus, proponents of the study regard social value as an important factor in contributing to the safety of its people and the community.

Proposed Design

The proponents proposed two designs methods for the SMART-TRACE system. The benefits and drawbacks of both systems will be weighed using a trade-off analysis using the Pugh matrix.

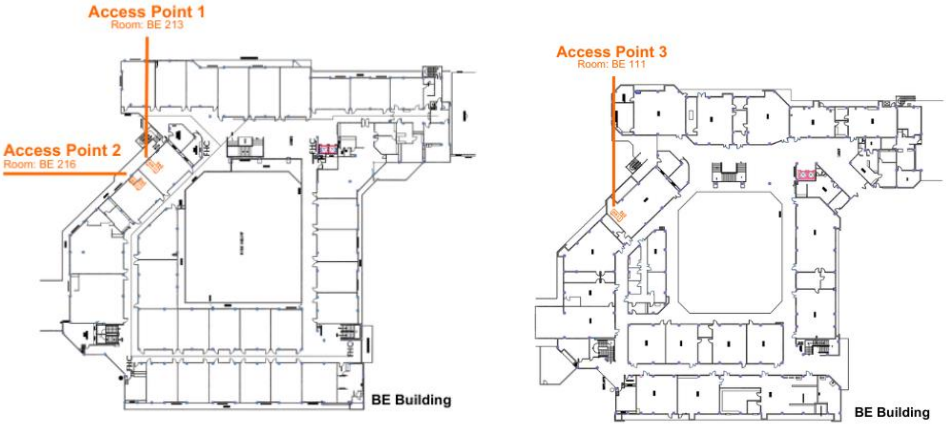


Fig. 1.1 Location of the Access Point inside the BE Building: (a) Second Floor, (b) First Floor

The setup for fingerprinting and propagation models is the same as shown in the figure. However, the parameters and factors used in the study will determine the most suitable design for the WiFi-RSSI indoor positioning, as shown in Table I.

Results show that Design 1 is superior to Design 2. With a score of 4, Design 1 shows that its accuracy, convenience, and economic prowess exceeds Design 2, which only has a score of 1. Following the findings of the trade-off analysis, the researchers shall develop a WiFi-RSSI using Design 1 as its research design. The researchers shall use three APs through a fingerprinting-based method for indoor positioning.

TABLE I. TRADEOFF ANALAYSIS

Selection Criteria	**Design 1 Fingerprinting**	**Design 2 Propagation Model**	**Total**
Accuracy	+	-	0
Convenient	+	0	1
Ideal installation	+	+	2
Economical	+	-	0
Environment	+	+	2
Care of Maintenance	-	+	0
Manufacturability	0	0	0
Total	4	1	
Rank	1st	2nd	

APPENDIX B

Data Sheets

Personal Computer

Parts	Model	Specification
Processor	Intel Core i7-13700HX	Total Cores 16 Number of Performance-cores: 8 Number of Efficient cores: 8 Total Threads: 24 Max Turbo Frequency: 5.00 GHz
Graphics	GeForce® RTX™ 4060	Memory Size: 8 GB Memory Type: GDDR6 Memory Bus: 128 bits Bandwidth: 272.0 GB/s
Memory	16 GB DDR5	Capacities: 16gb Speeds: 5600MT/s Latencies: CL40 Voltage: 1.1V, 1.25V, 1.35V Operating Temperature: 0 °C to 85 °C
SSID	512 GB PCIe NVMe M.2	Capacity: 512GB Form Factor: M.2 Write Host Performance Data Transfer (ATTO): Up to 2100 MB/s Write IOPS Random(4K) QD = 32(Max): 100K Non-Operating Temperature: -40°C to 85°C
Power Supply	3-pin 330 W 5.5 PHY AC adapter	AC Input Voltage: 100-240V 50 60Hz DC Output: 19.5V 16.92A (Compatible with 16.9A 14.36A 11.8A) Output Power: 330W (Compatible with 280W 230W)

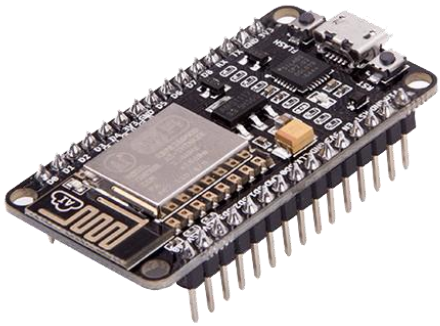
UHF RFID Reader

Frequency	860-960 MHz
Protocol	ISO 18000-6C
Adjustable Range of Power	0.1W – 1W
Antenna Gain	12dbi
Antenna	Built-in Circular Polarization Antenna
RFID UHF Reading Distance	Up to 10 meters
Voltage	12VDC ± 10%
Communication Port	USB: Support USB Communication, USB Upgrade Program RS-232, RS-485 Wigan 26 or Wigan 34
Working Mode	Software Sets Frequency Hopping or Constant Frequency
Communication Rate	Software Setting, Default 9600, Up to 115200bps
Application	Vehicle Access Control, Automatic Charging Without Stopping, Personnel Access Control Management, Logistics Monitoring, Production Automation Management and Other Fields Optional Pug: US, EU, UK, AU

MIKROTIK RouterBOARD hAP lite (RB941-2nD-TC)

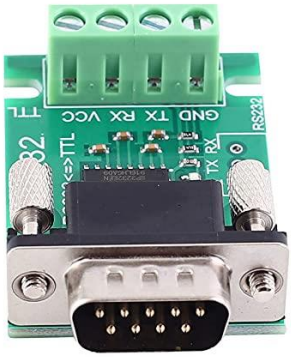
Product code	RB941-2nD-TC
Architecture	SMIPS
CPU	QCA9533
CPU core count	1
CPU nominal frequency	650 MHz
Switch chip model	QCA9533-BL3A
Operating System	RouterOS
Wireless 2.4 GHz Max data rate	300 Mbit/s
Wireless 2.4 GHz standards	802.11b/g/n
Wireless 2.4 GHz generation	Wi-Fi 4
10/100 Ethernet ports	4
MicroUSB input Voltage	5v
Max power consumption	3.5 W

ESP8266 Wireless Module Based



Microcontroller	ESP-8266 32-bit
NodeMCU Model	Clone LoLin
Clock Speed	80 MHz
USB to Serial	CH340G
USB Connector	Micro USB
Operating Voltage	3.3V
Input Voltage	4.5V-10V
WiFi Built-In	802.11 b/g/n
Temperature Range	-40C - 125C
Digital I/O Pins	11
Analog In Pins	1

MAX3232 (RS232 to TTL to UART Serial Module)



Product Name	RS232 to TTL to UART Serial Module
Product Chip	SP3232EEN
Working Voltage	DC 3V~5.5V
Work Temperature	-25℃~85℃
Package	1pcs RS232 to TTL to UART Serial Module 2pcs DB9 Male Screw
Feature	Pure data signal conversion. The module is bidirectional interchangeable conversion. It can directly receive the COM port of the computer and lead out the RS232 response signal.

APPENDIX C

Data Gathering and Analysis

TABLE II. TIME READING OF UHF RFID READER

Trials	Time in				
	1	2	3	4	5
1	16:48:00.137	16:48:00.831	16:48:01.223	16:48:02.571	16:48:03.731
2	16:49:13.431	16:49:14.141	16:49:14.535	16:49:17.810	16:49:19.777
3	16:50:06.355	16:50:07.593	16:50:10.446	16:50:14.970	16:50:18.225
4	16:51:03.833	16:51:04.111	16:51:06.277	16:51:07.283	16:51:10.788
5	16:51:52.571	16:51:53.268	16:51:55.418	16:51:55.725	16:51:59.271
6	16:52:36.671	16:52:37.695	16:52:40.870	16:52:45.200	16:52:45.791
7	16:53:29.510	16:53:30.746	16:53:31.142	16:53:33.418	16:53:37.946
8	16:54:22.076	16:54:22.477	16:54:23.593	16:54:27.183	16:54:30.435
9	16:55:14.057	16:55:14.757	16:55:16.313	16:55:17.353	16:55:18.525
10	16:55:53.745	16:55:54.216	16:55:56.649	16:56:00.291	16:56:01.185
11	16:56:31.931	16:56:31.723	16:56:32.727	16:56:34.875	16:56:39.530
12	16:57:36.208	16:57:38.150	16:57:40.128	16:57:43.793	16:57:47.288
13	16:58:18.725	16:58:20.642	16:58:22.191	16:58:23.148	16:58:27.962
14	16:59:48.712	16:59:49.537	16:59:51.380	16:59:52.705	16:59:55.775
15	17:00:26.316	17:00:27.207	17:00:28.595	17:00:29.069	17:00:33.075
16	17:01:24.790	17:01:26.194	17:01:28.725	17:01:31.096	17:01:34.290
17	17:02:23.580	17:02:24.576	17:02:25.395	17:02:26.917	17:02:30.227
18	17:03:04.707	17:03:05.315	17:03:07.251	17:03:8.485	17:03:11.238
19	17:03:45.722	17:03:46.176	17:03:46.905	17:03:50.272	17:03:54.270
20	17:04:29.061	17:04:29.802	17:04:30.195	17:04:30.621	17:04:34.783
21	17:05:09.548	17:05:11.157	17:05:12.279	17:05:14.146	17:05:16.965
22	17:05:49.266	17:05:49.574	17:05:50.697	17:05:51.209	17:05:04.174
23	17:06:31.326	17:06:32.154	17:06:33.887	17:06:35.959	17:06:37.307
24	17:07:14.517	17:07:14.853	17:07:16.915	17:07:19.576	17:07:22.649
25	17:07:55.215	17:07:56.343	17:07:56.758	17:07:58.196	17:07:59.095
26	17:08:34.331	17:08:35.162	17:08:35.753	17:08:36.357	17:08:39.476
27	17:09:28.818	17:09:29.626	17:09:30.432	17:09:31.272	17:09:34.526
28	17:10:10.096	17:10:10.795	17:10:12.015	17:10:13.159	17:10:16.266
29	17:10:53.096	17:10:53.688	17:10:57.279	17:10:59.120	17:10:59.851
30	17:11:37.000	17:11:37.535	17:11:39.065	17:11:40.800	17:11:41.564

TABLE III. TIME INTERVAL READING OF THE UHF RFID READER

Trials	Time Interval (s)			
	1	2	3	4
1	0.694	0.392	1.348	1.16
2	0.71	0.394	3.275	1.967
3	1.238	2.853	4.524	3.255
4	1.278	2.166	1.006	3.505
5	0.697	2.15	0.307	3.546
6	1.024	3.175	4.33	0.591
7	1.236	0.396	2.276	3.528
8	0.401	1.116	3.59	3.252
9	0.7	1.556	1.04	1.172
10	0.471	6.433	3.642	0.894
11	0.792	1.004	2.148	4.655
12	1.942	1.978	3.665	3.495
13	1.917	1.549	0.957	3.814
14	0.825	1.843	1.325	3.07
15	0.891	1.388	0.474	3.006
16	1.404	2.531	2.371	3.194
17	1.996	0.819	1.522	3.31
18	0.608	1.936	5.234	2.753
19	0.454	0.729	3.367	3.998
20	0.741	0.393	0.426	3.162
21	1.609	1.122	1.867	2.819
22	0.308	1.123	0.512	3.965
23	0.828	1.733	2.072	1.348
24	0.336	2.062	2.661	3.073
25	1.128	0.415	1.438	0.899
26	0.831	0.591	0.604	3.119
27	0.808	0.806	0.84	3.254
28	0.699	1.22	1.144	3.107
29	0.592	3.591	1.841	0.731
30	0.535	1.53	1.735	0.764
μ	**0.923**	**1.500**	**1.918**	**2.680**
σ	**0.308**	**0.392**	**0.307**	**0.591**
Max	**1.996**	**3.591**	**4.524**	**4.655**
Min	**0.308**	**0.392**	**0.307**	**0.591**

TABLE IV. COORDINATE LOCATION OF DEVICE 1 IN THREE DIFFERENT ROOMS

Device 1						
Trials	**Theoretical**		**Predicted**		**Relative Error**	**Actual**
	Latitude (x_a)	***Longitude (y_a)***	***Latitude (x_p)***	***Longitude (y_p)***	***x***	***y***
1	7.065809636	125.596406	7.065825778	125.5964129	0.0016142	0.00069
2	7.065687876	125.5963854	7.065824607	125.596451	0.0136731	0.00656
3	7.065800987	125.5964283	7.065797327	125.5964389	0.000366	0.00106
4	7.0657658	125.5964142	7.065795912	125.5964147	0.0030112	5E-05
5	7.06578052	125.5964578	7.065765712	125.5963898	0.0014808	0.0068
6	7.065735102	125.5964779	7.065820947	125.596462	0.0085845	0.00159
7	7.065768042	125.5964915	7.065818455	125.5964094	0.0050413	0.00821
8	7.065824607	125.596451	7.065825778	125.5964129	0.0001171	0.00381
9	7.065766212	125.5964396	7.065797327	125.5964389	0.0031115	7E-05
10	7.065813792	125.5964836	7.065795912	125.5964147	0.001788	0.00689
11	7.065630791	125.5964374	7.065653083	125.596408	0.0022292	0.00294
12	7.065644765	125.5964303	7.065641272	125.5964413	0.0003493	0.0011
13	7.065648425	125.5964198	7.065668389	125.5964018	0.0019964	0.0018
14	7.065664396	125.5964129	7.065676536	125.5964555	0.001214	0.00426
15	7.065683194	125.5963978	7.065702826	125.5963792	0.0019632	0.00186
16	7.065694842	125.5964022	7.065686857	125.596423	0.0007985	0.00208
17	7.065713974	125.5963834	7.065722291	125.5964349	0.0008317	0.00515
18	7.06572512	125.5963878	7.065728113	125.5964156	0.0002993	0.00278
19	7.065751405	125.5963839	7.065769289	125.5963777	0.0017884	0.00062
20	7.065722291	125.5964349	7.065735102	125.5964779	0.0012811	0.0043
21	7.065833589	125.5964458	7.06582394	125.5964964	0.0009649	0.00506
22	7.065790335	125.5964833	7.06581296	125.5964122	0.0022625	0.00711
23	7.065842906	125.5964471	7.065831593	125.5964407	0.0011313	0.00064
24	7.065835919	125.5964676	7.065791998	125.5964471	0.0043921	0.00205
25	7.065813626	125.5964558	7.065790335	125.5964833	0.0023291	0.00275
26	7.065768042	125.5964877	7.065788671	125.5964786	0.0020629	0.00091
27	7.065825604	125.5964069	7.065786342	125.5964736	0.0039262	0.00667
28	7.065773365	125.5964608	7.065819615	125.5964877	0.004625	0.00269
29	7.065795326	125.5964947	7.065788005	125.5964974	0.0007321	0.00027
30	7.065811629	125.5964327	7.065792664	125.5964076	0.0018965	0.00251
Average					0.00252871	0.00310933

TABLE V. COORDINATE LOCATION OF DEVICE 2 IN THREE DIFFERENT ROOMS

Device 2						
Trials	**Theoretical**		**Predicted**		**Relative Error**	**Actual**
	Latitude (x_a)	***Longitude (y_a)***	***Latitude (x_p)***	***Longitude (y_p)***	***x***	***y***
1	7.065739094	125.5964665	7.065781517	125.5964835	0.0042423	0.0017
2	7.065750245	125.5964587	7.065825778	125.5964129	0.0075533	0.00458
3	7.065761476	125.5964515	7.065827768	125.5964389	0.0066292	0.00126
4	7.065757565	125.5964365	7.065795912	125.5964147	0.0038347	0.00218
5	7.065783685	125.5964211	7.065765712	125.5963898	0.0017973	0.00313
6	7.0658004	125.5964026	7.065823029	125.5963977	0.0022629	0.00049
7	7.065766551	125.596465	7.065818455	125.5964094	0.0051904	0.00556
8	7.065797327	125.5964389	7.065820947	125.596462	0.002362	0.00231
9	7.065801321	125.5964541	7.065782934	125.596396	0.0018387	0.00581
10	7.065769289	125.5963777	7.065824607	125.596451	0.0055318	0.00733
11	7.06574708	125.5963965	7.065736099	125.5963928	0.0010981	0.00037
12	7.065706159	125.5964057	7.065686857	125.596423	0.0019302	0.00173
13	7.065722291	125.5964349	7.065721125	125.5963998	0.0001166	0.00351
14	7.065653083	125.596408	7.065672568	125.5963904	0.0019485	0.00176
15	7.065749742	125.596433	7.065735765	125.596429	0.0013977	0.0004
16	7.065682864	125.596433	7.065713976	125.5964211	0.0031112	0.00119
17	7.065675567	125.5964183	7.065664396	125.5964129	0.0011171	0.00054
18	7.06571181	125.5964697	7.06571547	125.5964578	0.000366	0.00119
19	7.065695341	125.5964369	7.065687876	125.5963854	0.0007465	0.00515
20	7.065718465	125.5963708	7.065696671	125.5963624	0.0021794	0.00084
21	7.065821944	125.5964919	7.065815622	125.5965006	0.00063219	0.00086794
22	7.065783015	125.5965048	7.065786841	125.5965155	0.00038264	0.00107285
23	7.065772367	125.5964796	7.065790335	125.5964833	0.00179674	0.00036886
24	7.065817951	125.5964649	7.065819948	125.5964702	0.00019964	0.00053643
25	7.065764382	125.5964957	7.065779022	125.5964957	0.00146401	0
26	7.06578002	125.5964759	7.065797322	125.5964793	0.0017302	0.00033537
27	7.065760056	125.5964853	7.065821611	125.5964753	0.0061555	0.00100573
28	7.065834255	125.596427	7.065785676	125.596432	0.00485786	0.00050267
29	7.065823275	125.5964012	7.065800316	125.5964253	0.00229584	0.00240952
30	7.065790335	125.5963995	7.065840244	125.5964598	0.00499095	0.00603042
Average					0.00265865	0.00213866

TABLE VI. COORDINATE LOCATION OF DEVICE 3 IN THREE DIFFERENT ROOMS

Device 3						
Trials	**Theoretical**		**Predicted**		**Relative Error**	**Actual**
	Latitude (x_a)	***Longitude (y_a)***	***Latitude (x_p)***	***Longitude (y_p)***	***x***	***y***
1	7.065656577	125.5963693	7.065641272	125.5964413	0.0015305	0.0072
2	7.06565659	125.5964354	7.065644765	125.5964303	0.0011825	0.00051
3	7.065675567	125.5964183	7.065679536	125.5964441	0.0003969	0.00258
4	7.065682864	125.596433	7.06569168	125.5964483	0.0008816	0.00153
5	7.06571547	125.5964578	7.065740258	125.5963794	0.0024788	0.00784
6	7.065668389	125.5964018	7.06572512	125.5963878	0.0056731	0.0014
7	7.065732433	125.5964384	7.065769289	125.5963777	0.0036856	0.00607
8	7.065732433	125.5964384	7.065769289	125.5963777	0.0036856	0.00607
9	7.065652085	125.5963812	7.065677372	125.5963797	0.0025287	0.00015
10	7.065681365	125.5963697	7.065664396	125.5964129	0.0016969	0.00432
11	7.065749742	125.596433	7.065735102	125.5964779	0.001464	0.00449
12	7.065757565	125.5964365	7.065751405	125.5963839	0.000616	0.00526
13	7.065766212	125.5964396	7.065762891	125.5964756	0.0003321	0.0036
14	7.065775528	125.5964429	7.065795661	125.5963876	0.0020133	0.00553
15	7.065785265	125.5964474	7.065746917	125.5964692	0.0038348	0.00218
16	7.065793335	125.596451	7.065813797	125.596422	0.0020462	0.0029
17	7.065823029	125.5963977	7.065811304	125.5964582	0.0011725	0.00605
18	7.065786348	125.5964984	7.065814044	125.5963943	0.0027696	0.01041
19	7.065809466	125.5964947	7.065813792	125.5964836	0.0004326	0.00111
20	7.065811304	125.5964582	7.065801321	125.5964541	0.0009983	0.00041
21	7.06578368	125.5964274	7.065754321	125.596472	0.0029359	0.00446
22	7.065772367	125.5964796	7.065793329	125.5965125	0.0020962	0.00329
23	7.065834255	125.596427	7.065797655	125.5964207	0.00366	0.00063
24	7.065783015	125.5965048	7.06582602	125.5964845	0.0043005	0.00203
25	7.065775694	125.5964478	7.065811629	125.5964327	0.0035935	0.00151
26	7.065772034	125.5963942	7.065783015	125.5964441	0.0010981	0.00499
27	7.065788671	125.5964786	7.065813626	125.5964558	0.0024955	0.00228
28	7.065825937	125.5964253	7.065804975	125.5964347	0.0020962	0.00094
29	7.065840244	125.5964598	7.065827933	125.5964488	0.0012311	0.0011
30	7.065783015	125.5964029	7.065835586	125.5964501	0.0052571	0.00472
Average					0.0022728	0.00351867

TABLE VII. COORDINATE LOCATION OF DEVICE 4 IN THREE DIFFERENT ROOMS

Device 4						
Trials	**Theoretical**		**Predicted**		**Relative Error**	**Actual**
	Latitude (x_a)	***Longitude (y_a)***	***Latitude (x_p)***	***Longitude (y_p)***	***x***	***y***
1	7.0657657	125.59639	7.0657695	125.5964	0.0003747	0.00141
2	7.0658061	125.59644	7.0657919	125.5964	0.0014227	0.00434
3	7.0657863	125.5965	7.0657658	125.59641	0.0020548	0.00842
4	7.0657771	125.59647	7.0657892	125.59644	0.0012065	0.00331
5	7.0657543	125.59647	7.0658358	125.59649	0.0081431	0.00135
6	7.0657534	125.59645	7.0657504	125.59648	0.0002994	0.00356
7	7.065771	125.59643	7.0657657	125.59639	0.0005246	0.00376
8	7.0657576	125.59644	7.0658061	125.59644	0.004858	0.00059
9	7.0657536	125.59642	7.0657629	125.59648	0.0009322	0.00541
10	7.065814	125.59639	7.0658105	125.59643	0.0003574	0.00374
11	7.0656566	125.59637	7.0656484	125.59642	0.0013473	0.00305
12	7.0656391	125.59642	7.0656526	125.59645	0.0004489	0.00117
13	7.0656716	125.59643	7.0656671	125.59644	0.0006985	0.00166
14	7.0656922	125.59637	7.0656992	125.59639	0.0003808	0.00114
15	7.0656644	125.59641	7.0656606	125.59642	0.0025278	0.0003
16	7.0657324	125.59644	7.0657072	125.59644	0.0018488	0.00207
17	7.0656531	125.59641	7.0656716	125.59643	0.0006635	0.00171
18	7.0656792	125.59641	7.0656726	125.59639	0.0134592	0.0027
19	7.0657025	125.59642	7.0658371	125.59644	0.0050565	0.0006
20	7.0657072	125.59644	7.0656566	125.59644	0.00287811	0.00149198
21	7.0657868	125.59652	7.0658156	125.5965	0.00287811	0.00149198
22	7.0657807	125.5965	7.0657868	125.59652	0.00061139	0.00147523
23	7.0658316	125.59648	7.0658451	125.59649	0.00134756	0.00100586
24	7.0658199	125.59647	7.0658311	125.59649	0.00111465	0.00229675
25	7.0657953	125.59647	7.0658381	125.59649	0.00427558	0.00147531
26	7.0658402	125.59646	7.0658558	125.59648	0.00155551	0.0015758
27	7.0657777	125.59645	7.0658086	125.5965	0.00309439	0.00513024
28	7.065816	125.59644	7.065807	125.5965	0.00089837	0.00570021
29	7.0657977	125.59644	7.0658286	125.59649	0.00309439	0.00509676
30	7.0657903	125.5964	7.0658474	125.59647	0.00570632	0.00743853
Average					0.00246864	0.00281562

TABLE VIII. COORDINATE LOCATION OF DEVICE 5 IN THREE DIFFERENT ROOMS

Device 5						
Trials	**Theoretical**		**Predicted**		**Relative Error**	**Actual**
	Latitude (x_a)	***Longitude (y_a)***	***Latitude (x_p)***	***Longitude (y_p)***	***x***	***y***
1	7.065735102	125.5964779	7.065739589	125.5964194	0.0004487	0.00585
2	7.065769289	125.5963777	7.065735765	125.596429	0.0033524	0.00513
3	7.065765712	125.5963898	7.065761724	125.5964248	0.0003988	0.0035
4	7.065786966	125.5963842	7.065773158	125.5963928	0.0013808	0.00086
5	7.06577753	125.5964949	7.065790175	125.5964868	0.0012645	0.00081
6	7.065814044	125.5963943	7.065786928	125.5964115	0.0027116	0.00172
7	7.065798827	125.5964909	7.065802986	125.5964803	0.0004159	0.00106
8	7.065824607	125.596451	7.065827768	125.5964389	0.0003161	0.00121
9	7.065770958	125.5964274	7.065782934	125.596396	0.0011976	0.00314
10	7.065811304	125.5964582	7.065825778	125.5964129	0.0014474	0.00453
11	7.065728943	125.5964491	7.065742754	125.5964558	0.0013811	0.00067
12	7.06565691	125.5963968	7.065674544	125.5963546	0.0017634	0.00422
13	7.065667082	125.5964404	7.065683194	125.5963978	0.0016112	0.00426
14	7.065696671	125.5963624	7.065688021	125.5964603	0.000865	0.00979
15	7.065713976	125.5964211	7.065722374	125.5964735	0.0008398	0.00524
16	7.065722374	125.5964735	7.065702826	125.5963792	0.0019548	0.00943
17	7.065648425	125.5964198	7.065664396	125.5964129	0.0015971	0.00069
18	7.065639109	125.5964159	7.065647759	125.5963931	0.000865	0.00228
19	7.065668389	125.5964018	7.065672568	125.5963904	0.0004179	0.00114
20	7.065695341	125.5964369	7.065688021	125.5964603	0.000732	0.00234
21	7.065793001	125.5964251	7.065808635	125.5965044	0.0015634	0.00793
22	7.065847398	125.5964739	7.065849644	125.5964785	0.0002246	0.00046
23	7.065838081	125.5964893	7.06577636	125.5964893	0.0061721	0
24	7.065755065	125.5964873	7.065811629	125.5964736	0.0056564	0.00137
25	7.065796989	125.5964994	7.065771036	125.5964558	0.0025953	0.00436
26	7.065772034	125.5963942	7.065802646	125.59643	0.0030612	0.00358
27	7.065779354	125.5964347	7.065823275	125.5964012	0.0043921	0.00335
28	7.065813958	125.5964374	7.065830262	125.5964169	0.0016304	0.00205
29	7.065835586	125.5964501	7.065830262	125.5964169	0.0005324	0.00332
30	7.065858045	125.59648	7.065748078	125.5964847	0.0109967	0.00047
Average					0.00205952	0.00315867

TABLE IX. COORDINATE LOCATION OF DEVICE 6 IN THREE DIFFERENT ROOMS

Device 6						
Trials	**Theoretical**		**Predicted**		**Relative Error**	**Actual**
	Latitude (x_a)	***Longitude (y_a)***	***Latitude (x_p)***	***Longitude (y_p)***	***x***	***y***
1	7.065707152	125.5963663	7.06569085	125.5964126	0.0016302	0.00463
2	7.065707152	125.5963663	7.065769459	125.5964039	0.0062307	0.00376
3	7.065685857	125.5963586	7.065713976	125.5964211	0.0028119	0.00625
4	7.065667082	125.5964404	7.065724952	125.5964256	0.005787	0.00148
5	7.065652582	125.5964464	7.065751405	125.5963839	0.0098823	0.00625
6	7.065653083	125.596408	7.065692179	125.5963742	0.0039096	0.00338
7	7.065696671	125.5963624	7.065695341	125.5964369	0.000133	0.00745
8	7.065717298	125.5964106	7.065719132	125.5964454	0.0001834	0.00348
9	7.065724952	125.5964256	7.065743746	125.5964091	0.0018794	0.00165
10	7.065724957	125.5964619	7.065721125	125.5963998	0.0003832	0.00621
11	7.065809636	125.596406	7.065795912	125.5964147	0.0013724	0.00087
12	7.065804813	125.5964186	7.065818455	125.5964094	0.0013642	0.00092
13	7.065801321	125.5964541	7.065758398	125.5964874	0.0042923	0.00333
14	7.065798827	125.5964909	7.065769459	125.5964039	0.0029368	0.0087
15	7.065772202	125.5964798	7.065785265	125.5964474	0.0013063	0.00324
16	7.065758398	125.5964874	7.065786966	125.5963842	0.0028568	0.01032
17	7.065795661	125.5963876	7.065773158	125.5963928	0.0022503	0.00052
18	7.06577495	125.5964168	7.065765712	125.5963898	0.0009238	0.0027
19	7.065795657	125.5965024	7.065790175	125.5964868	0.0005482	0.00156
20	7.065827768	125.5964389	7.065780108	125.5964311	0.004766	0.00078
21	7.065800816	125.5965085	7.06582394	125.5964199	0.0023124	0.00886
22	7.065800816	125.5965085	7.06582394	125.5964199	0.0023124	0.00886
23	7.06584091	125.5964421	7.065815622	125.5965006	0.0025288	0.00585
24	7.065824938	125.5964612	7.065807304	125.5964401	0.0017634	0.00211
25	7.065770038	125.5964927	7.065845069	125.5964863	0.0075031	0.00064
26	7.065832259	125.5964216	7.065779354	125.5964347	0.0052905	0.00131
27	7.065790002	125.5964414	7.065788005	125.5964367	0.0001997	0.00047
28	7.065755065	125.5964575	7.065783015	125.5964441	0.002795	0.00134
29	7.065818949	125.5964095	7.065838913	125.596437	0.0019964	0.00275
30	7.065774696	125.596424	7.065795326	125.5964333	0.002063	0.00093
Average					0.00280708	0.00368667

APPENDIX D

RSS Data Summary

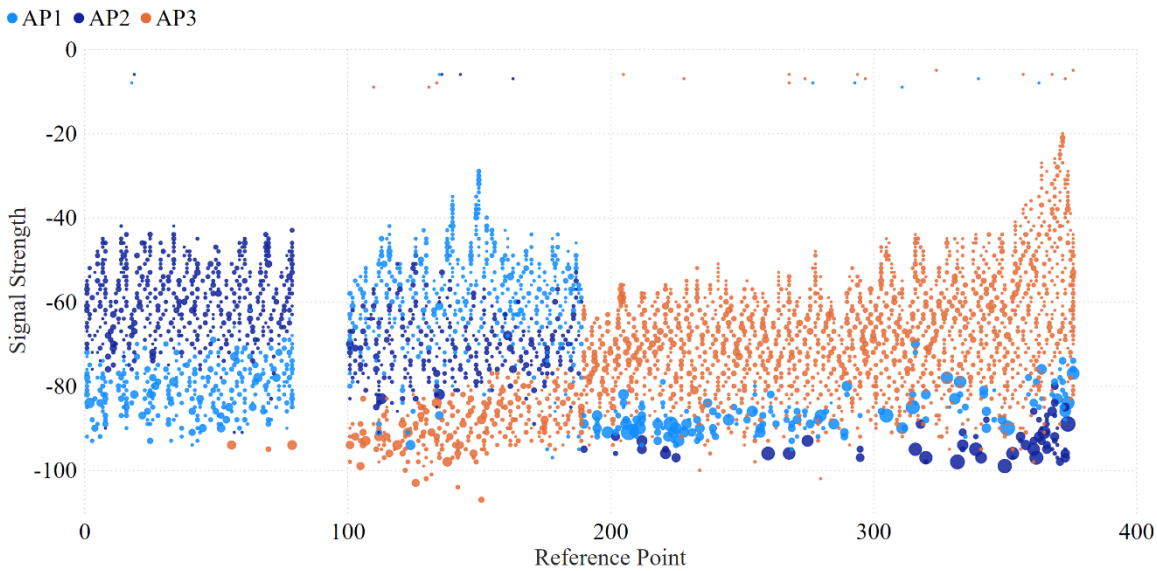


Fig. 4.1 Device 4 RSSI Reading from Three Access Points

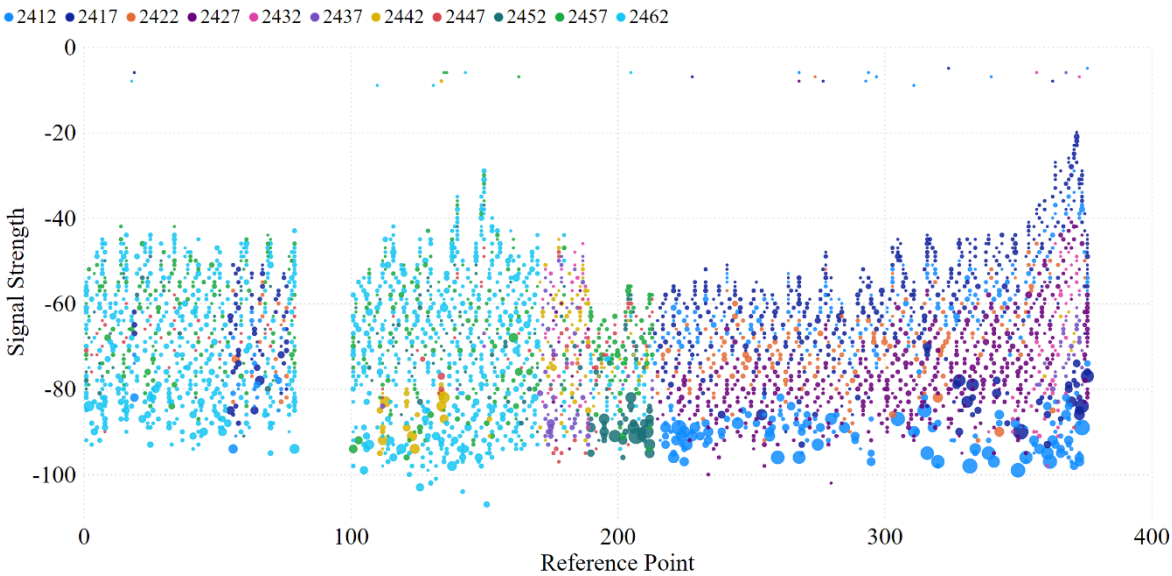


Fig. 4.2 Device 4 RSSI Reading from Different Channels

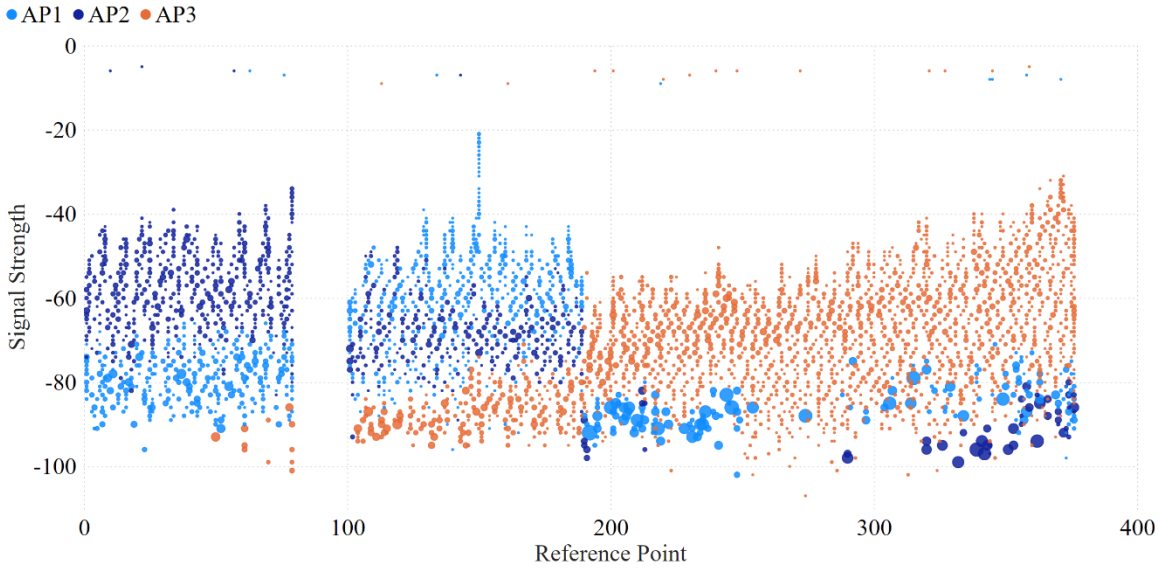


Fig. 4.3 Device 6 RSSI Reading from Three Access Points

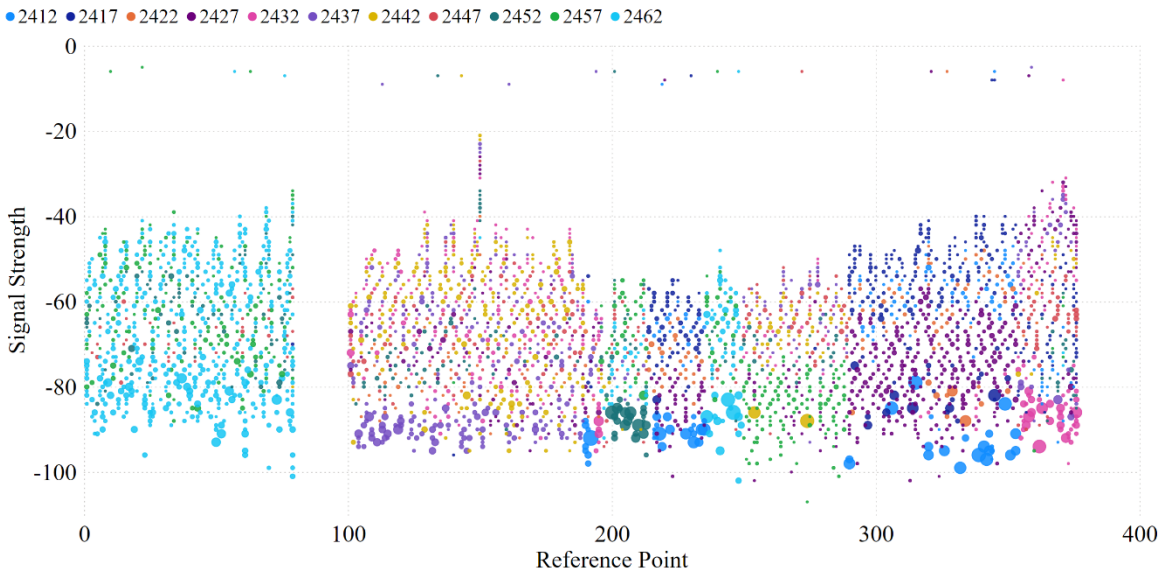


Fig. 4.4 Device 6 RSSI Reading from Different Channels

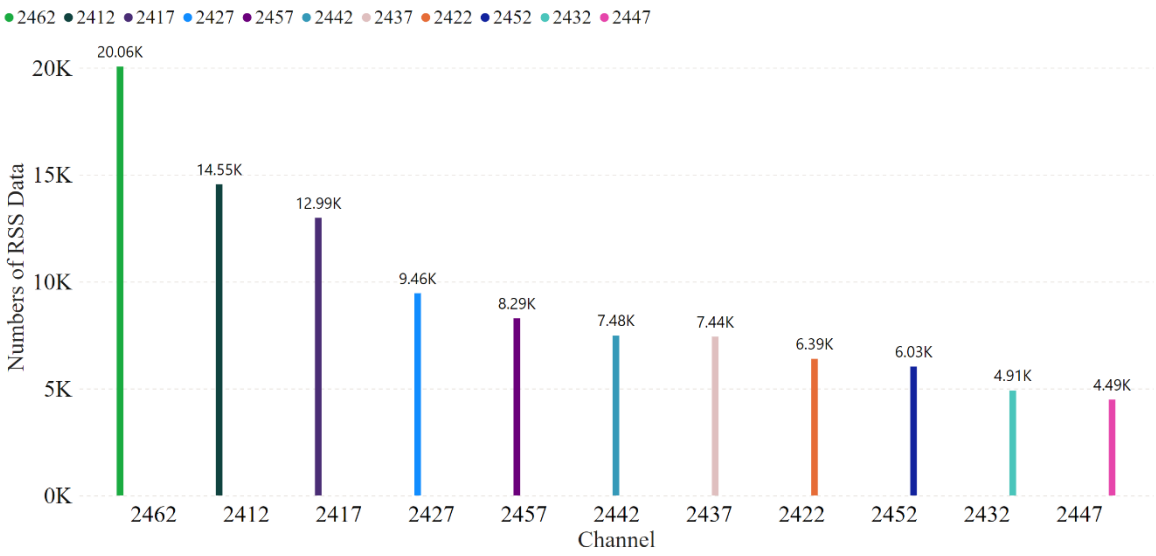


Fig. 4.5 Device 4 Number of RSS Data from Different Channels

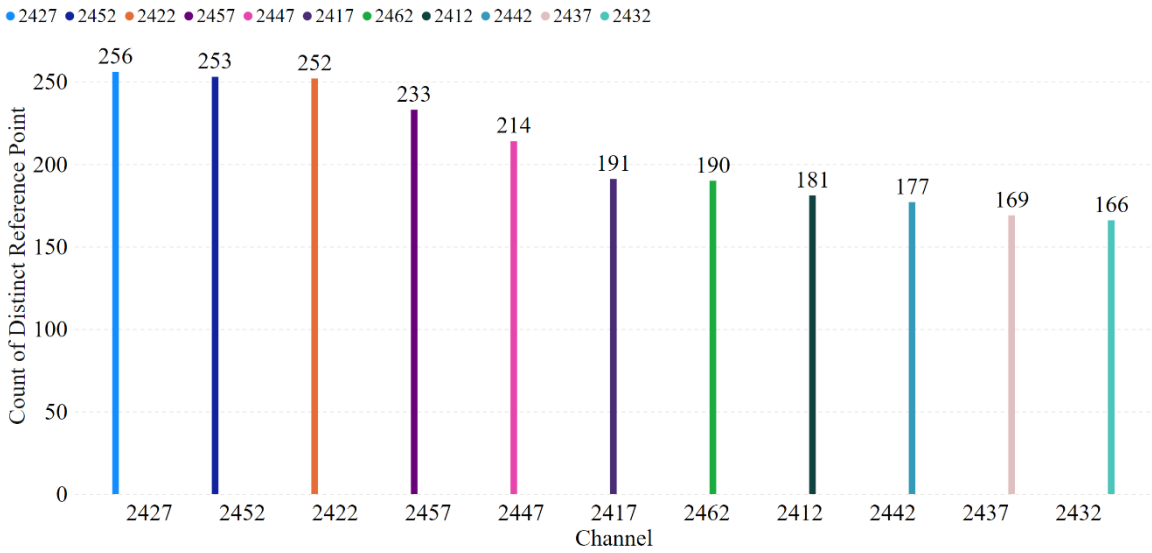


Fig. 4.6 Device 6 Number of RSS Data from Different Channels

APPENDIX E

Long-term Signal Analysis

ACCESS POINT 1

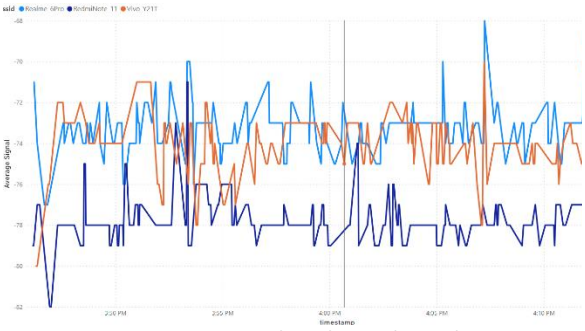


Fig. 5.1 Long-term Signal Analysis: Channel 2412

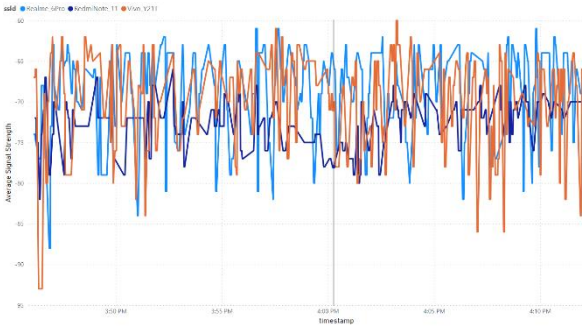


Fig. 5.2 Long-term Signal Analysis: Channel 2417

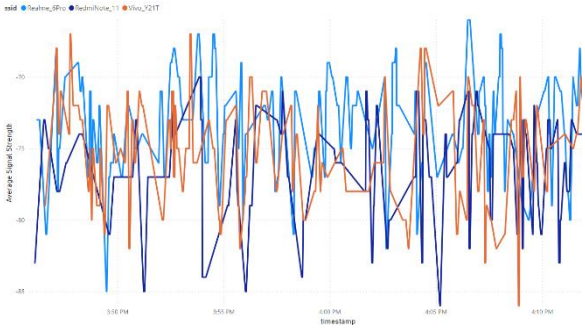


Fig. 5.3 Long-term Signal Analysis: Channel 2422

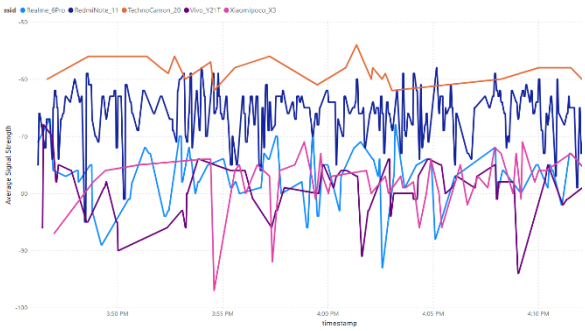


Fig. 5.4 Long-term Signal Analysis: Channel 2427

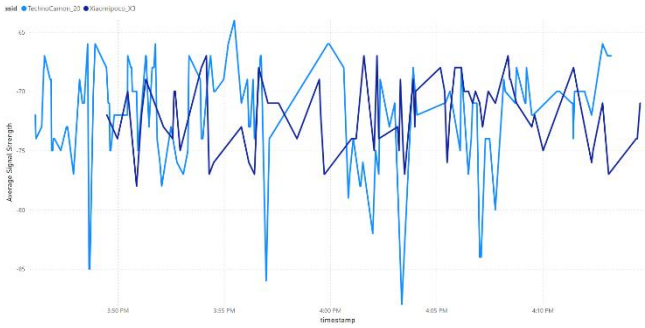


Fig. 5.5 Long-term Signal Analysis: Channel 2432

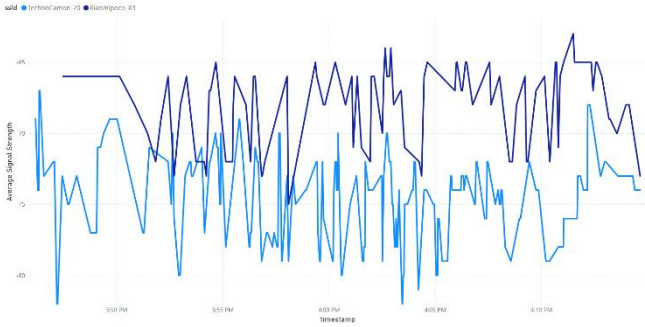


Fig. 5.6 Long-term Signal Analysis: Channel 2437

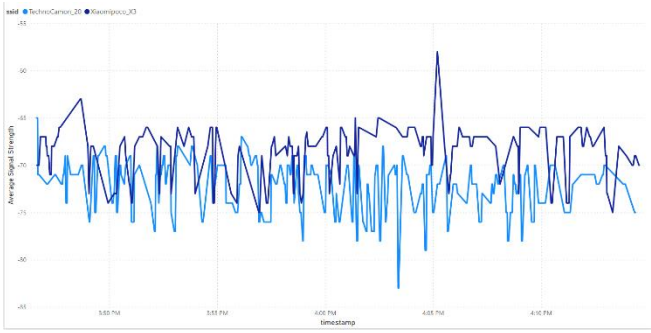


Fig. 5.7 Long-term Signal Analysis: Channel 2442

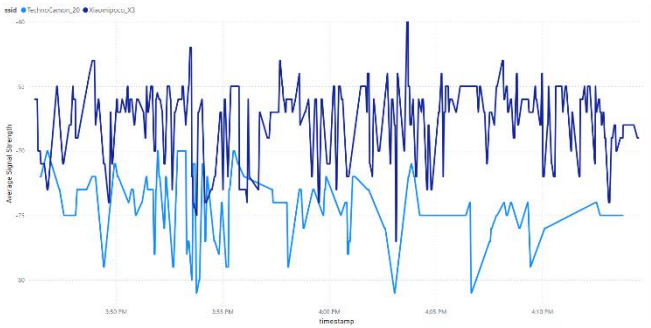


Fig. 5.8 Long-term Signal Analysis: Channel 2457

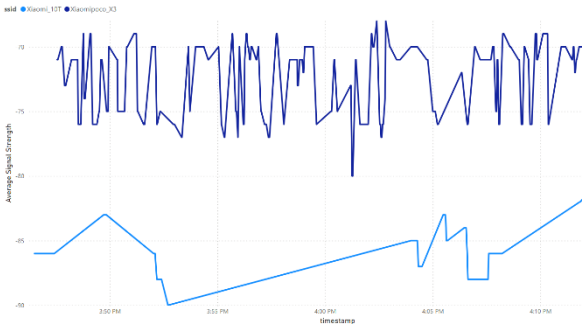


Fig. 5.9 Long-term Signal Analysis: Channel 2452

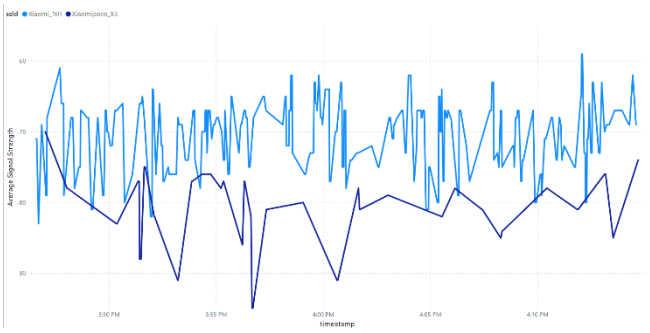


Fig. 5.10 Long-term Signal Analysis: Channel 2457

ACCESS POINT 2

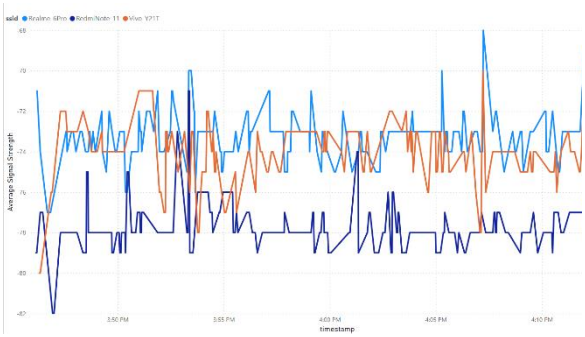


Fig. 5.11 Long-term Signal Analysis: Channel 2412

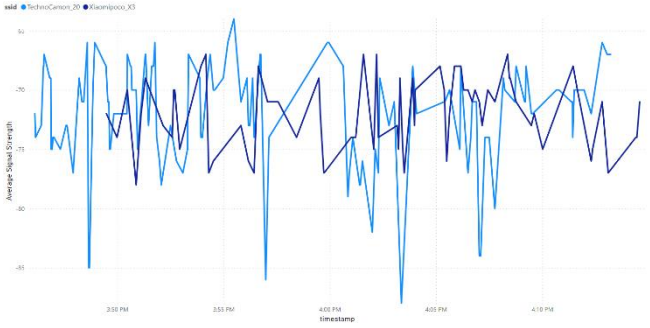


Fig. 5.14 Long-term Signal Analysis: Channel 2432

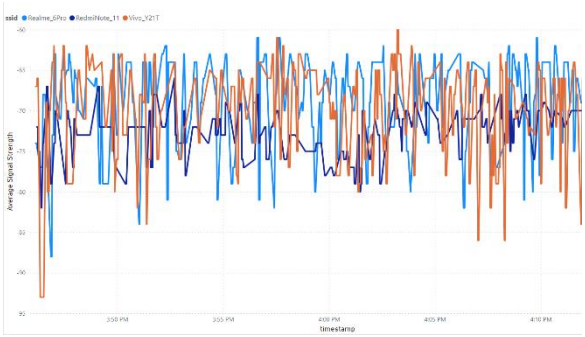


Fig. 5.12 Long-term Signal Analysis: Channel 2417

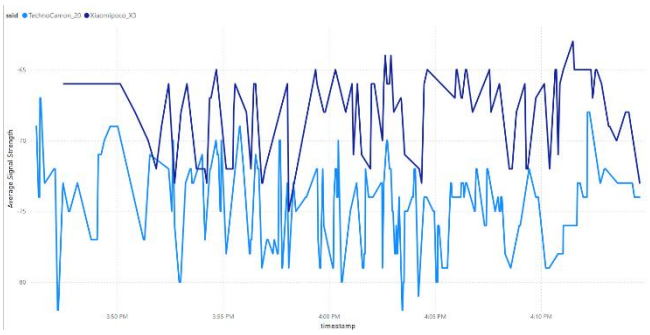


Fig. 5.15 Long-term Signal Analysis: Channel 2437

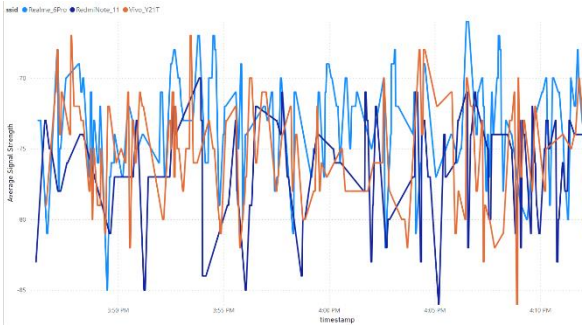


Fig. 5.13 Long-term Signal Analysis: Channel 2422

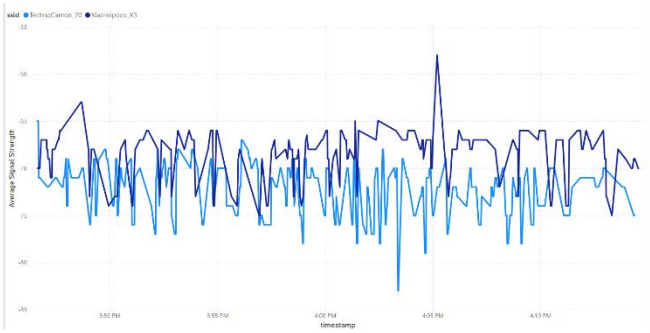


Fig. 5.16 Long-term Signal Analysis: Channel 2442

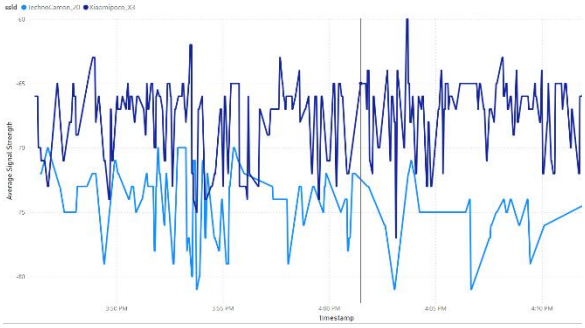


Fig. 5.17 Long-term Signal Analysis: Channel 2447

ACCESS POINT 3

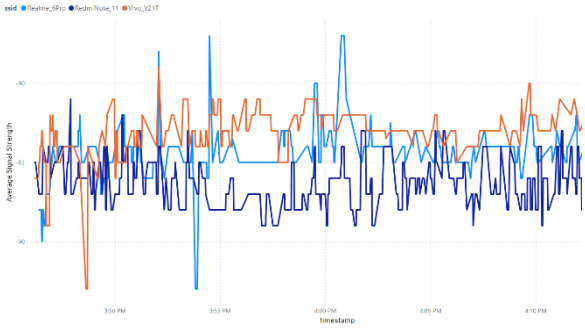


Fig. 5.18 Long-term Signal Analysis: Channel 2412

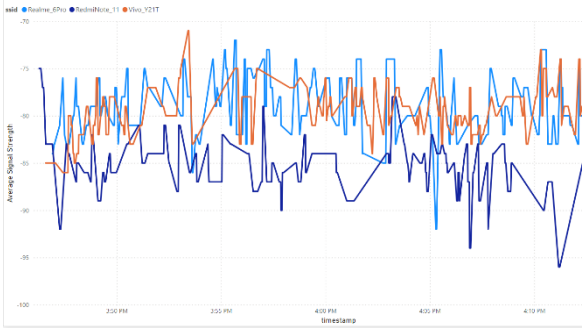


Fig. 5.19 Long-term Signal Analysis: Channel 2417

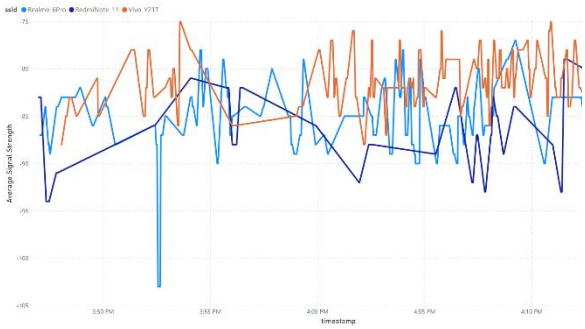


Fig. 5.20 Long-term Signal Analysis: Channel 2422

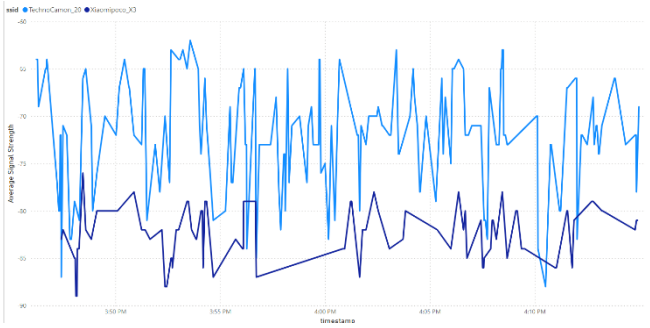


Fig. 5.21 Long-term Signal Analysis: Channel 2432

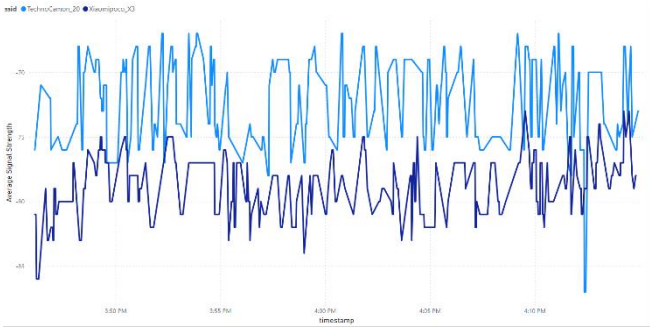


Fig. 5.22 Long-term Signal Analysis: Channel 2437

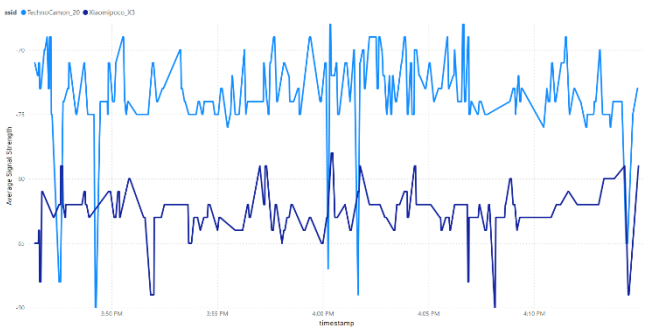


Fig. 5.23 Long-term Signal Analysis: Channel 2442

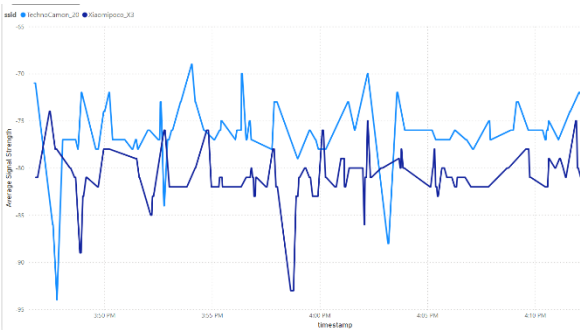


Fig. 5.24 Long-term Signal Analysis: Channel 2447

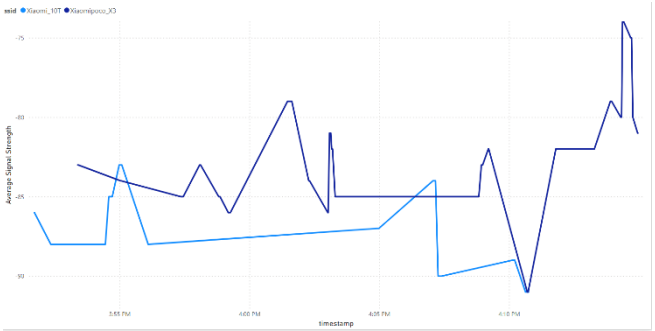


Fig. 5.25 Long-term Signal Analysis: Channel 2452

APPENDIX F

Heatmap

DEVICE 4 HEATMAP READING FROM ACCESS POINT 1

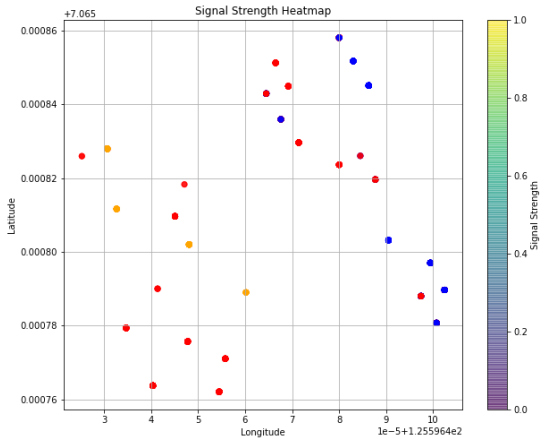
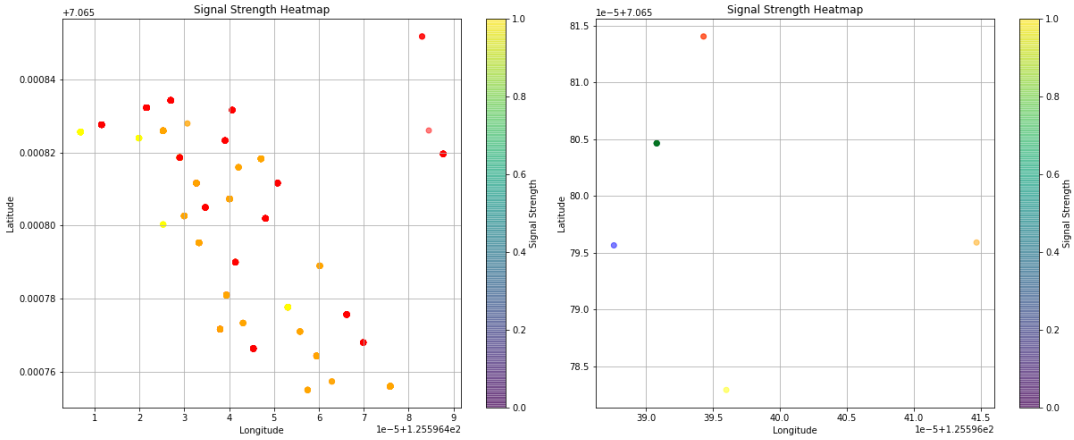
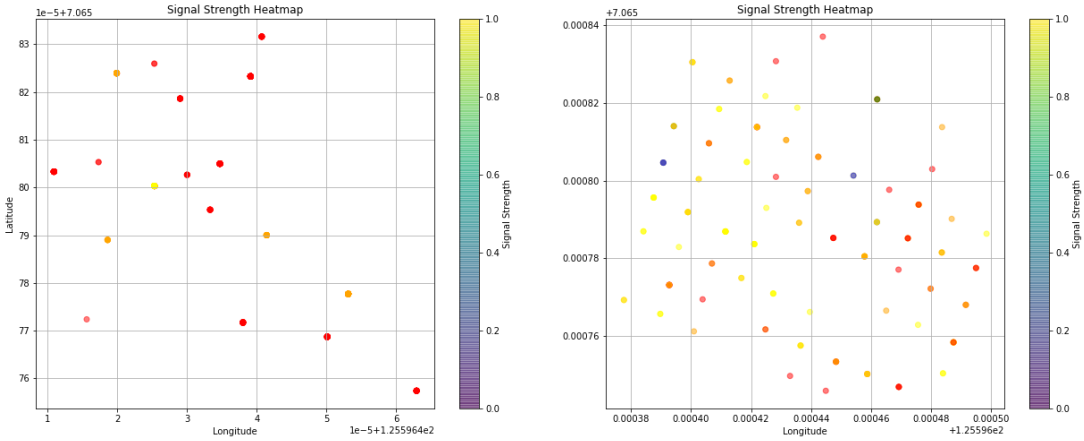


Fig. 6.1. Heatmap - Channel 2412: Floor 1



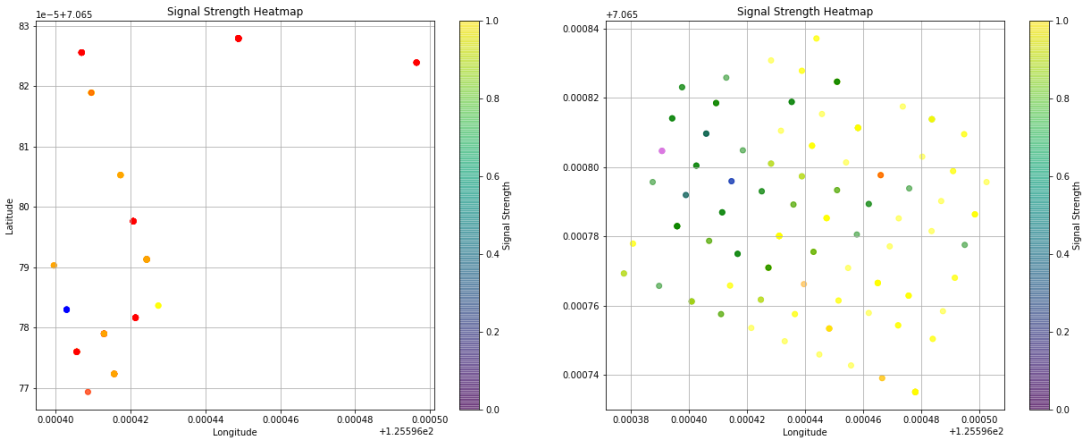
(a) (b)

Fig. 6.2. Heatmap - Channel 2417: (a) Floor 1 (b) Floor 2



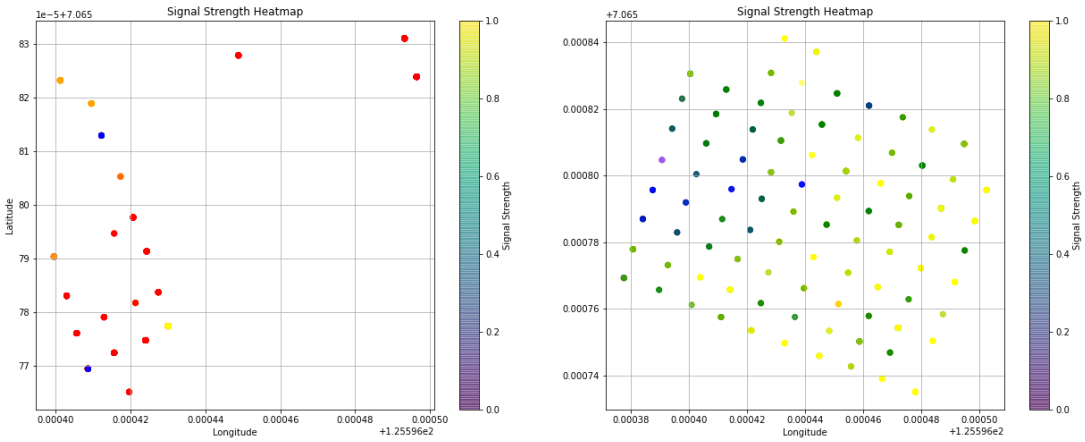
(a) (b)

Fig. 6.3. Heatmap - Channel 2422: (a) Floor 1 (b) Floor 2



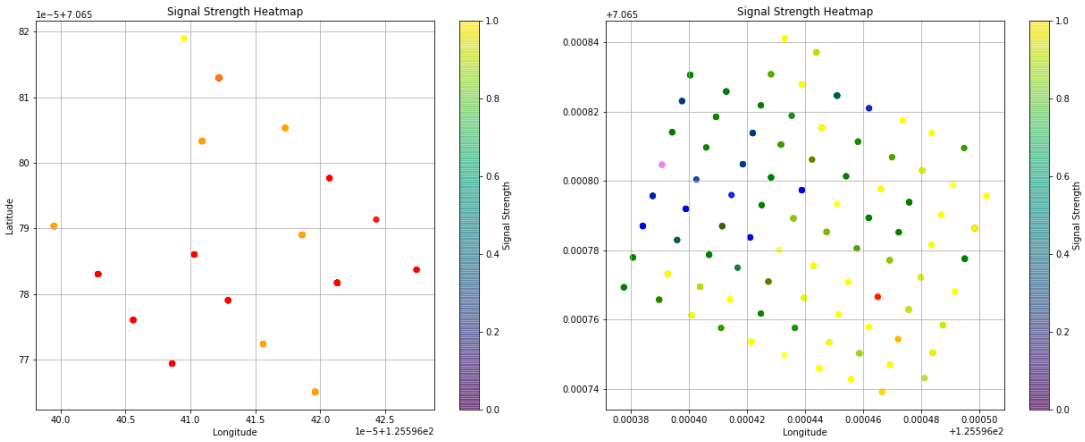
(a) (b)

Fig. 6.4. Heatmap - Channel 2427: (a) Floor 1 (b) Floor 2



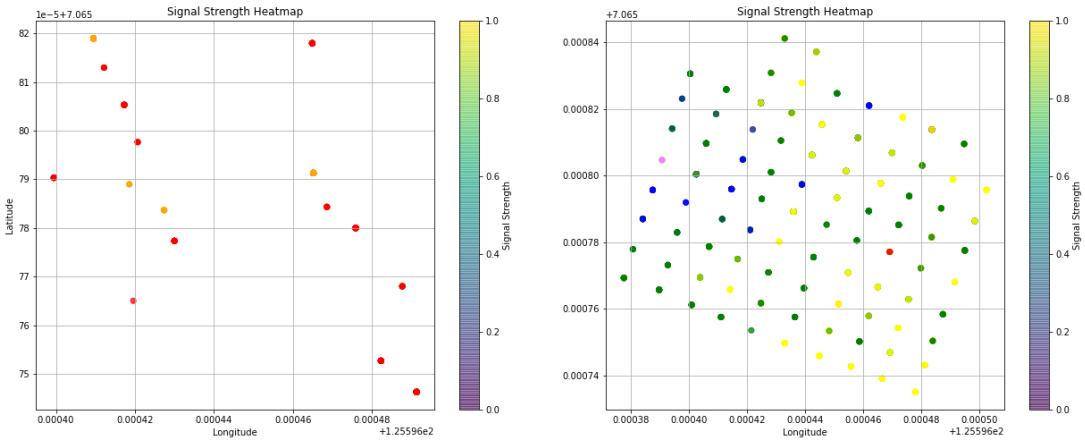
(a) (b)

Fig. 6.5. Heatmap - Channel 2432: (a) Floor 1 (b) Floor 2



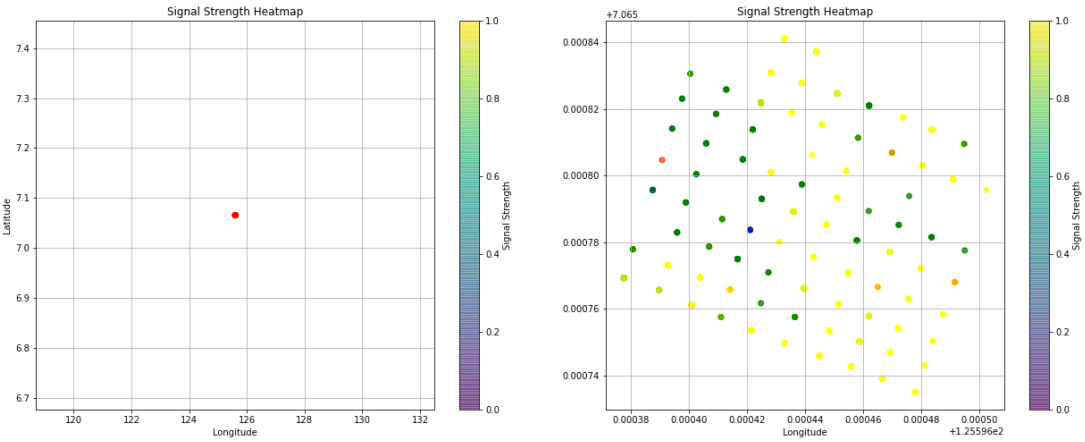
(a) (b)

Fig. 6.6. Heatmap - Channel 2437: (a) Floor 1 (b) Floor 2



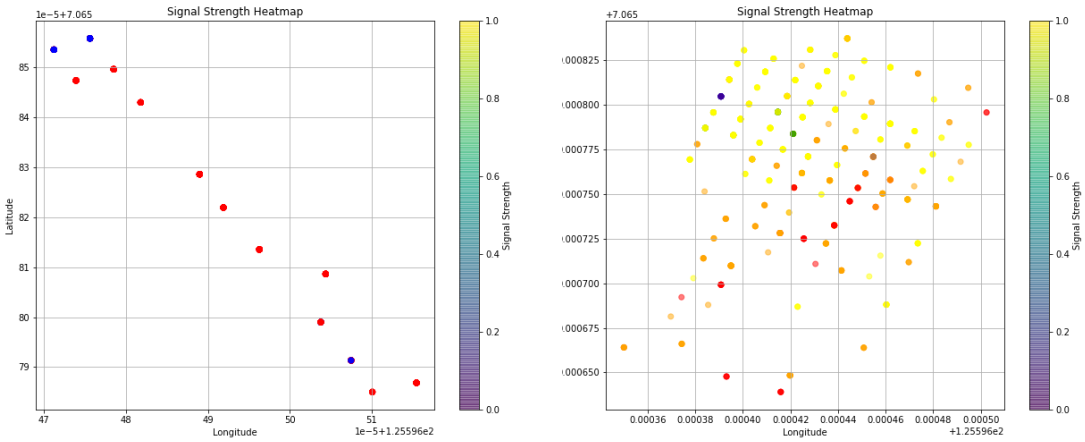
(a) (b)

Fig. 6.7. Heatmap - Channel 2442: (a) Floor 1 (b) Floor 2



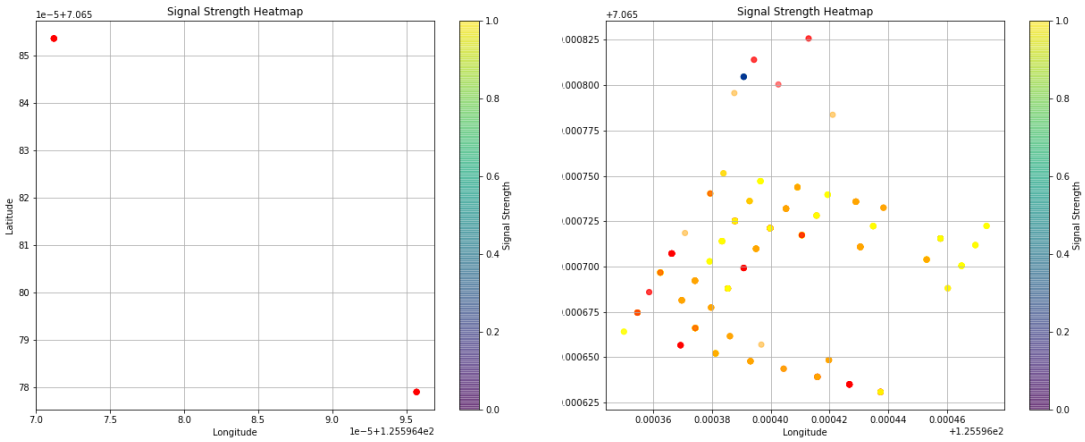
(a) (b)

Fig. 6.8. Heatmap - Channel 2447: (a) Floor 1 (b) Floor 2



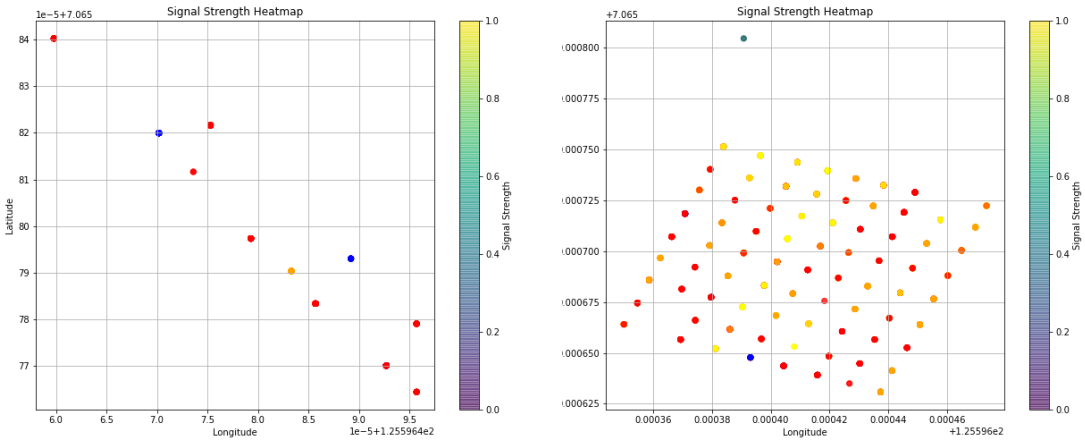
(a) (b)

Fig. 6.9. Heatmap - Channel 2452: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.10. Heatmap - Channel 2457: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.11. Heatmap - Channel 2462: (a) Floor 1 (b) Floor 2

DEVICE 4 HEATMAP READING FROM ACCESS POINT 2

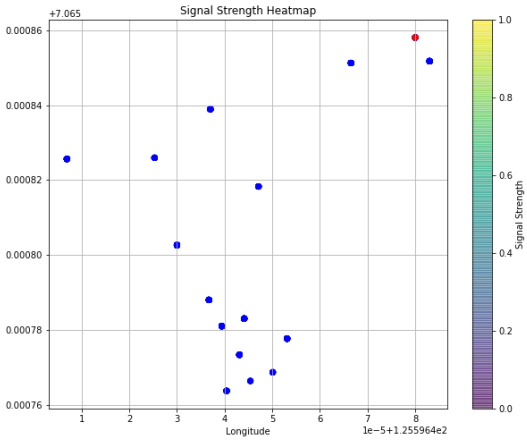


Fig. 6.12. Heatmap - Channel 2412: Floor 1

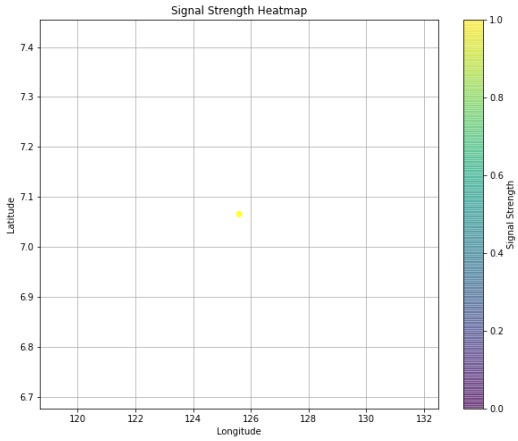


Fig. 6.13. Heatmap - Channel 2417: Floor 2

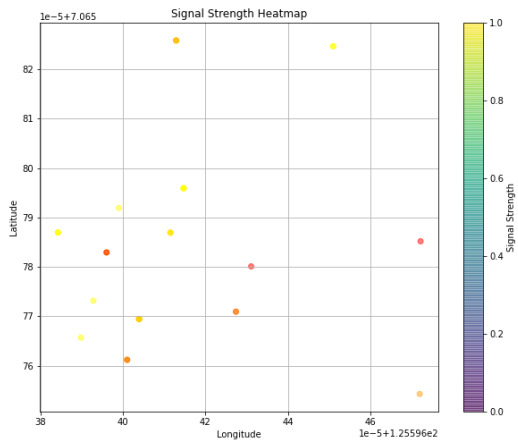
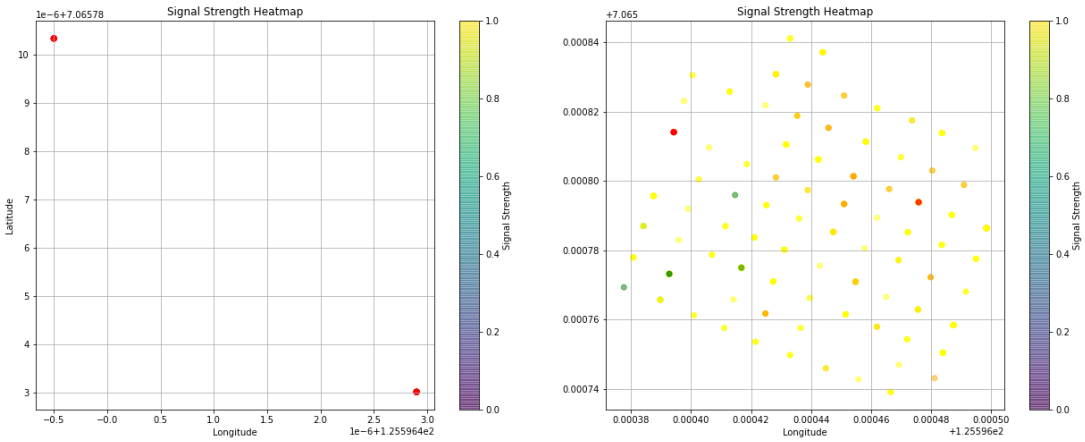
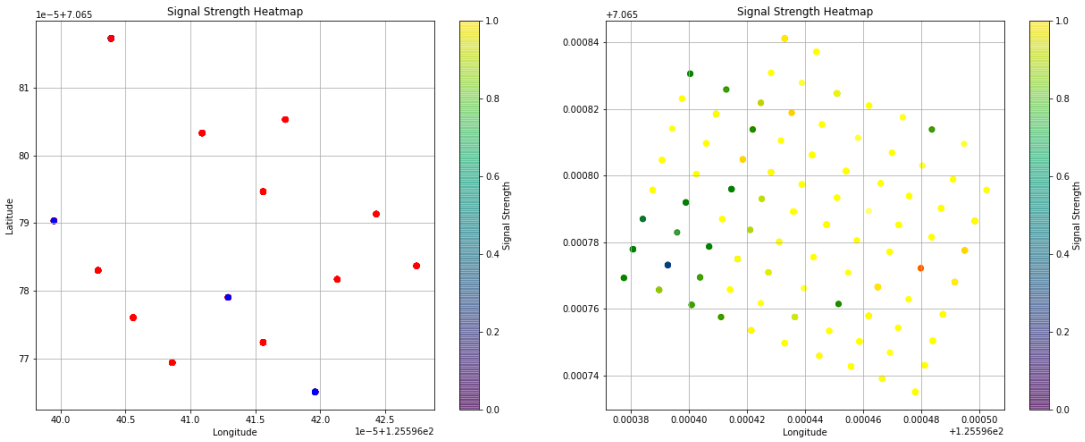


Fig. 6.14. Heatmap - Channel 2422: Floor 2



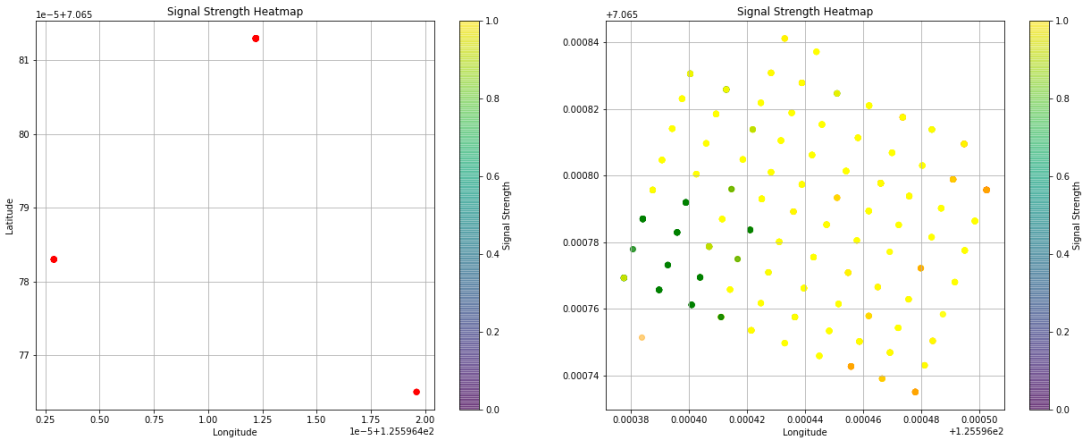
(a) (b)

Fig. 6.15. Heatmap - Channel 2427: (a) Floor 1 (b) Floor 2



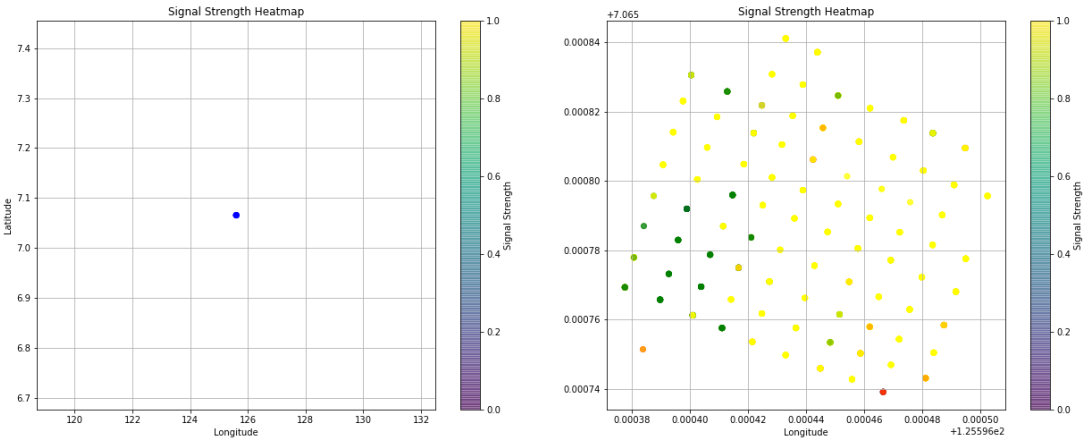
(a) (b)

Fig. 6.16. Heatmap - Channel 2432: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.17. Heatmap - Channel 2437: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.18. Heatmap - Channel 2442: (a) Floor 1 (b) Floor 2

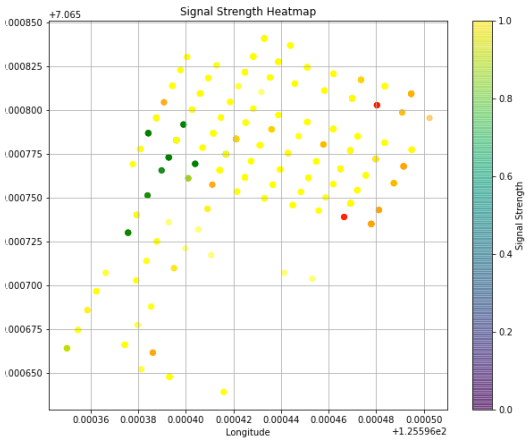
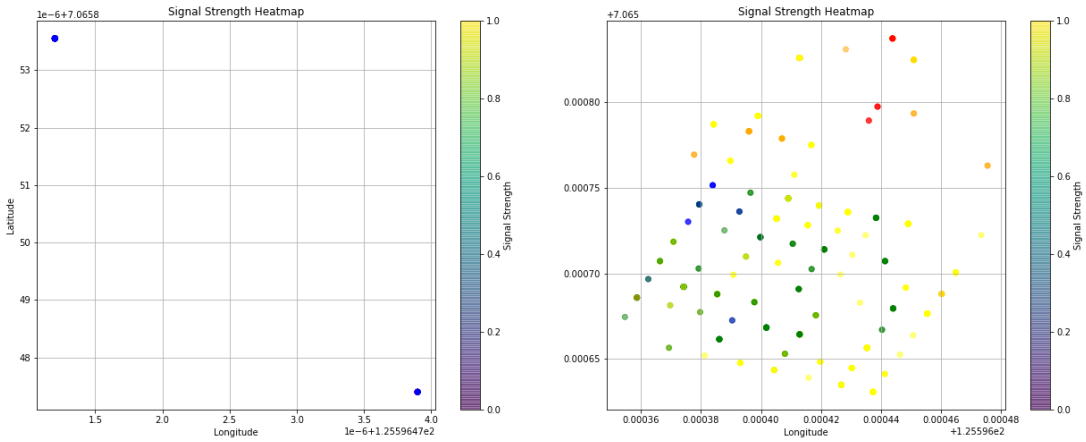
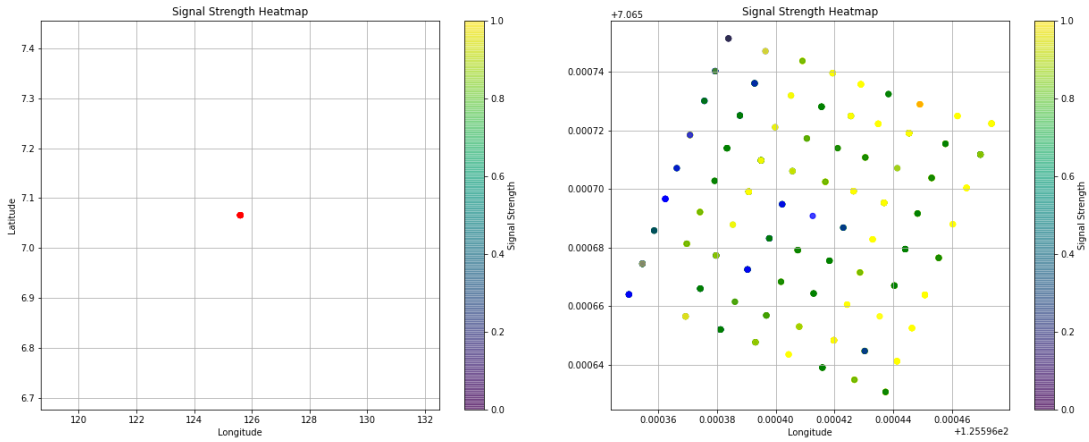


Fig. 6.19. Heatmap - Channel 2447: Floor 2



(a) (b)

Fig. 6.20. Heatmap - Channel 2452: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.21. Heatmap - Channel 2457: (a) Floor 1 (b) Floor 2

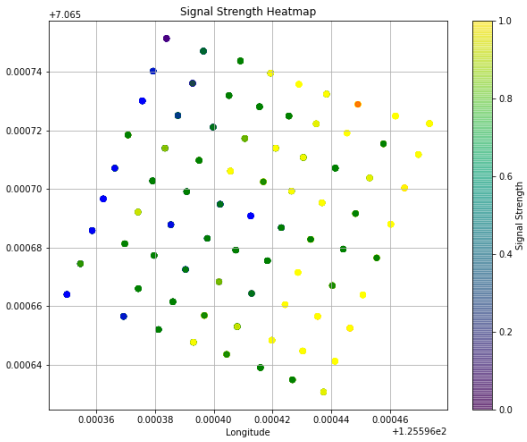


Fig. 6.22. Heatmap - Channel 2462: Floor 2

DEVICE 4 HEATMAP READING FROM ACCESS POINT 3

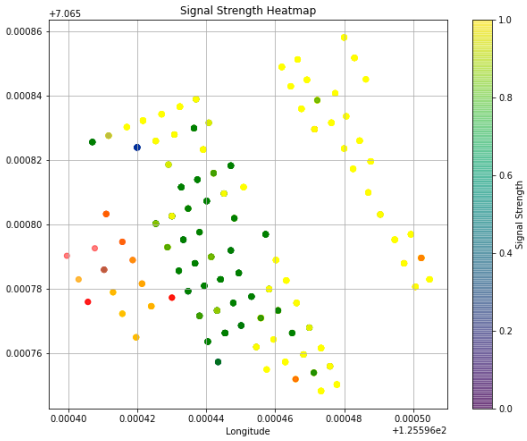


Fig. 6.23. Heatmap - Channel 2412: Floor 1

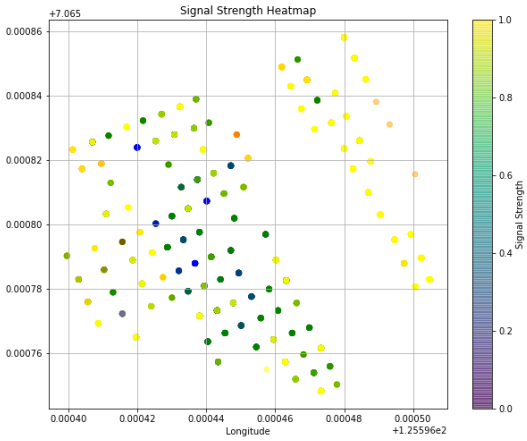


Fig. 6.24. Heatmap - Channel 2417: Floor 1

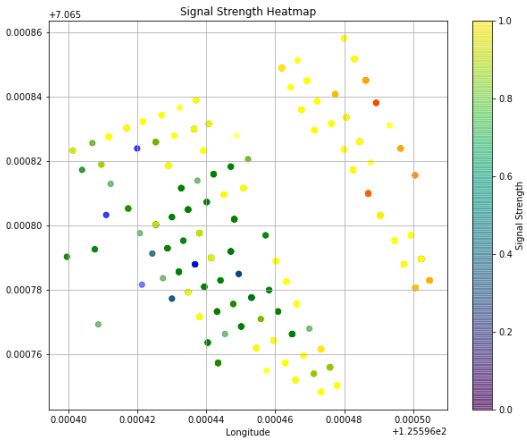
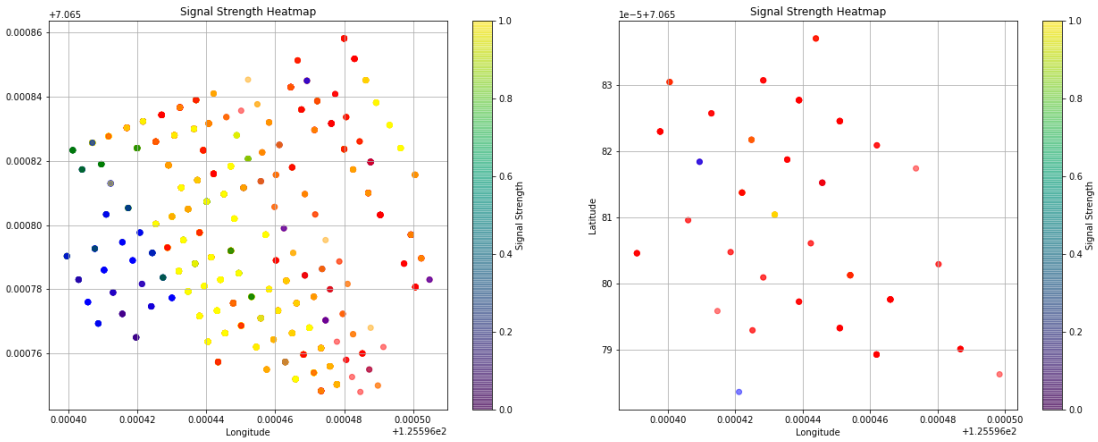
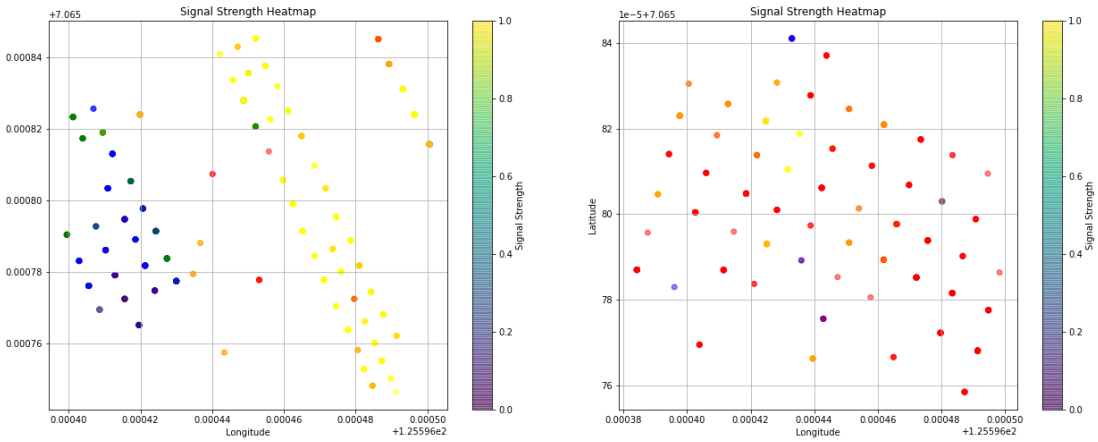


Fig. 6.25. Heatmap - Channel 2422: Floor 1



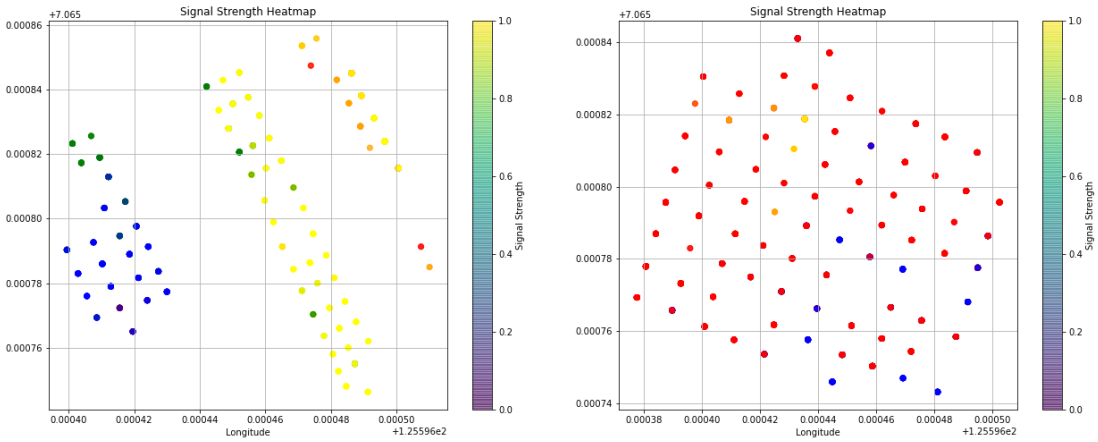
(a) (b)

Fig. 6.26. Heatmap - Channel 2427: (a) Floor 1 (b) Floor 2



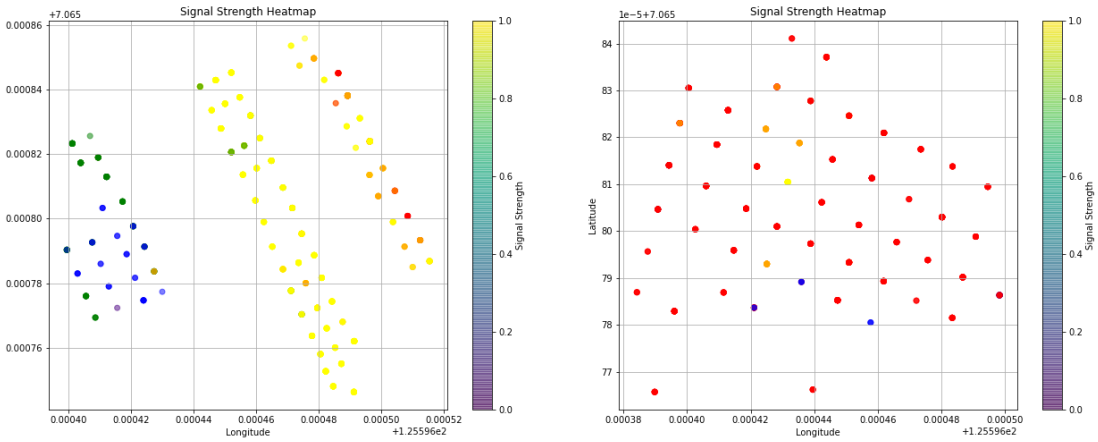
(a) (b)

Fig. 6.27. Heatmap - Channel 2432: (a) Floor 1 (b) Floor 2



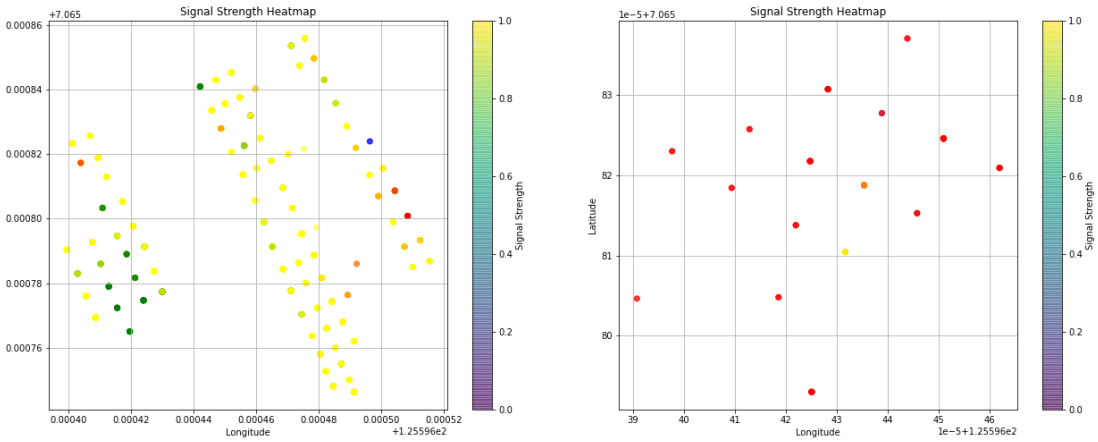
(a) (b)

Fig. 6.28. Heatmap - Channel 2437: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.29. Heatmap - Channel 2442: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.30. Heatmap - Channel 2447: (a) Floor 1 (b) Floor 2

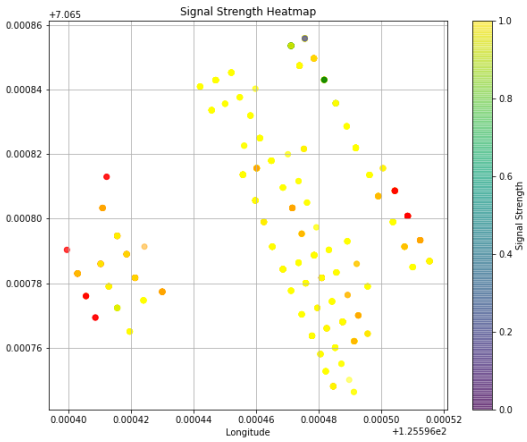


Fig. 6.31. Heatmap - Channel 2452: Floor 1

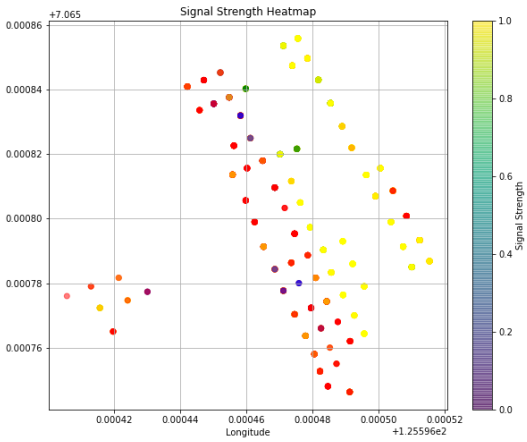
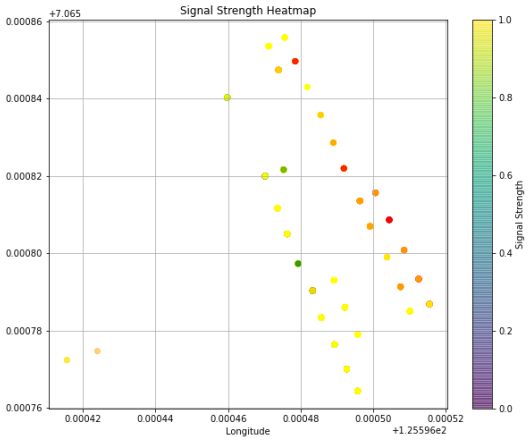
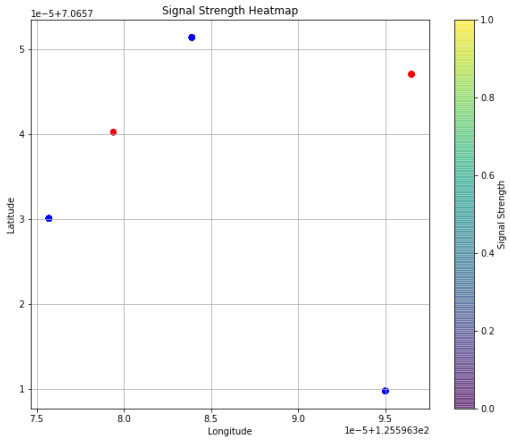


Fig. 6.32. Heatmap - Channel 2457: Floor 1

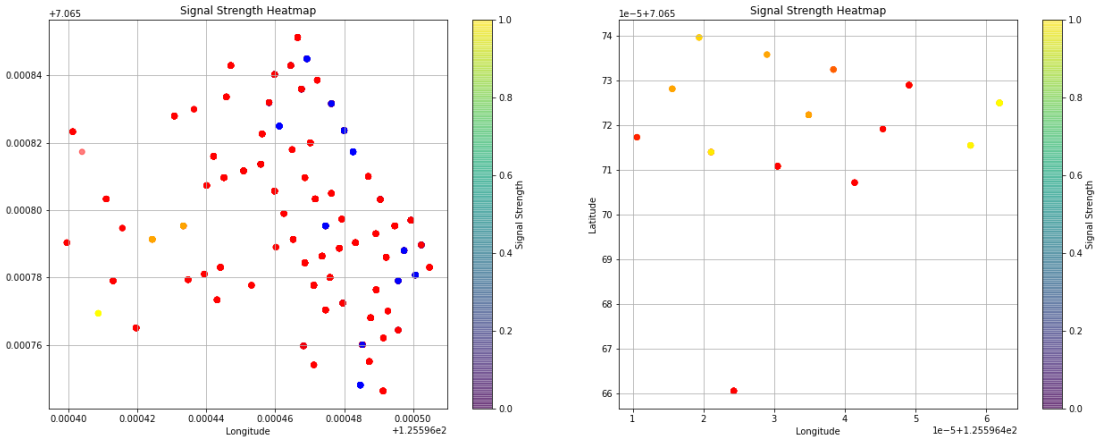




(a) (b)

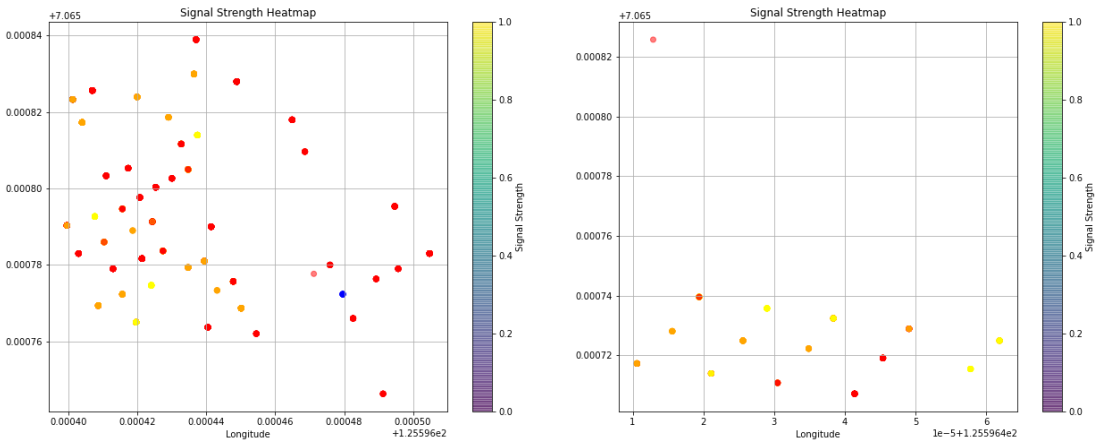
Heatmap - Channel 2462: (a) Floor 1 (b) Floor 2

DEVICE 6 HEATMAP READING FROM ACCESS POINT 1



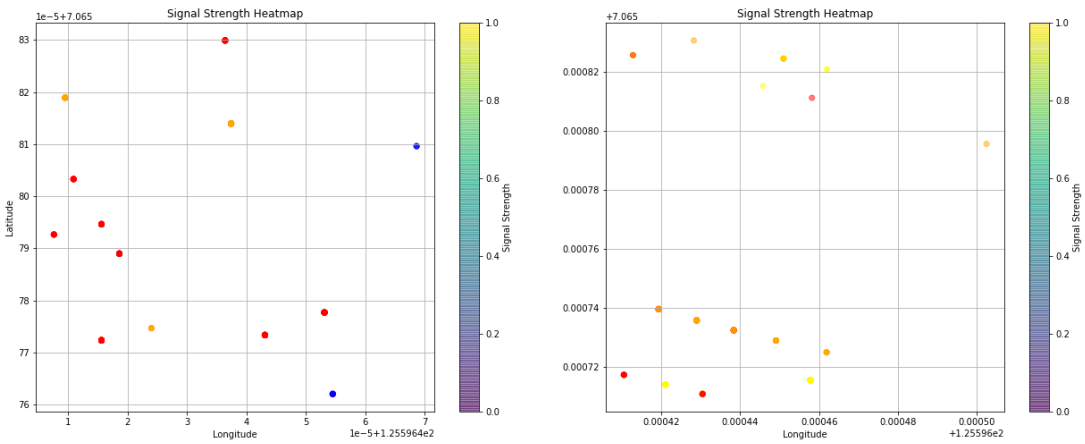
(a) (b)

Fig. 6.33. Heatmap - Channel 2412: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.34. Heatmap - Channel 2417: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.35. Heatmap - Channel 2422: (a) Floor 1 (b) Floor 2

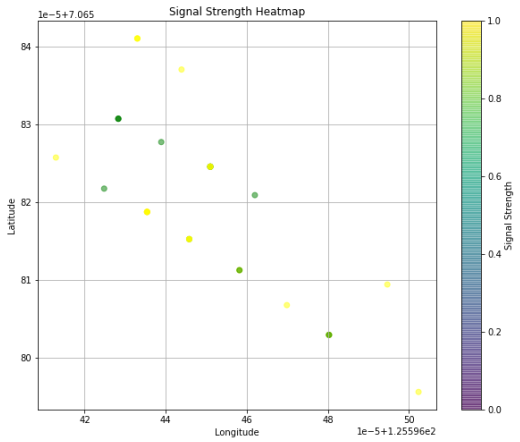


Fig. 6.36. Heatmap - Channel 2427: Floor 2

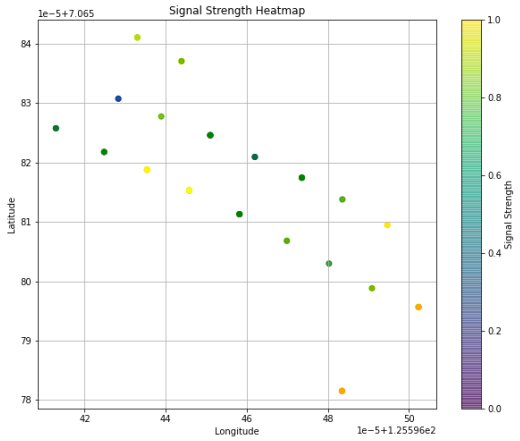


Fig. 6.37. Heatmap - Channel 2432: Floor 2

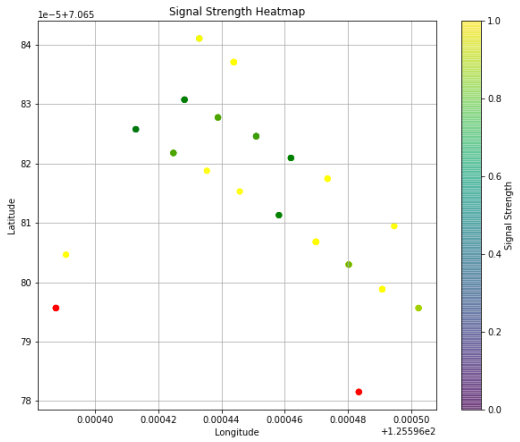


Fig. 6.38. Heatmap - Channel 2437: Floor 2

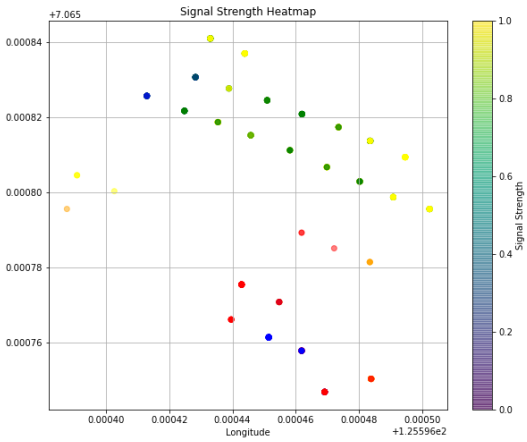


Fig. 6.39. Heatmap - Channel 2442: Floor 2

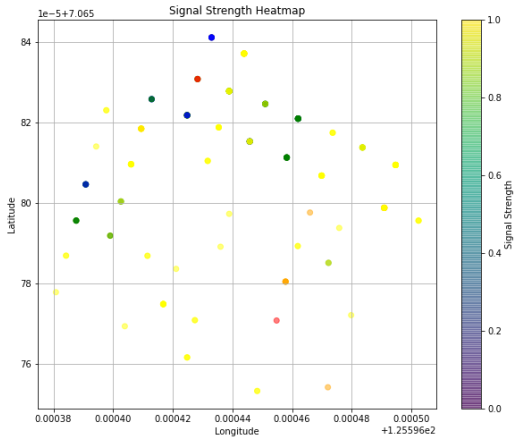
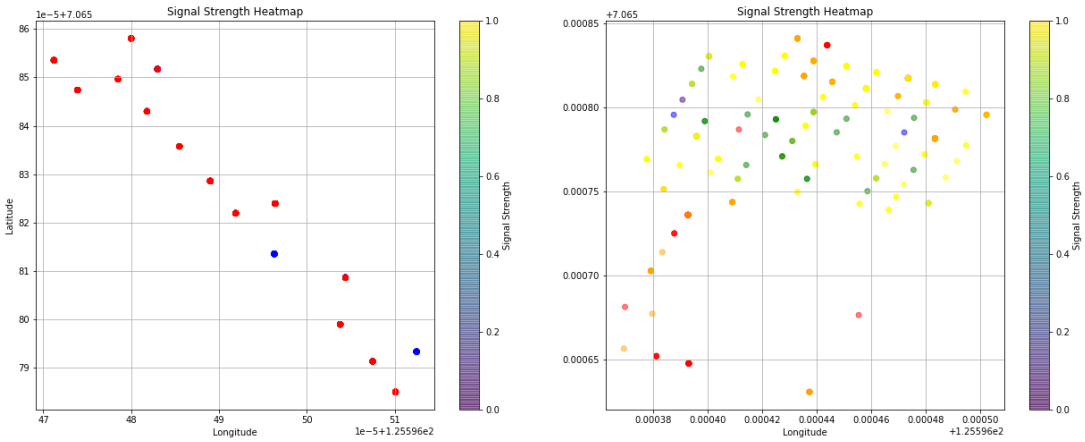
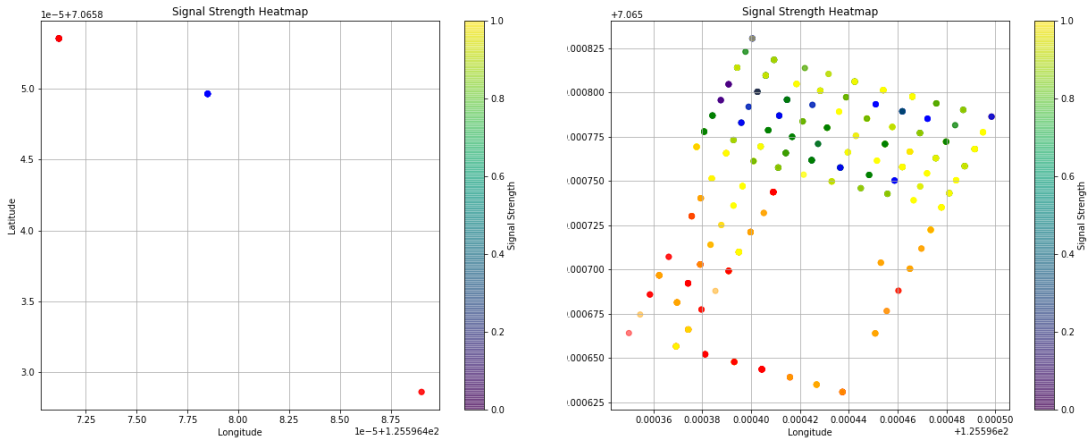


Fig. 6.40. Heatmap - Channel 2447: Floor 2



(a) (b)

Fig. 6.41. Heatmap - Channel 2452: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.42. Heatmap - Channel 2457: (a) Floor 1 (b) Floor 2

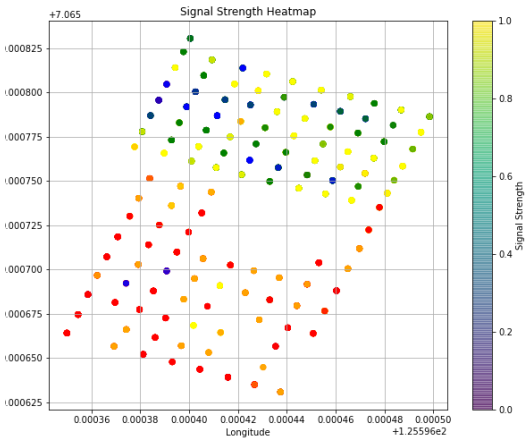
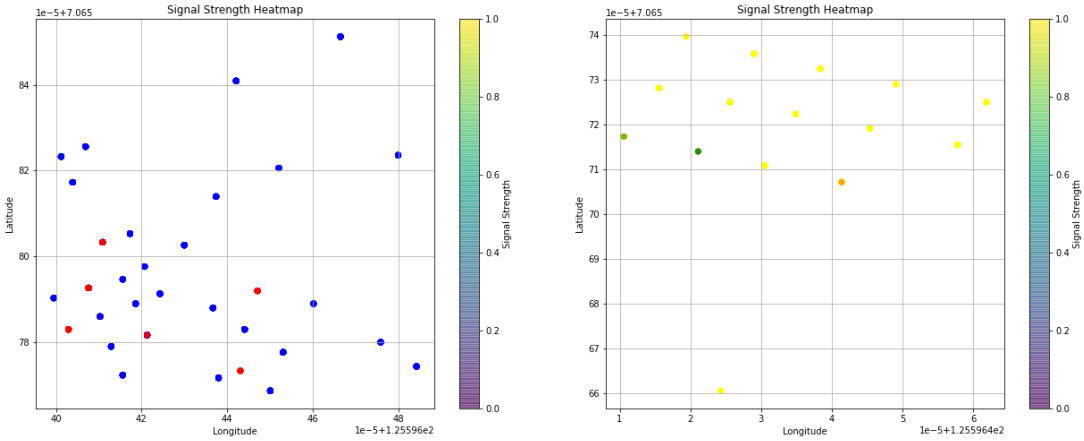


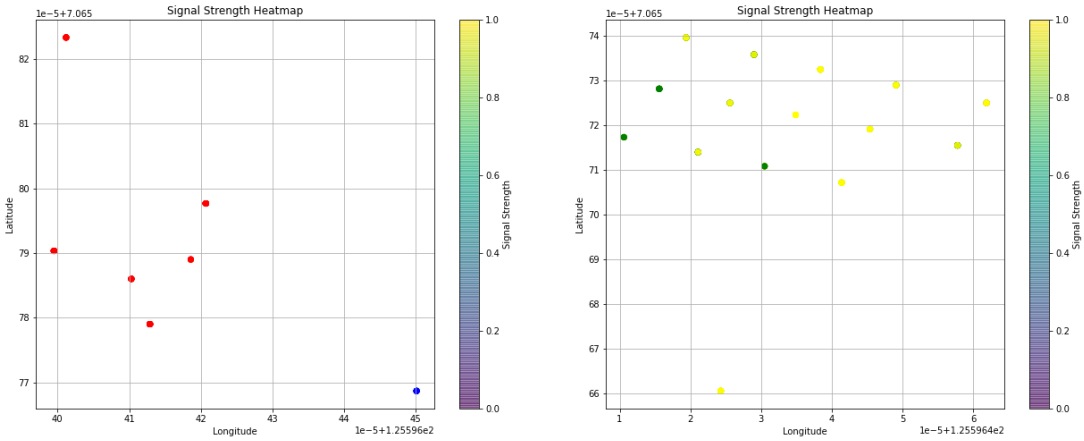
Fig. 6.43. Heatmap - Channel 2462: Floor 2

DEVICE 6 HEATMAP READING FROM ACCESS POINT 2



(a) (b)

Fig. 6.44. Heatmap - Channel 2412: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.45. Heatmap - Channel 2417: (a) Floor 1 (b) Floor 2

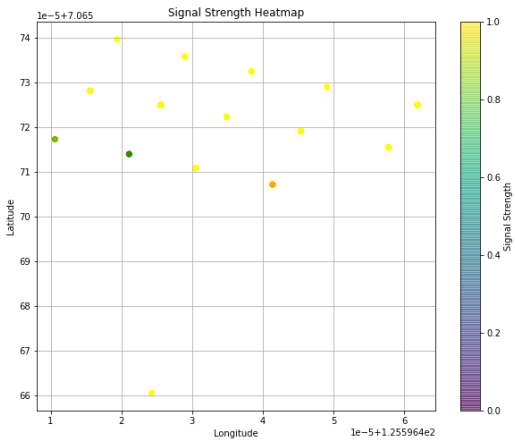


Fig. 6.46. Heatmap - Channel 2422: Floor 2

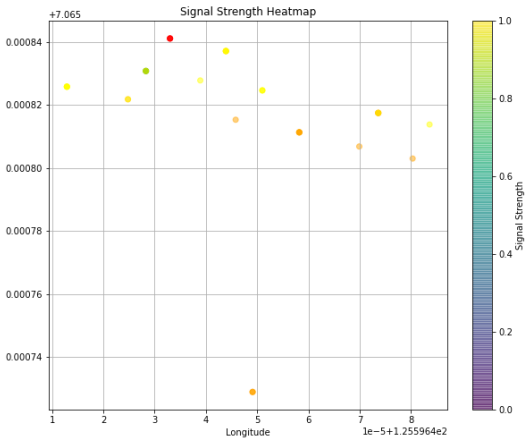


Fig. 6.47. Heatmap - Channel 2427: Floor 2

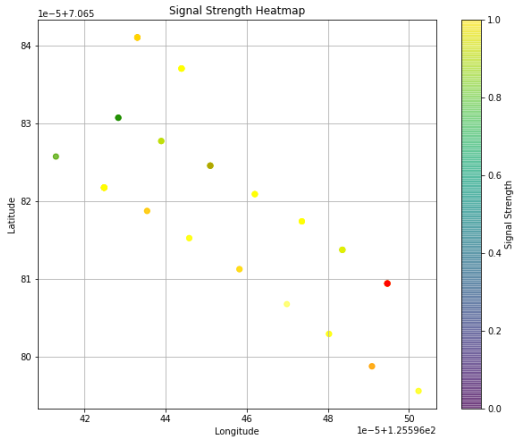


Fig. 6.48. Heatmap - Channel 2432: Floor 2

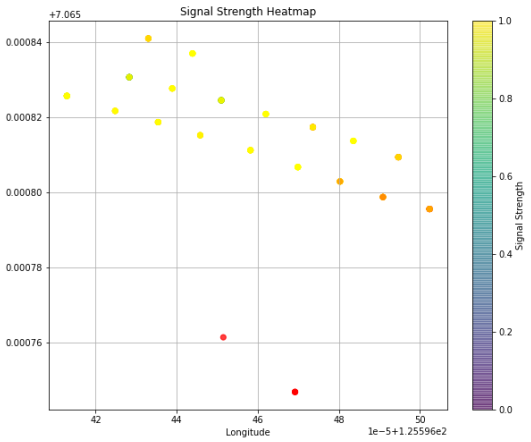


Fig. 6.49. Heatmap - Channel 2437: Floor 2

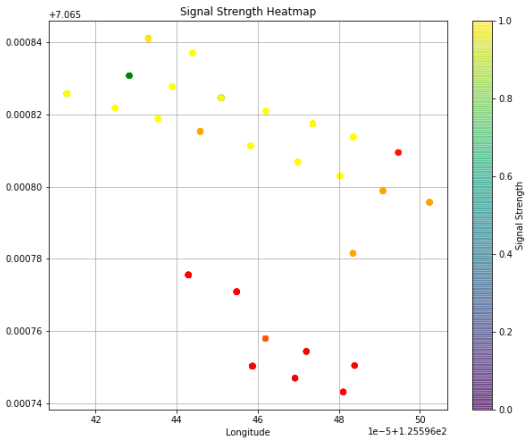


Fig. 6.50. Heatmap - Channel 2442: Floor 2

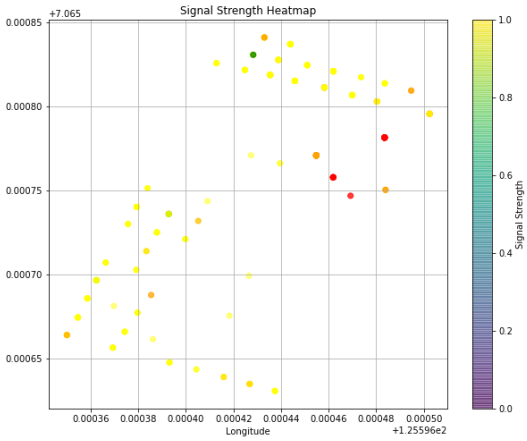
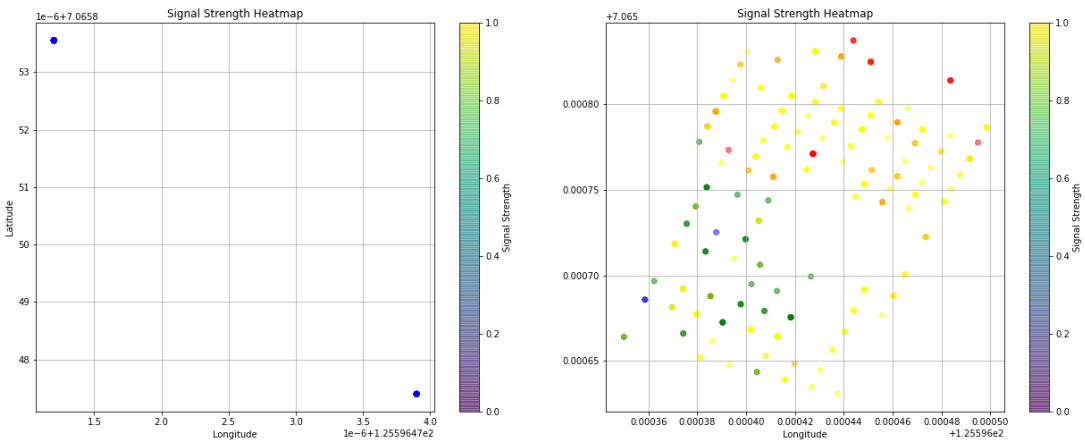


Fig. 6.51. Heatmap - Channel 2447: Floor 2



(a) (b)

Fig. 6.52. Heatmap - Channel 2452: (a) Floor 1 (b) Floor 2

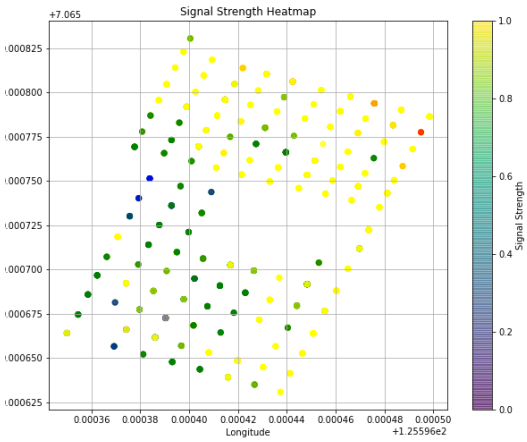


Fig. 6.53. Heatmap - Channel 2457: Floor 2

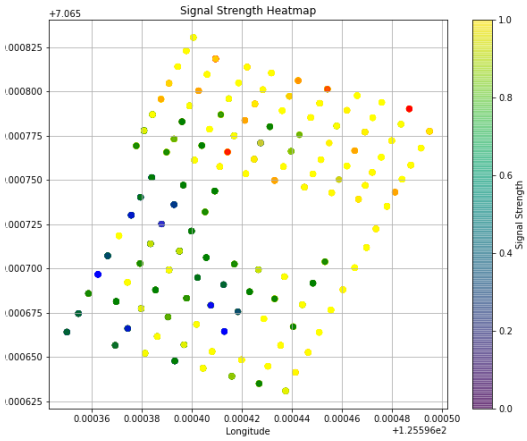
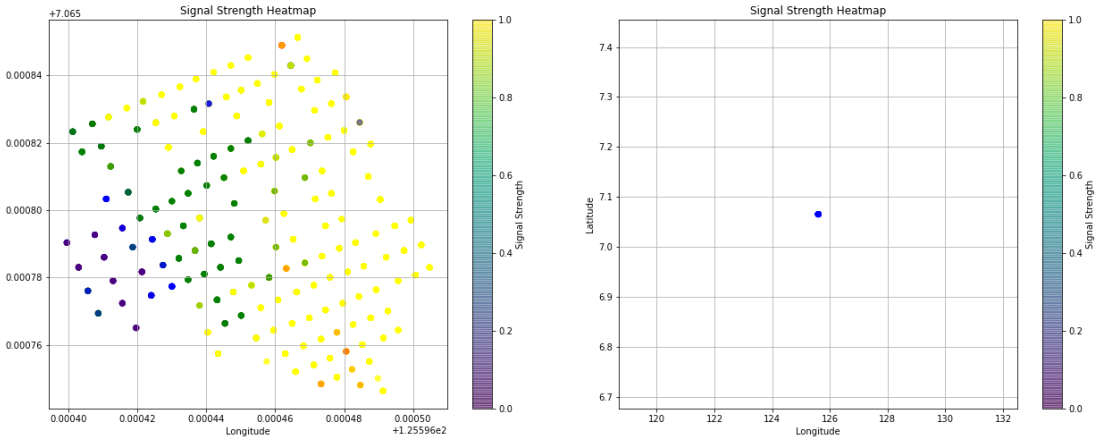


Fig. 6.54. Heatmap - Channel 2462: Floor 2

DEVICE 6 HEATMAP READING FROM ACCESS POINT 3



(a) (b)

Fig. 6.55. Heatmap - Channel 2412: (a) Floor 1 (b) Floor 2

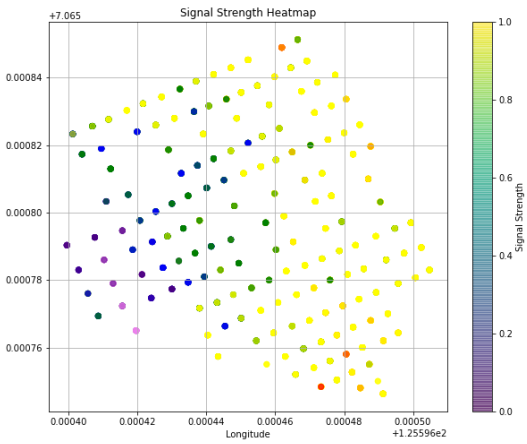


Fig. 6.56. Heatmap - Channel 2417: Floor 1

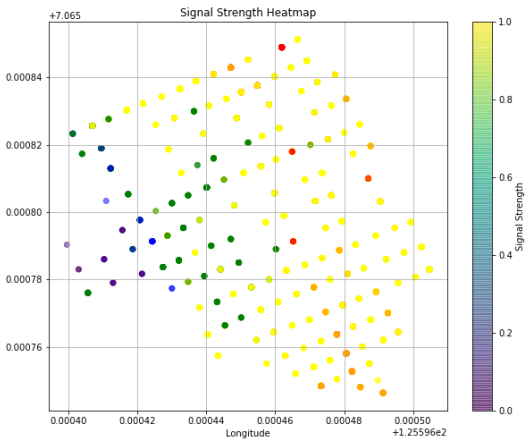
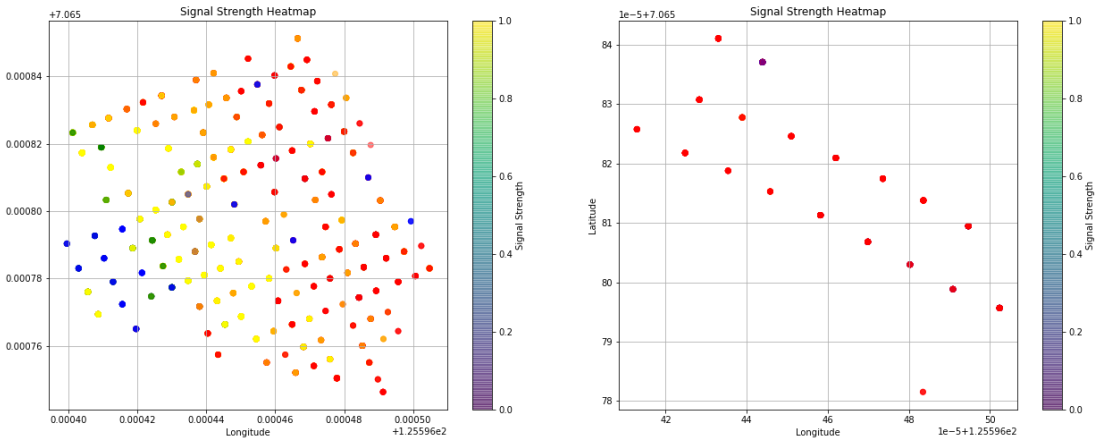
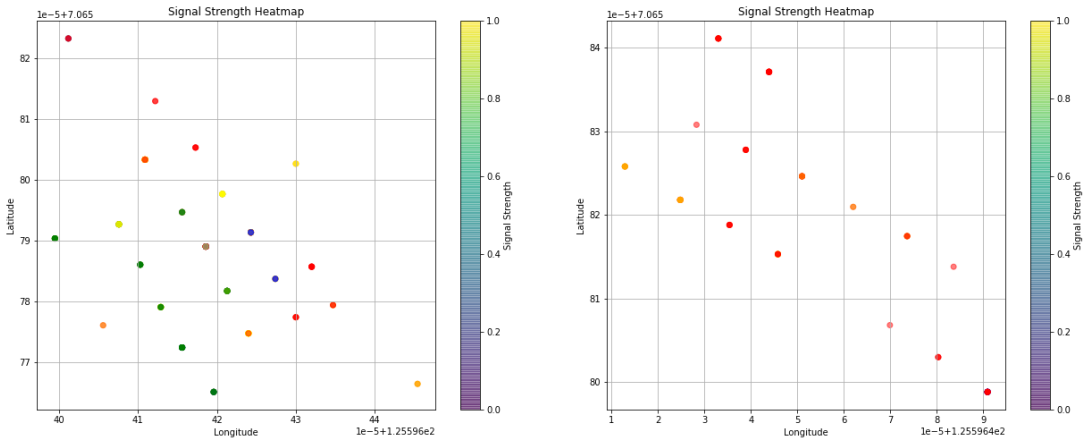


Fig. 6.57. Heatmap - Channel 2422: Floor 1



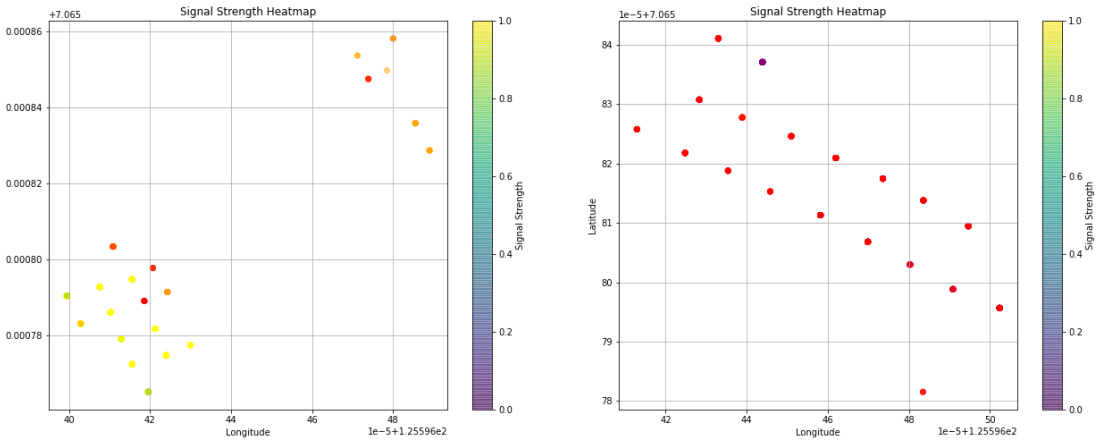
(a) (b)

Fig. 6.58. Heatmap - Channel 2427: (a) Floor 1 (b) Floor 2



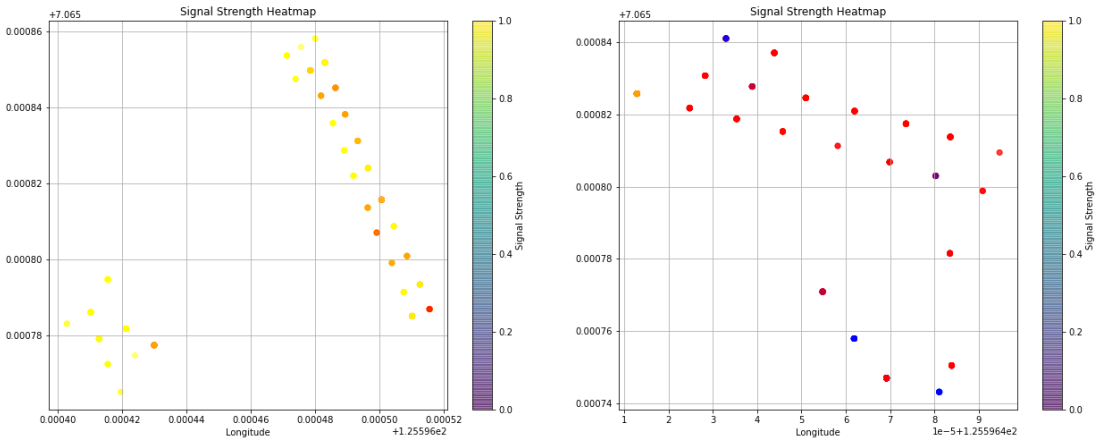
(a) (b)

Fig. 6.59. Heatmap - Channel 2432: (a) Floor 1 (b) Floor 2



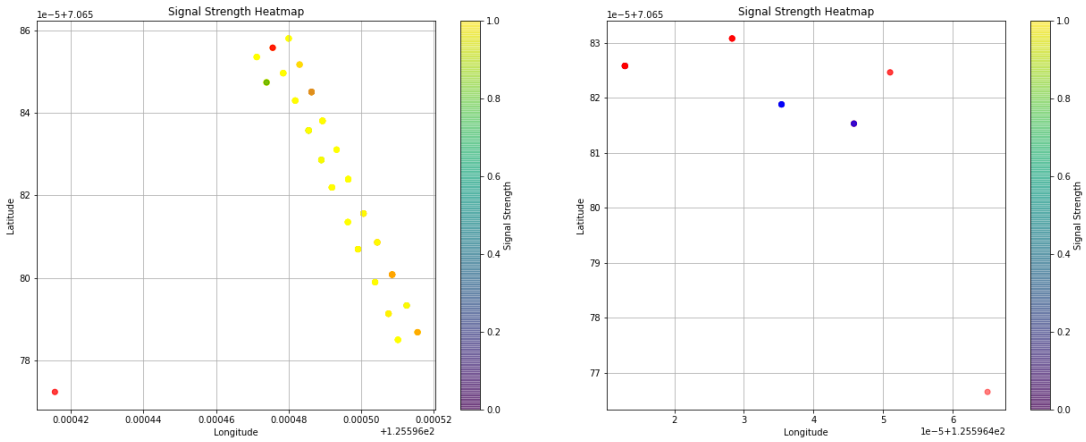
(a) (b)

Fig. 6.60. Heatmap - Channel 2437: (a) Floor 1 (b) Floor 2



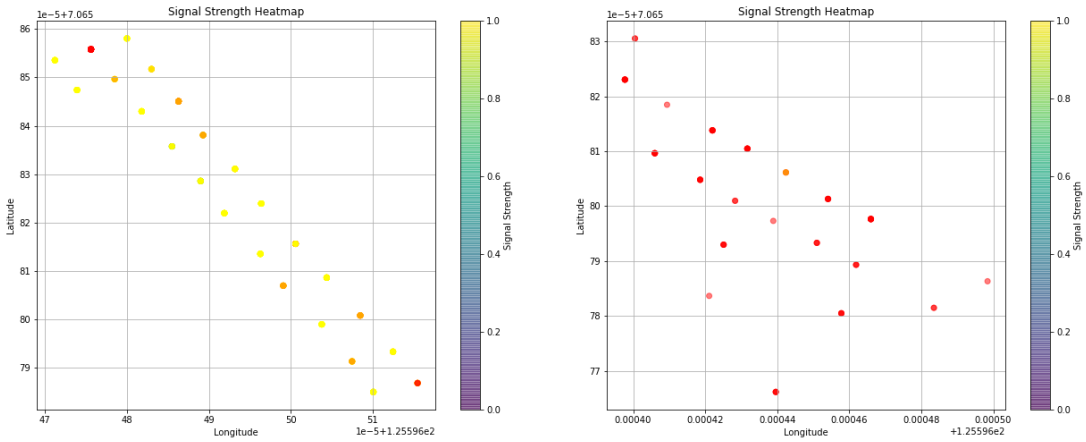
(a) (b)

Fig. 6.61. Heatmap - Channel 2442: (a) Floor 1 (b) Floor 2



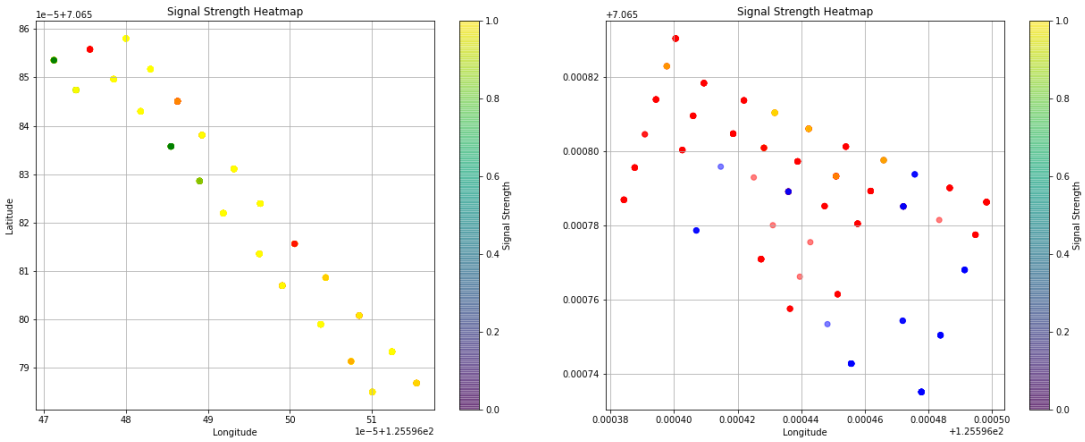
(a) (b)

Fig. 6.62. Heatmap - Channel 2447: (a) Floor 1 (b) Floor 2



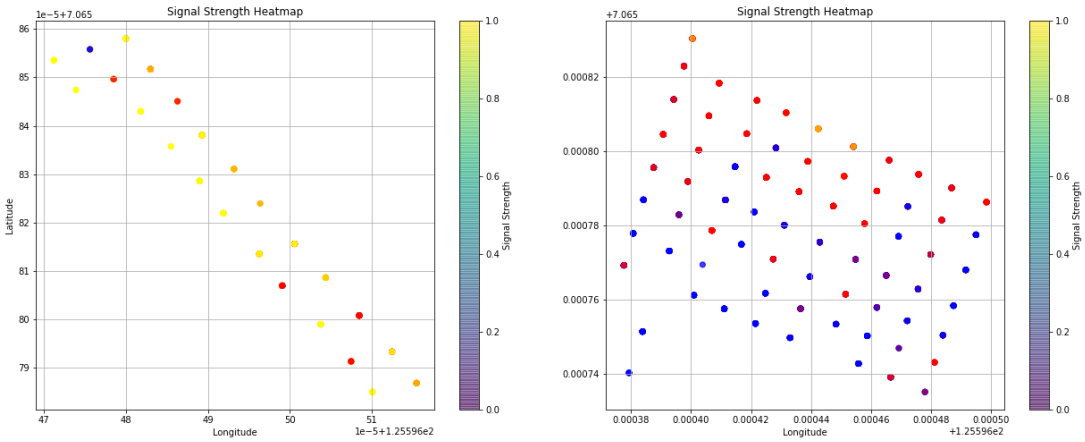
(a) (b)

Fig. 6.63. Heatmap - Channel 2452: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.64. Heatmap - Channel 2457: (a) Floor 1 (b) Floor 2



(a) (b)

Fig. 6.65. Heatmap - Channel 2462: (a) Floor 1 (b) Floor 2

APPENDIX G

Calibration Coefficients and Linear Regression Analysis

These are the calibration coefficients obtained from three access points along with their corresponding channels. The researcher was able to survey various RSSI data but was unable to cover all of the channels completely. The lack of RSSI data for certain channels is due to the location where the data was surveyed. Frequencies automatically shift to stronger RSSI signals, so the closer the device, the stronger the RSSI data. The table shows a blue indicator corresponding to the calibration coefficient from the surveyed RSSI data from device 4, while the red indicator represents the data from device 6.

TABLE X. DEVICE CALIBRATION COEFFICIENT a_S FROM ACCESS POINT 1

Device ID	**Channel**										
	2412	*2417*	*2422*	*2427*	*2432*	*2437*	*2442*	*2447*	*2452*	*2457*	*2462*
1	-0.024	0.04	-0.0778	0.6731				-0.6667	0.2067	0.08	- 0.1609
2	0.2827	-0.0036	-0.0465	0.2202				- 0.119	0.149	0.425	0.2363
3			0.6694	-0.8452	0.2655	0.0444	-0.2599	- 0.1774	- 0.3546	0.1188	- 0.2958
5			-0.1676					-0.1727	- 0.7143	- 0.0498	

TABLE XI. DEVICE CALIBRATION COEFFICIENT b_S FROM ACCESS POINT 1

Device ID	**Channel**										
	2412	*2417*	*2422*	*2427*	*2432*	*2437*	*2442*	*2447*	*2452*	*2457*	*2462*
1	- 61.023	- 56.859	- 70.582	- 25.269				-124.17	-41.894	- 53.266	- 64.342
2	- 36.555	-53.377	-57.052	- 47.943				- 83.807	-44.829	- 28.729	- 38.416
3			-33.635	9.2976	-37.651	- 50.416	- 69.391	- 67.56	- 91.039	- 44.925	- 106.97
5			-82.611					-79.866	- 114.57	- 72.19	

TABLE XII. DEVICE CALIBRATION COEFFICIENT a_S FROM ACCESS POINT 2

Device ID	**Channel**										
	2412	*2417*	*2422*	*2427*	*2432*	*2437*	*2442*	*2447*	*2452*	*2457*	*2462*
1	0.2101	0.2417	0.3673						0.132	0.2378	0.1852
2	- 0.0482	0.1959	0.3049					- 0.119	0.9795	0.0334	0.1738
3					- 0.0883	0.7726	0.2563	0.3658	-0.7243	0.0499	0.2272
5	0.225	0.0288	0.0153								

TABLE XIII. DEVICE CALIBRATION COEFFICIENT b FROM ACCESS POINT 2

Device ID	**Channel**										
	2412	*2417*	*2422*	*2427*	*2432*	*2437*	*2442*	*2447*	*2452*	*2457*	*2462*
1	- 57.413	- 51.294	- 44.935						- 60.967	- 50.732	- 56.06
2	- 80.835	- 59.09	- 51.899					- 83.807	5.2032	- 63.397	- 50.895
3					- 65.034	- 21.769	- 54.682	- 49.46	- 137.61	- 64.936	- 53.008
5	- 62.581	- 68.802	- 73.517								

TABLE XIV. DEVICE CALIBRATION COEFFICIENT a_S FROM ACCESS POINT 3

Device ID	**Channel**										
	2412	*2417*	*2422*	*2427*	*2432*	*2437*	*2442*	*2447*	*2452*	*2457*	*2462*
1	0.7291	0.39	0.3685		0.6207	0.3518	- 0.1255	- 0.0812	0.0747	0.1002	0.1534
2	-0.0186	- 0.2875	0.324		0.3929	0.1458	0.1047	0.0601	0.3107	-0.0519	0.277
3					0.7924	0.7078	-0.081	- 0.0129	0.175	0.0073	0.2018
5	0.3878	0.021	0.0396	- 1.0054		0.0695					

TABLE XV. DEVICE CALIBRATION COEFFICIENT b_S FROM ACCESS POINT 3

Device ID	**Channel**										
	2412	*2417*	*2422*	*2427*	*2432*	*2437*	*2442*	*2447*	*2452*	*2457*	*2462*
1	- 23.486	- 47.237	- 54.482		- 23.241	- 56.25	- 71.997	- 53.422	- 70.878	- 31.957	- 65.258
2	- 87.503	- 108.68	- 60.304		- 36.143	- 63.171	- 53.554	- 43.927	- 56.937	- 82.597	-12.981
3					- 6.1627	- 16.978	-71.33	- 52.832	- 26.45	- 38.155	- 59.922
5	- 9.6743	- 36.732	- 37.967	- 97.567		- 68.684					

MULTIVARIATE LINEAR REGRESSION ANALYSIS FROM ACCESS POINT 1 USING DEVICE 4 AS THE REFERENCE DEVICE

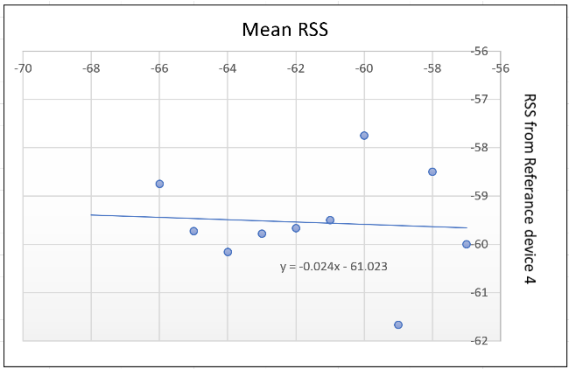


Fig. 7.1 Multivariate Linear Regression Analysis: Channel 2412 between Device 4 and Device 1

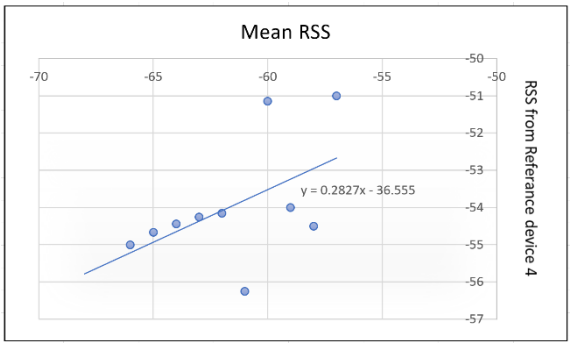


Fig. 7.2 Multivariate Linear Regression Analysis: Channel 2412 between Device 4 and Device 2

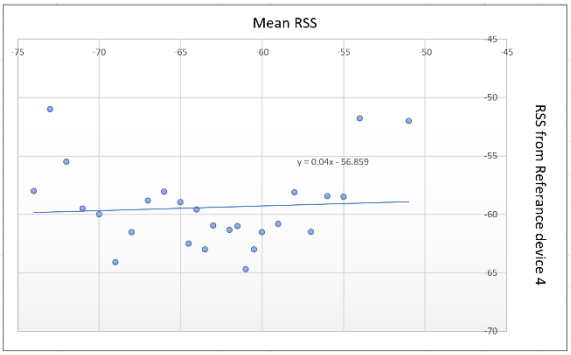


Fig. 7.3 Multivariate Linear Regression Analysis: Channel 2417 between Device 4 and Device 1

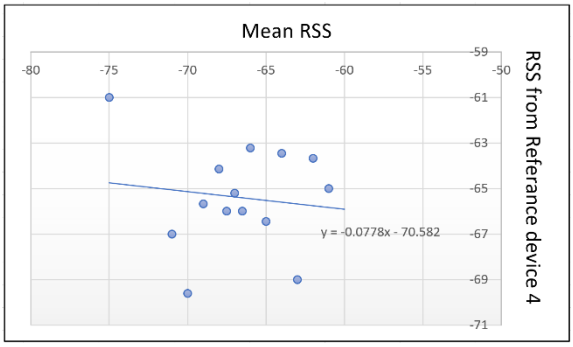


Fig. 7.4 Multivariate Linear Regression Analysis: Channel 2422 between Device 4 and Device 1

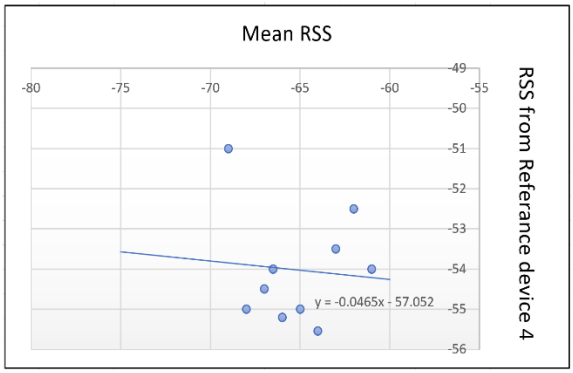


Fig. 7.5 Multivariate Linear Regression Analysis: Channel 2422 between Device 4 and Device 2

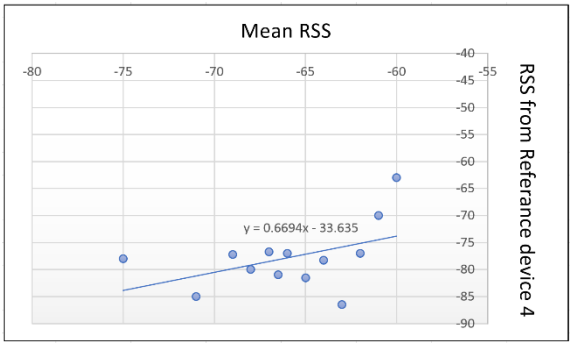


Fig. 7.6 Multivariate Linear Regression Analysis: Channel 2422 between Device 4 and Device 3

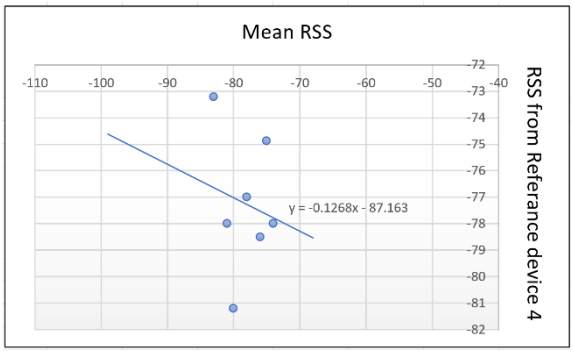


Fig. 7.7 Multivariate Linear Regression Analysis: Channel 2427 between Device 4 and Device 1

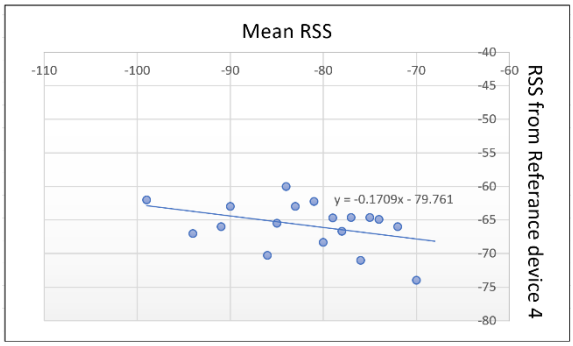


Fig. 7.8 Multivariate Linear Regression Analysis: Channel 2427 between Device 4 and Device 2

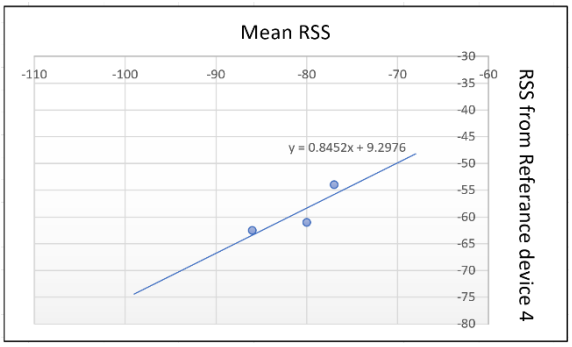


Fig. 7.9 Multivariate Linear Regression Analysis: Channel 2427 between Device 4 and Device 3

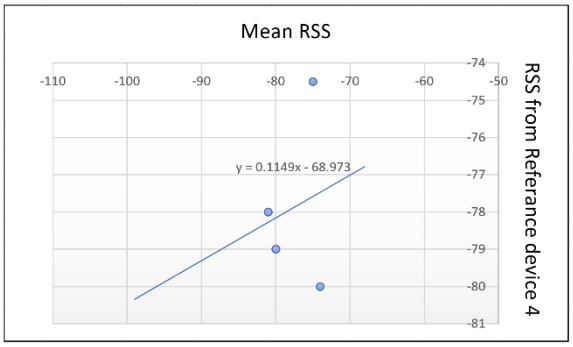


Fig. 7.10 Multivariate Linear Regression Analysis: Channel 2427 between Device 4 and Device 6

MULTIVARIATE LINEAR REGRESSION ANALYSIS FROM ACCESS POINT 2 USING DEVICE 4 AS THE REFERENCE DEVICE

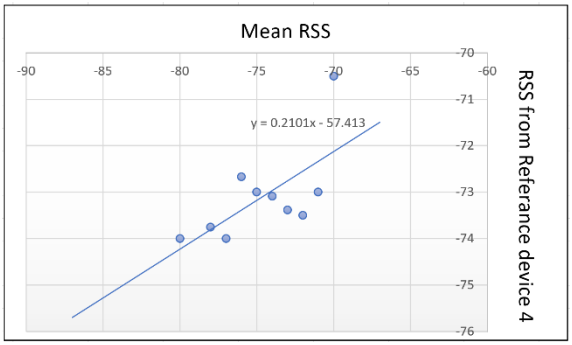


Fig. 7.11 Multivariate Linear Regression Analysis: Channel 2412 between Device 4 and Device 1

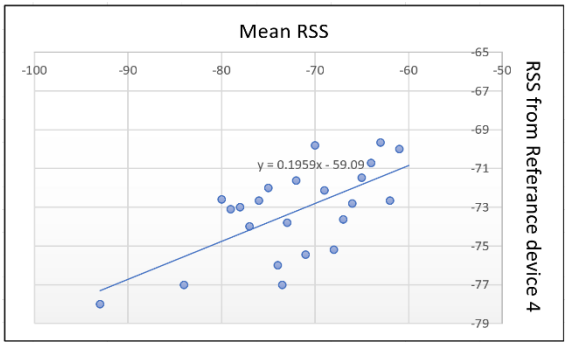


Fig. 7.14 Multivariate Linear Regression Analysis: Channel 2417 between Device 4 and Device 2

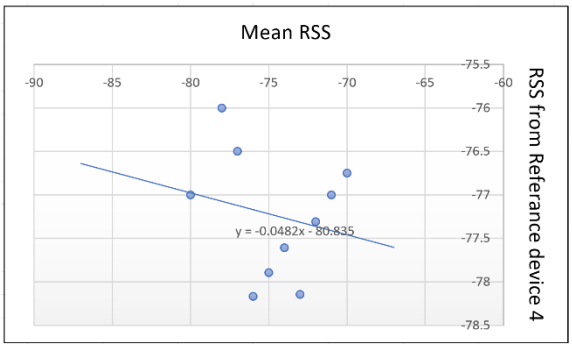


Fig. 7.12 Multivariate Linear Regression Analysis: Channel 2412 between Device 4 and Device 2

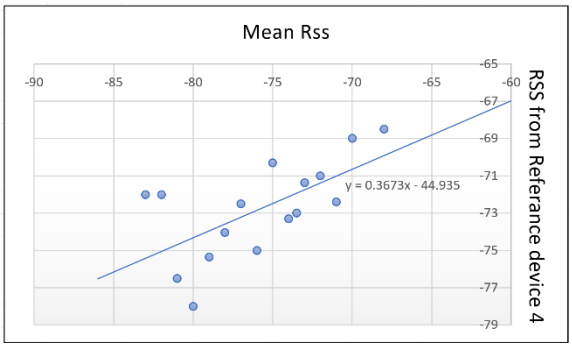


Fig. 7.15 Multivariate Linear Regression Analysis: Channel 2422 between Device 4 and Device 1

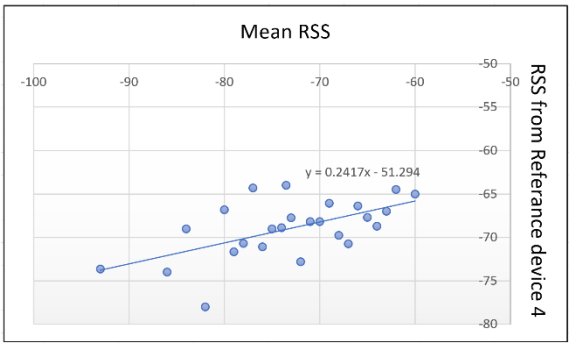


Fig. 7.13 Multivariate Linear Regression Analysis: Channel 2417 between Device 4 and Device 1

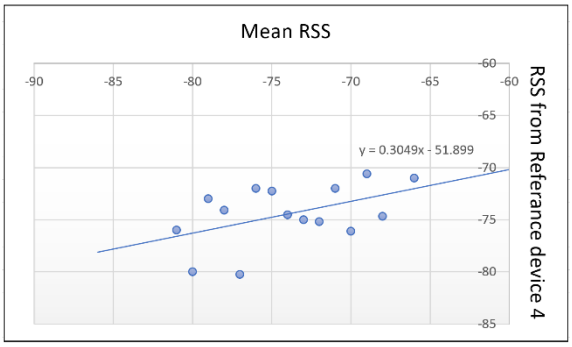


Fig. 7.16 Multivariate Linear Regression Analysis: Channel 2422 between Device 4 and Device 2

MULTIVARIATE LINEAR REGRESSION ANALYSIS FROM ACCESS POINT 3 USING DEVICE 4 AS THE REFERENCE DEVICE

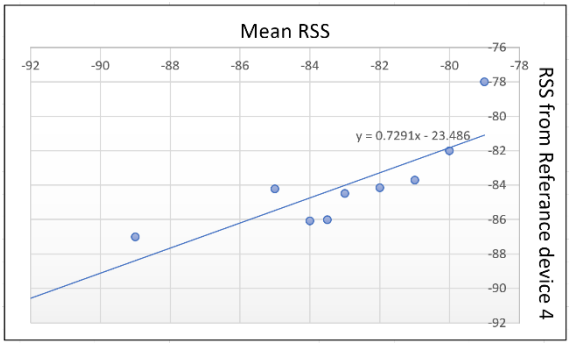


Fig. 7.17 Multivariate Linear Regression Analysis: Channel 2412 between Device 4 and Device 1

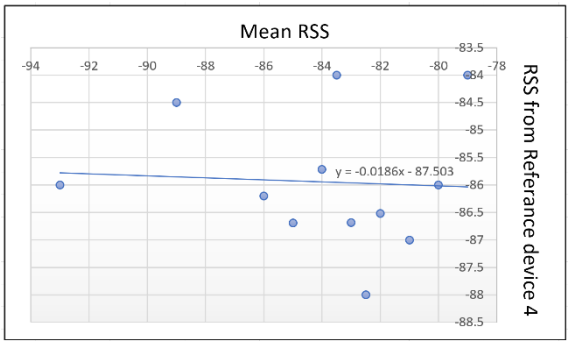


Fig. 7.18 Multivariate Linear Regression Analysis: Channel 2412 between Device 4 and Device 2

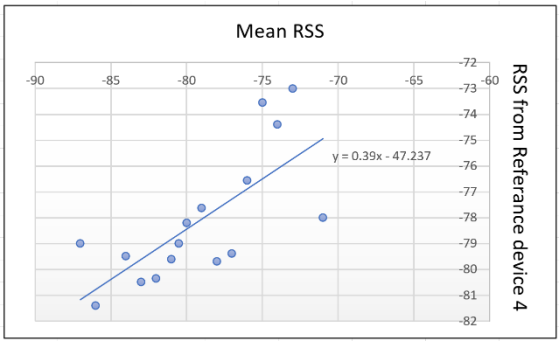


Fig. 7.19 Multivariate Linear Regression Analysis: Channel 2417 between Device 4 and Device 1

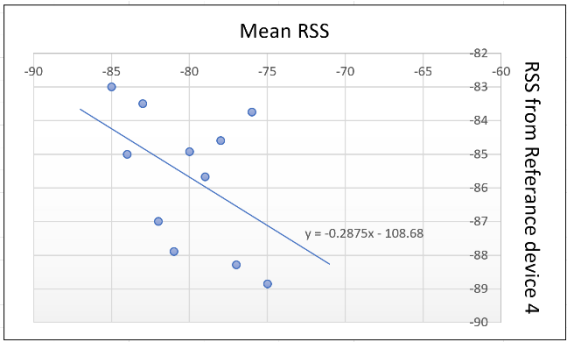


Fig. 7.20 Multivariate Linear Regression Analysis: Channel 2417 between Device 4 and Device 2

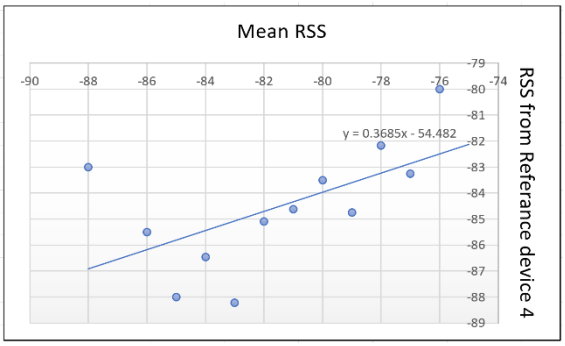


Fig. 7.21 Multivariate Linear Regression Analysis: Channel 2422 between Device 4 and Device 1

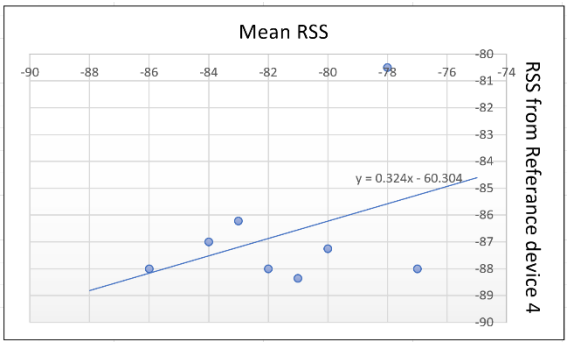


Fig. 7.22 Multivariate Linear Regression Analysis: Channel 2422 between Device 4 and Device 2

MULTIVARIATE LINEAR REGRESSION ANALYSIS FROM ACCESS POINT 1 USING DEVICE 6 AS THE REFERENCE DEVICE

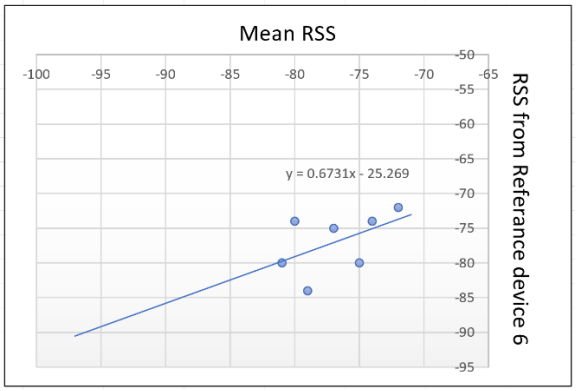


Fig. 7.23 Multivariate Linear Regression Analysis: Channel 2427 between Device 6 and Device 1

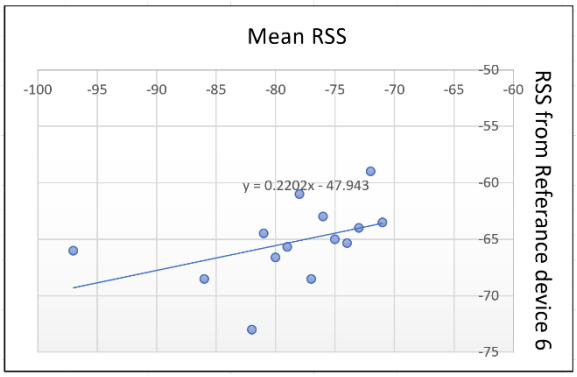


Fig. 7.24 Multivariate Linear Regression Analysis: Channel 2427 between Device 6 and Device 2

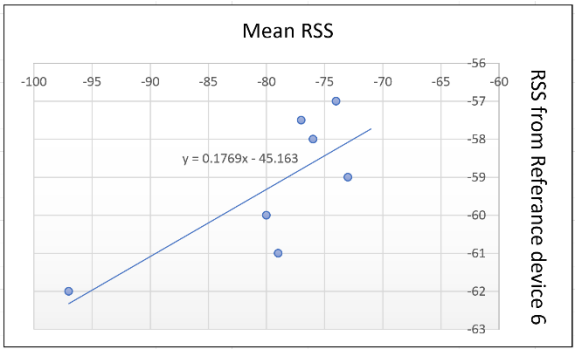


Fig. 7.25 Multivariate Linear Regression Analysis: Channel 2427 between Device 6 and Device 3

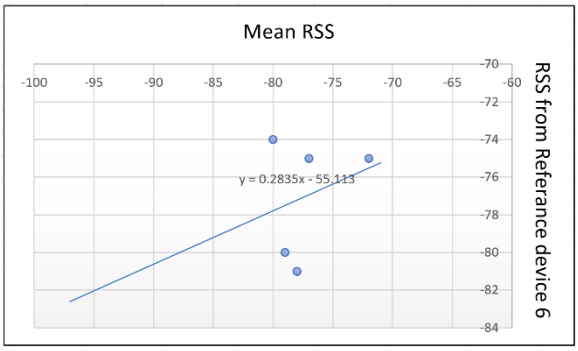


Fig. 7.26 Multivariate Linear Regression Analysis: Channel 2427 between Device 6 and Device 4

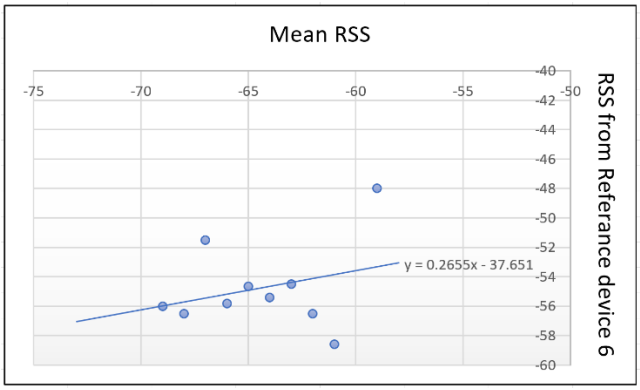


Fig. 7.27 Multivariate Linear Regression Analysis: Channel 2432 between Device 6 and Device 3

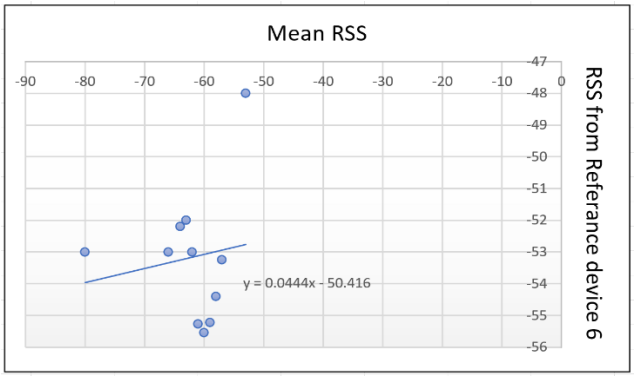


Fig. 7.28 Multivariate Linear Regression Analysis: Channel 2437 between Device 6 and Device 3

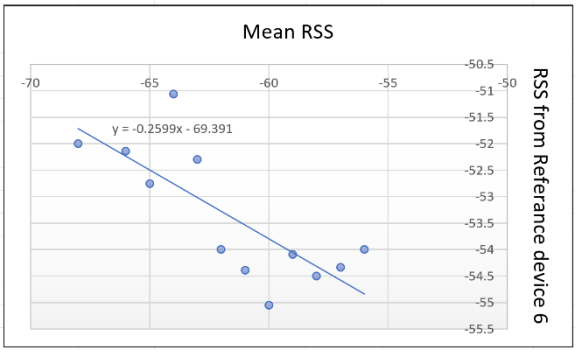


Fig. 7.29 Multivariate Linear Regression Analysis: Channel 2442 between Device 6 and Device 3

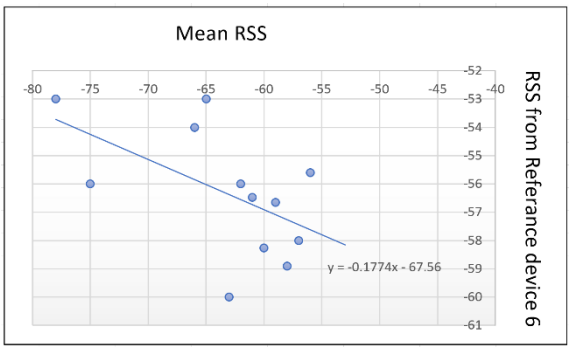


Fig. 7.30 Multivariate Linear Regression Analysis: Channel 2447 between Device 6 and Device 3

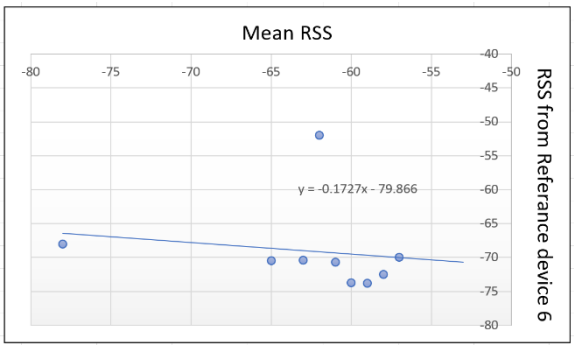


Fig. 7.31 Multivariate Linear Regression Analysis: Channel 2447 between Device 6 and Device 5

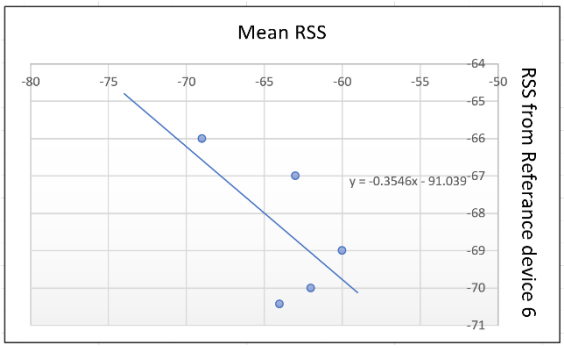


Fig. 7.32 Multivariate Linear Regression Analysis: Channel 2452 between Device 6 and Device 3

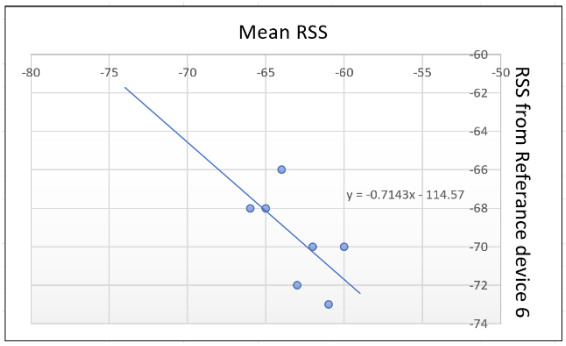


Fig. 7.33 Multivariate Linear Regression Analysis: Channel 2452 between Device 6 and Device 5

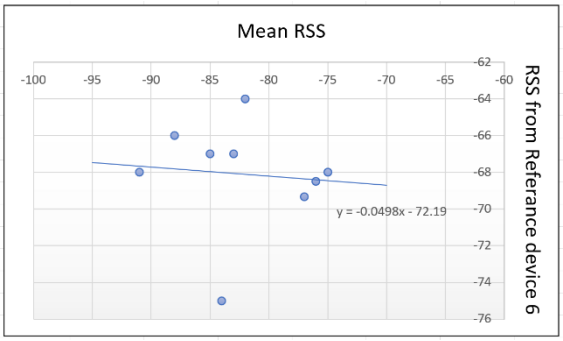


Fig. 7.34 Multivariate Linear Regression Analysis: Channel 2457 between Device 6 and Device 5

MULTIVARIATE LINEAR REGRESSION ANALYSIS FROM ACCESS POINT 2 USING DEVICE 6 AS THE REFERENCE DEVICE

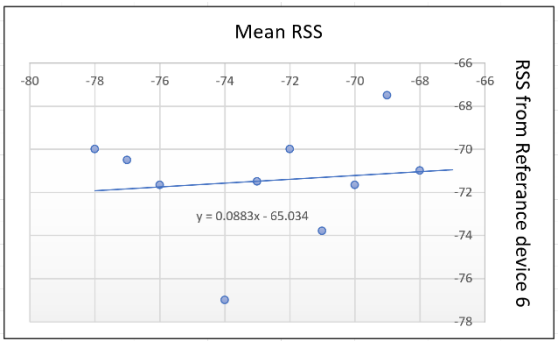


Fig. 7.35 Multivariate Linear Regression Analysis: Channel 2432 between Device 6 and Device 3

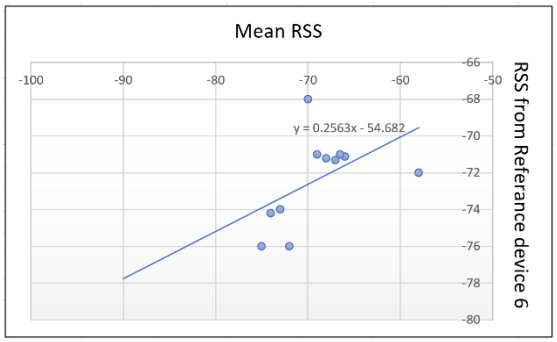


Fig. 7.37 Multivariate Linear Regression Analysis: Channel 2442between Device 6 and Device 3

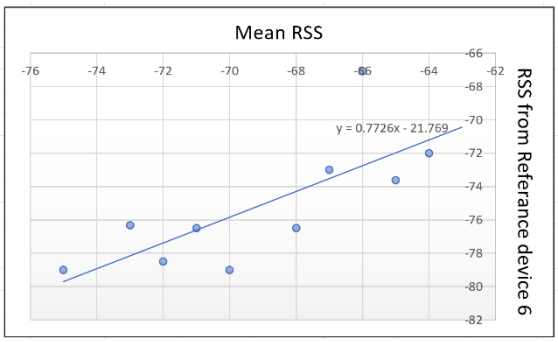


Fig. 7.36 Multivariate Linear Regression Analysis: Channel 2437 between Device 6 and Device 3

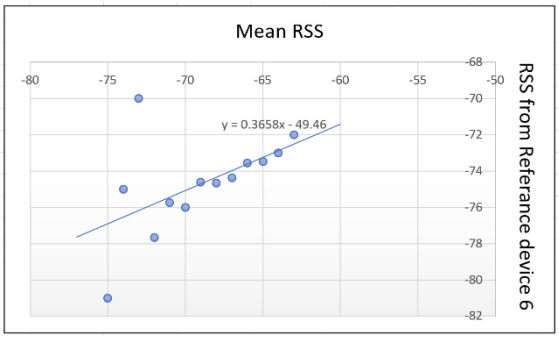


Fig. 7.38 Multivariate Linear Regression Analysis: Channel 2447 between Device 6 and Device 3

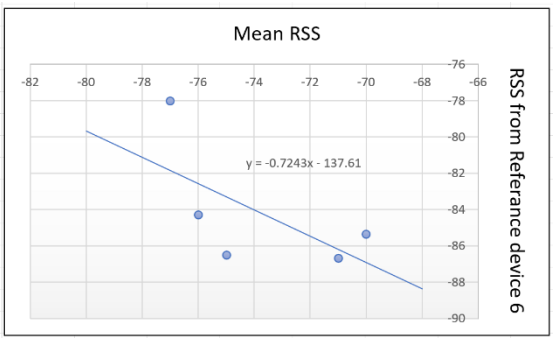


Fig. 7.39 Multivariate Linear Regression Analysis: Channel 2452 between Device 6 and Device 5

MULTIVARIATE LINEAR REGRESSION ANALYSIS FROM ACCESS POINT 3 USING DEVICE 6 AS THE REFERENCE DEVICE

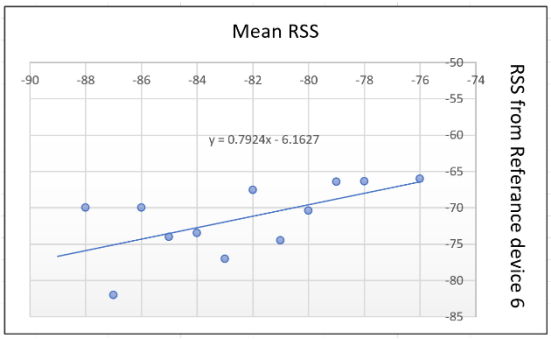


Fig. 7.40 Multivariate Linear Regression Analysis: Channel 2432 between Device 6 and Device 3

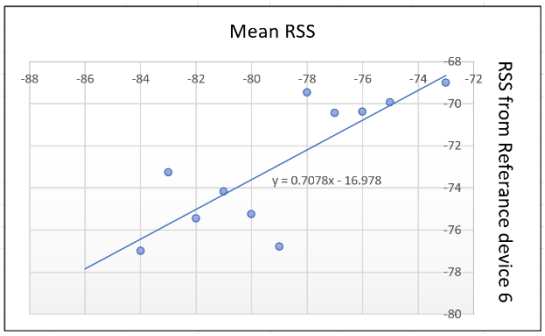


Fig. 7.41 Multivariate Linear Regression Analysis: Channel 2437 between Device 6 and Device 3

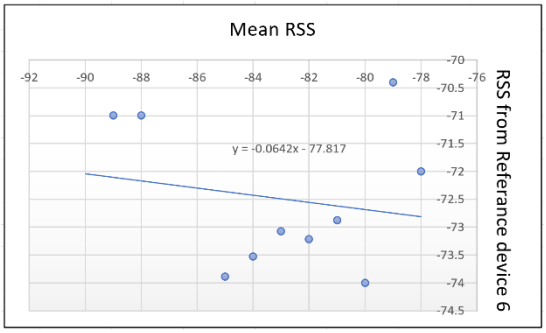


Fig. 7.42 Multivariate Linear Regression Analysis: Channel 2442 between Device 6 and Device 3

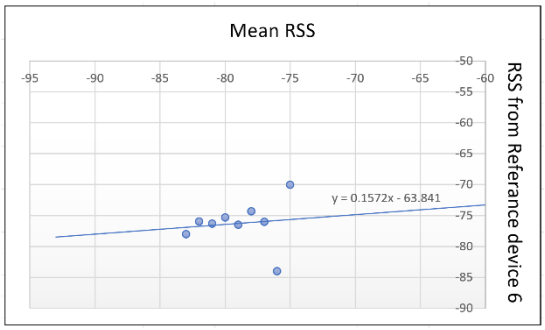


Fig. 7.43 Multivariate Linear Regression Analysis: Channel 2447 between Device 6 and Device 3

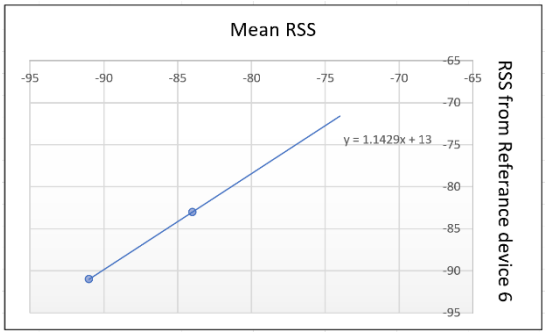


Fig. 7.44 Multivariate Linear Regression Analysis: Channel 2452 between Device 6 and Device 5

APPENDIX H

Source Code

Arduino Code for Scanning UHF RFID Tag

```
//***************************** RFID DATABASING **********************
//********************************** ESP8266 ****************************

#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <SoftwareSerial.h>
#include <ESP8266WebServer.h>
#include <SPI.h>
#include <string.h>

WiFiClient client;
//*****************************************************************************
const int LEDgreen = D3;
const int LEDred = D4;
//*****************************************************************************
unsigned char incomingbyte;
String CardID = "";

SoftwareSerial RFID1 (D5,D6); //232_TX,232_RX Serial connection between RS232 shield and microcontroller

//*****************************************************************************

/* Set these to your desired credentials. */
const char *ssid = "AP4";
const char *password = "engineering02";
const char* device_token  = "e33ecebe831f152a";

//*****************************************************************************

String URL = "http://192.168.201.248/rfid_ips/getdata.php"; //computer IP or the server domain
String getData, Link;
String OldCardID = "";
unsigned long previousMillis = 0;
int n = 1; // new ID
String RFIDcode[100], code = "";

//*****************************************************************************
void setup() {
  delay(1000);
  Serial.begin(9600);

  while (!Serial) {
        ; // wait for serial port to connect.
    }

  RFID1.begin(9600);
  connectToWiFi();
}
//*****************************************************************************
void loop() {

  //check if there's a connection to Wi-Fi or not
  if(!WiFi.isConnected()){
    connectToWiFi();    //Retry to connect to Wi-Fi
  }
  //---------------------------------------------

  if (millis() - previousMillis >= 15000) {
    previousMillis = millis();
  }

  delay(50);

  A:

  while(RFID1.available ()> 0)
  {
    incomingbyte = RFID1.read();
    //Serial.print(incomingbyte,HEX);
    code = code + String(incomingbyte, HEX);

    if(code.length() == 30) // HEX = 22 , DEC = 32
    {
```

```
    String checker = String(code.charAt(0)) + String(code.charAt(1)) + String(code.charAt(2)) + String(code.charAt(3)) +
String(code.charAt(4)) + String(code.charAt(5));
    //Serial.println(checker);

    if(checker != "303132")
    {
      Serial.println(code);
      Serial.println("error RFID card ID reloading");
      Serial.println();
        while(RFID1.available ()> 0) {
          code = code + String(incomingbyte, HEX);
          code = "";
          break;
        }
      code = "";
      n = 2;
      goto A;
      break;
    }

    RFIDcode[n] = code;
    code = "";

    if(n == 2)
      {
      if(RFIDcode[1] == RFIDcode[2]) //look for new card
      {
          break;
      }
      else
        {
          Serial.println();

          RFIDcode[2] == RFIDcode[1];
          RFIDcode[2] = "";
          n = 1;

          break;
        }
      }

    else if(n == 1) // New Card Present
    {
      //Serial.print(RFIDcode[1]);
      Serial.println();

      Serial.print("CARDID FIRST PART: ");
      Serial.println(RFIDcode[1]);
      SendCardID(RFIDcode[1]);

      //delay(2000);
      n++;
    }

   }
  }
  code = "";

}

//************send the Card UID to the website*************
void SendCardID(String Card_uid){

  Serial.println("Sending the Card ID");
  if(WiFi.isConnected())
  {
    HTTPClient http;    //Declare object of class HTTPClient

   //GET Data
    getData = "?card_uid=" + String(Card_uid) + "&device_token=" + String(device_token); // Add the Card ID to the GET array in
order to send it
    Serial.println(getData);
    //GET methode
    Link = URL + getData;
```

```
    http.begin(client,Link); //initiate HTTP request   //Specify content-type header

    int httpCode = http.GET();   //Send the request
    String payload = http.getString();    //Get the response payload

    if (httpCode == 200)
    {


      if (payload.substring(0, 5) == "login")
      {
        String user_name = payload.substring(5);
        Serial.println(user_name);
      }
  else if (payload.substring(0, 6) == "logout")
      {
        String user_name = payload.substring(6);
        Serial.println(user_name);
      }

      else
      {
        String user_name = payload.substring(5);
        Serial.println(user_name);
        Serial.println("Not registered");
        Serial.println(user_name);
      }

      http.end();  //Close connection

    }
  }
}

//********************connect to the WiFi******************
void connectToWiFi(){
   WiFi.mode(WIFI_OFF);        //Prevents reconnection issue (taking too long to connect)
   delay(1000);
   WiFi.mode(WIFI_STA);
   Serial.print("Connecting to ");
   Serial.println(ssid);
   WiFi.begin(ssid, password);

   while (WiFi.status() != WL_CONNECTED) {
     delay(500);
     Serial.print(".");
   }
   Serial.println("");
   Serial.println("Connected");

   Serial.print("IP address: ");
   Serial.println(WiFi.localIP());  //IP address assigned to your ESP

   delay(1000);
}
```

Python Code for Scanning RSS Values from the Access Points

```
import paramiko
import re
import pandas as pd
import csv
import mysql.connector
from mysql.connector import Error
from datetime import datetime

# Define the MikroTik router's IP address, username, and password
router_ip = '[fe80::6e3b:6bff:fe5a:e788%5]'
username = 'thesis2.0'
password = 'admin'

# Database configuration
db_config = {
    "host": "localhost",
    "user": "root",
    "password": "",
    "database": "rfid_ips"
}

# List of specific MAC addresses to filter
specific_ssid = [
                "Vivo_Y21T",
                "Xiaomipoco_X3",
                "Realme_6Pro",
                "RedmiNote_11",
                "TechnoCamon_20",
                "Xiaomi_10T"
              ]

def connect_to_database():
    try:
        connection = mysql.connector.connect(**db_config)
        if connection.is_connected():
            print("Connected to MySQL database")
            return connection
    except Error as e:
        print(f"Error: {e}")
    return None

def transfer_to_database(data, connection):
    if "already running" not in data:
        if connection:
            cursor = connection.cursor()
            try:
                # Insert data into the database, including the timestamp
                insert_query = "INSERT INTO position_scanner_cap1 (mac_address, ssid, channel, signal_strength, source, timestamp)
VALUES (%s, %s, %s, %s, %s, %s)"
                current_timestamp = datetime.now()  # Capture the current timestamp
                data_with_timestamp = [(row[0], row[1], row[2], row[3], row[4], current_timestamp) for row in data]
                cursor.executemany(insert_query, data_with_timestamp)
                connection.commit()
                print(f"Transferred {len(data)} records to the database with timestamp {current_timestamp}.")
            except Error as e:
                connection.rollback()
                print(f"Error transferring data to the database: {e}")
            finally:
                cursor.close()

def extract_numeric_channel(channel):
    # Use regular expression to extract numeric part from the channel
    match = re.match(r'(\d+)', str(channel))  # Ensure the result is always a string
    if match:
        return match.group(1)
    else:
        return None

def main():

    ssh = paramiko.SSHClient()
    ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
```

```
ssh.connect(router_ip, username=username, password=password)

# Create the command
command = f'/caps-man/interface/scan cap1'

# Execute the command
stdin, stdout, stderr = ssh.exec_command(command)

while True:
    try:

        # Define the CAP interface names ('cap1' and 'cap1')
        cap_interfaces = ['cap1']

        # Create a list to store data from both CAP interfaces
        data = []

        pattern = r'(\S+)\s+(?:(\S+)\s+)?(\d+/\d+/\w+).+?(-\d+)'

        for cap_interface in cap_interfaces:

            output = stdout.read(3000).decode()
            lines = output.splitlines()

            #print(f"unprocessed data {cap_interface}:", output)
            #print(f"lines {cap_interface}:", lines)

            if "failure: already running" not in output:
                # Data collection and processing for the current CAP interface
                current_data = []

                latest_results = {}

                for line in lines:
                    match = re.search(pattern, line)
                    #print(f"line: {line}")
                    if match:
                        mac_address = match.group(1)
                        ssid = match.group(2)
                        channel = match.group(3)  # Capture the channel as a string
                        signal_strength = int(match.group(4))  # Capture the signal strength as an integer
                        #print(f"ssid output: {ssid}")
                        if ssid in specific_ssid:
                            current_timestamp = datetime.now()  # Capture the current timestamp
                            latest_results[ssid] = (
                                mac_address, ssid, channel, signal_strength, cap_interface, current_timestamp)
                # Wait for the scan to complete, adjust the sleep time as needed
                #time.sleep(1)  # You can adjust the sleep duration

                # If specific_ssid is not in the data, add a default entry
                for ssid in specific_ssid:
                    if ssid not in latest_results:
                        default_entry = ('0', ssid, '0', 100, cap_interface, datetime.now())
                        latest_results[ssid] = default_entry

                current_data.extend(latest_results.values())
                data.extend(current_data)

                print(f"data {cap_interface}:", current_data)



        if "failure: already running" not in output:  # Check the condition here as well

            # Connect to the database
            connection = connect_to_database()

            # Transfer data to the database with the floorid and timestamp
            if connection:
                transfer_to_database(data, connection)

                # Extract data to the database
                # Replace 'your_query' with the SQL query to select the data you want
                your_query = "SELECT * FROM `position_scanner_cap1`"
```

```
# Use pandas to read the query result into a DataFrame
df = pd.read_sql_query(your_query, connection)

# Replace 'output_file.csv' with the desired file name
position_scanner_cap1 = 'position_scanner_cap1.csv'

# Save the DataFrame to a CSV file
df.to_csv(position_scanner_cap1, index=False)

print(f"Data extracted and saved to {position_scanner_cap1}")

# Read the CSV file into a new DataFrame
position_scanner_cap1 = pd.read_csv(position_scanner_cap1)

print('position_scanner_cap1', position_scanner_cap1)

# Check for three similar rows based on specific columns
columns_to_check = ['mac_address', 'ssid', 'channel', 'signal_strength']

duplicates = position_scanner_cap1[
    position_scanner_cap1.duplicated(subset=columns_to_check, keep=False)]

if not duplicates.empty:
    # Group by 'ssid', 'channel', and 'signal_strength' columns and count the number of duplicate rows for each group
    duplicate_counts = position_scanner_cap1[
        position_scanner_cap1.duplicated(subset=columns_to_check, keep=False)].groupby(
        ['ssid', 'channel', 'signal_strength']).size()

    # Print the results and transfer data for groups with duplicate_count >= 4
    for (ssid, channel, signal_strength), count in duplicate_counts.items():

        # Print the total count for each SSID before processing the data
        total_count = position_scanner_cap1[position_scanner_cap1['ssid'] == ssid].shape[0]
        print(f"SSID: {ssid}, Total Count: {total_count}, Number of Duplicate Rows: {count}")

        print(
            f"SSID: {ssid}, Channel: {channel}, Signal Strength: {signal_strength}, Number of Duplicate Rows: {count}")

        # Check if duplicate_count is 4 or more
        if count >= 5:
            # Extract the data for the current group
            group_data = position_scanner_cap1[
                (position_scanner_cap1['ssid'] == ssid) &
                (position_scanner_cap1['channel'] == channel) &
                (position_scanner_cap1['signal_strength'] == signal_strength)
                ]

            # Select the row with the maximum signal strength
            #representative_row = group_data.loc[group_data['signal_strength'].idxmax()]

            # Save the representative row to "scanned_aps_cap1.csv"
            #representative_row.to_csv("scanned_aps_cap1.csv", index=False, mode='a', header=False)

            # Create and open a CSV file for writing
            with open('scanned_aps_cap1.csv', 'w', newline='') as csv_file:
                fieldnames = ['mac_address', 'ssid', 'channel', 'signal_strength', 'source',
                              'timestamp']  # mac_address, ssid, signal_strength, channel, source, timestamp
                csv_writer = csv.DictWriter(csv_file, fieldnames=fieldnames)
                csv_writer.writeheader()

                # Write the data to the CSV file
                for row in data:
                    numeric_channel = extract_numeric_channel(row[2])
                    # Convert the timestamp to a string for CSV
                    timestamp_str = row[5].strftime('%Y-%m-%d %H:%M:%S')
                    csv_writer.writerow(
                        {'mac_address': row[0], 'ssid': row[1], 'channel': numeric_channel,
                         'signal_strength': row[3], 'source': row[4],
                         'timestamp': timestamp_str})

            print(
                f"Data transferred to scanned_aps_cap1.csv for SSID: {ssid}, Channel: {channel}, Signal Strength: {signal_strength}")

            # Connect to the database
```

```
                    with connect_to_database() as connection:
                        delete_data_from_database(ssid, connection)

                if total_count >= 10:
                    print(f"Total count for SSID {ssid} is 20 or more. Performing actions...")

                    # Connect to the database
                    with connect_to_database() as connection:
                        delete_data_from_database(ssid, connection)

            else:
                print("No rows with similar values found.")

    except paramiko.AuthenticationException:
        print("Authentication failed. Please check your credentials.")
    except paramiko.SSHException as e:
        print(f"SSH connection failed: {str(e)}")
    except Exception as e:
        print(f"An unexpected error occurred: {str(e)}")

def delete_data_from_database(ssid, connection):
    if connection:
        try:
            with connection.cursor() as cursor:
                delete_query = "DELETE FROM position_scanner_cap1 WHERE ssid = %s"
                cursor.execute(delete_query, (ssid,))
            connection.commit()
            print(f"All data with SSID {ssid} deleted from the database.")
        except Error as e:
            connection.rollback()
            print(f"Error deleting data from the database: {e}")

if __name__ == "__main__":
    main()
```

Python Code for Offline Training Phase

```
import numpy as np
import pandas as pd
import time
from itertools import product
import math

#Preprocessing
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.decomposition import PCA
from scipy.sparse import lil_matrix

#Models
from sklearn.naive_bayes import GaussianNB
from skmultilearn.problem_transform import BinaryRelevance
from skmultilearn.problem_transform import ClassifierChain
from skmultilearn.problem_transform import LabelPowerset
from skmultilearn.adapt import MLkNN
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

#Scoring Metrics
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import accuracy_score

#*****************************************************************************

# Load the dataset
file_path = r'C:\Users\Thesis2.0\django_thesis\KNN Algorithm\ap_data_8r.csv'
ap_data = pd.read_csv(file_path)

# Define the columns for duplication check
duplication_columns = ['ssid', 'channel', 'source', 'signal_strength', 'floorid']

# Count the number of occurrences for each combination of columns
ap_data['duplication_count'] = ap_data.groupby(duplication_columns)['ssid'].transform('count')

#*****************************************************************************
```

```
# Filter rows where duplication count is greater than or equal to 4
filtered_data = ap_data[ap_data['duplication_count'] >= 4]

#****************************************************************************

# Group by duplication columns and aggregate values using, for example, the mean
merged_data = filtered_data.groupby(duplication_columns, as_index=False).mean()

# Drop the 'duplication_count' column if you don't need it in the final result
merged_data = merged_data.drop(columns=['duplication_count'])

# Count unique channels for each ssid, floorid, and source
merge_data = merged_data.groupby(['ssid', 'floorid', 'source', 'roomid'])['channel'].nunique().reset_index()

#****************************************************************************

# Define a custom aggregation function to calculate the product
def calculate_product(x):
    return x.prod()

# Group by 'ssid' and 'floorid', then aggregate using the custom function
result = merge_data.groupby(['ssid', 'floorid']).agg({
    'channel': calculate_product
}).reset_index()

# Continue from the previous code
total_sum_combination = result['channel'].sum()

# Count the number of unique sources per 'ssid' and 'floorid'
result_counts = merge_data.groupby(['ssid', 'floorid'])['source'].nunique().reset_index()

# Rename the column to indicate the count of sources
result_counts = result_counts.rename(columns={'source': 'source_count'})

# Calculate total RSSI for each unique channel within each ssid, floorid, and source
total_number_rssi = merged_data.groupby(['ssid', 'floorid', 'source', 'channel'])['signal_strength'].nunique().reset_index()

#****************************************************************************
```

 # Step 1: Identify missing sources and add rows with 0 channel and 100 signal_strength

```
# List of sources to check for each ssid and floorid combination
sources_to_check = ['cap1', 'cap2', 'cap3']

# Create a MultiIndex from ssid and floorid
index_cols = ['ssid', 'floorid']
multi_index = pd.MultiIndex.from_frame(merged_data[index_cols])

# Iterate over unique ssid and floorid combinations
for ssid, floorid in multi_index.unique():
    # Filter data for the current ssid and floorid
    subset = merged_data[(merged_data['ssid'] == ssid) & (merged_data['floorid'] == floorid)]

    # Get the sources present in the subset
    present_sources = subset['source'].unique()

    # Find the missing sources
    missing_sources = set(sources_to_check) - set(present_sources)

    # Add rows for missing sources
    for missing_source in missing_sources:
        merged_data = merged_data.append({
            'ssid': ssid,
            'channel': 0,
            'source': missing_source,
            'signal_strength': 100,
            'floorid': floorid,
            'roomid': subset['roomid'].iloc[0]  # Assuming roomid is the same for all rows in the subset
        }, ignore_index=True)

# Sort the DataFrame by ssid and floorid
merged_data = merged_data.sort_values(by=['ssid', 'floorid']).reset_index(drop=True)
```

```
# Remove rows where ssid is 0
merged_data = merged_data[merged_data['ssid'] != 0]

# Step 2: Extract unique combinations of 'ssid', 'floorid', and 'roomid'
combinations = merged_data[['ssid', 'floorid', 'roomid']].drop_duplicates()

# Step 3: Calculate unique channels for each source
channel_counts = {}
for index, row in combinations.iterrows():
    ssid, floorid, roomid = row['ssid'], row['floorid'], row['roomid']
    channels = merged_data[(merged_data['ssid'] == ssid) &
                           (merged_data['floorid'] == floorid) &
                           (merged_data['roomid'] == roomid)].groupby('source')['channel'].unique()
    channel_counts[(ssid, floorid, roomid)] = channels

# Step 4 and 5: Generate combinations and create DataFrame
combinations_list = []
for (ssid, floorid, roomid), channels in channel_counts.items():
    for source, channel_list in channels.items():
        for channel in channel_list:
            signal_strengths = merged_data[
                (merged_data['ssid'] == ssid) &
                (merged_data['floorid'] == floorid) &
                (merged_data['roomid'] == roomid) &
                (merged_data['source'] == source) &
                (merged_data['channel'] == channel)
            ]['signal_strength'].tolist()

            total_signal_strengths = len(signal_strengths)
            combination = [ssid, floorid, roomid, source, f'channel_{channel}_signal_strength_{source}', signal_strengths]
            combinations_list.append(combination)

# Step 6: Combine the RSSI from each row
combined_rssi = pd.DataFrame(columns=['ssid', 'floorid', 'roomid', 'channel_cap1', 'channel_cap2', 'channel_cap3',
'signal_strength_cap1', 'signal_strength_cap2', 'signal_strength_cap3'])

for (ssid, floorid, roomid), channels in channel_counts.items():
    product_list = product(*(channels[source] for source in channels.keys()))

    for combination in product_list:
        row = {'ssid': ssid, 'floorid': floorid, 'roomid': merged_data[(merged_data['ssid'] == ssid) & (merged_data['floorid'] ==
floorid)]['roomid'].values[0]}
        for i, (source, channel) in enumerate(zip(channels.keys(), combination), start=1):
            row[f'channel_{source}'] = channel
            signal_strengths = merged_data[
                (merged_data['ssid'] == ssid) &
                (merged_data['floorid'] == floorid) &
                (merged_data['source'] == source) &
                (merged_data['channel'] == channel)
            ]['signal_strength'].tolist()

            row[f'signal_strength_cap{i}'] = signal_strengths if signal_strengths else None

        combined_rssi = combined_rssi.append(row, ignore_index=True)

# Step 7: Combine the RSSI from each row
combined_rssi_final = pd.DataFrame(columns=combined_rssi.columns)  # Initialize an empty DataFrame

for index, row in combined_rssi.head(total_sum_combination).iterrows():  # Iterate only over the first 9 rows
    ssid, floorid, roomid, channel_cap1, channel_cap2, channel_cap3 = row['ssid'], row['floorid'], row['roomid'], row['channel_cap1'],
row['channel_cap2'], row['channel_cap3']

    # Create all possible combinations of signal strengths
    signal_strengths = [row['signal_strength_cap1'], row['signal_strength_cap2'], row['signal_strength_cap3']]

    # Handle the case where signal strength is an integer (100)
    signal_combinations = [signal if isinstance(signal, list) else [signal] for signal in signal_strengths]

    for signal_combination in product(*signal_combinations):
        combined_rssi_final = combined_rssi_final.append({
            'ssid': ssid,
            'floorid': floorid,
            'roomid': roomid,
            'channel_cap1': channel_cap1,
```

```
        'channel_cap2': channel_cap2,
        'channel_cap3': channel_cap3,
        'signal_strength_cap1': signal_combination[0],
        'signal_strength_cap2': signal_combination[1],
        'signal_strength_cap3': signal_combination[2]
    }, ignore_index=True)
```

Python Code for Training Different Position Algorithms

```
# Load the dataset
trainingData = combined_rssi_final

#*****************************************************************************

def clean_data(df):

    # Remove selected columns...
    columns_removed = ['mac_address', 'timestamp']
    for col in columns_removed:
        df.drop(col, axis=1, inplace=True)

    # Fill NaN values with specific values
    fill_values = {
        'channel_cap1': 0,
        'channel_cap2': 0,
        'channel_cap3': 0,
        'signal_strength_cap1': 100,
        'signal_strength_cap2': 100,
        'signal_strength_cap3': 100
    }
    df.fillna(value=fill_values, inplace=True)

    # Create a condition to identify rows where all 'channel_cap' columns are zero
    zero_channel_condition = (df['channel_cap1'] == 0) & \
                             (df['channel_cap2'] == 0) & \
                             (df['channel_cap3'] == 0)

    # Filter the DataFrame to remove rows where the condition is True
    df = df[~zero_channel_condition]

    # Return the cleaned dataframe.
    return df

#Apply Cleaning

trainingData  = clean_data(trainingData)
cleaned_data = trainingData

#*****************************************************************************

# Group the data by "floorid" and count the number of rows for each unique floor ID
floorid_counts = trainingData.groupby('floorid').size()

# Convert the Series to a DataFrame with 'floorid' as index
floorid_counts_df = floorid_counts.reset_index(name='row_count')

# Display the number of rows for each unique floor ID
print("Number of Rows for Each Unique Floor ID:")
print(floorid_counts_df)

#*****************************************************************************

Load the dataset
trainingData = combined_rssi_final

# Calculate the row counts for each floorid
floorid_counts = trainingData.groupby('floorid').size().reset_index(name='row_count')

# Find the maximum number of rows associated with any floorid
max_rows = floorid_counts['row_count'].max()

# Create an empty DataFrame to store duplicated rows
duplicated_rows = pd.DataFrame()
```

```
# Iterate over unique floorids
for floorid, row_count in floorid_counts[['floorid', 'row_count']].values:
    # Calculate the number of duplications required to match the max_rows
    num_duplicates = max_rows - row_count

    # If the current number of rows is less than max_rows, duplicate the rows
    if num_duplicates > 0:
        # Filter rows for the current floorid
        floorid_rows = trainingData[trainingData['floorid'] == floorid]

        # Duplicate rows and append to the duplicated_rows DataFrame
        duplicated_rows = pd.concat([duplicated_rows, floorid_rows.sample(n=num_duplicates, replace=True)])

# Concatenate the original dataset with duplicated rows
balanced_trainingData = pd.concat([trainingData, duplicated_rows])

# Sort the DataFrame by floorid for clarity
balanced_trainingData = balanced_trainingData.sort_values(by='floorid').reset_index(drop=True)

# Group the data by "floorid" and count the number of rows for each unique floor ID
floorid_counts = balanced_trainingData.groupby('floorid').size()

# Convert the Series to a DataFrame with 'floorid' as index
floorid_counts_df = floorid_counts.reset_index(name='row_count')

#*****************************************************************************

def preprocess_data(df):
    """
    Separates trainingData into Features and Targets
    Will also be applied to validationData

    INPUT: Cleaned trainingData DataFrame
    OUTPUT: trainingData as Features and Targets
    """
    # split the data set into features and targets(Floor and BuildingID)
    X = df.drop(['floorid', 'roomid'], axis=1)
    y = df[['roomid']]

    # Extract unique channel values
    unique_channels = sorted(set(df['channel_cap1'].unique()) | set(df['channel_cap2'].unique()) | set(df['channel_cap3'].unique()))

    # Create new one-hot encoded columns
    for channel in unique_channels:
        X[f'channel_cap1_{channel}'] = (df['channel_cap1'] == channel).astype(int)
        X[f'channel_cap2_{channel}'] = (df['channel_cap2'] == channel).astype(int)
        X[f'channel_cap3_{channel}'] = (df['channel_cap3'] == channel).astype(int)

    # Drop the original 'channel_cap1', 'channel_cap2', and 'channel_cap3' columns
    X.drop(['channel_cap1', 'channel_cap2', 'channel_cap3'], axis=1, inplace=True)

    # Iterate over signal strength caps and channels to perform multiplication
    signal_columns = ['signal_strength_cap1', 'signal_strength_cap2', 'signal_strength_cap3']

    for signal_col in signal_columns:
        for channel in unique_channels:
            channel_col1 = f'channel_cap1_{channel}'
            channel_col2 = f'channel_cap2_{channel}'
            channel_col3 = f'channel_cap3_{channel}'

            if signal_col.endswith('cap1'):
                X[f'{signal_col}_{channel_col1}'] = df[signal_col] * X[channel_col1]
            elif signal_col.endswith('cap2'):
                X[f'{signal_col}_{channel_col2}'] = df[signal_col] * X[channel_col2]
            elif signal_col.endswith('cap3'):
                X[f'{signal_col}_{channel_col3}'] = df[signal_col] * X[channel_col3]

    # Drop the original 'signal_strength' columns
    X.drop(['signal_strength_cap1', 'signal_strength_cap2', 'signal_strength_cap3'], axis=1, inplace=True)

    # Drop unwanted columns
    unwanted_columns = [f'channel_cap{i}_{cap}' for i in range(1, 4) for cap in unique_channels]
    X.drop(unwanted_columns, axis=1, inplace=True)

    # create Dummies for the targets to feed into the model
```

```
    y = pd.get_dummies(data=y, columns=['roomid'])

    return X, y

#Apply preprocessing

X, y = preprocess_data(balanced_trainingData)

#*****************************************************************************
def split_data(preprocess_data):
# TO AVOID OVERFITTING: Split the training data into training and testing sets
    global X_train
    global X_test
    global y_train
    global y_test

    X_train, X_test, y_train, y_test = train_test_split(X,
                                        y,
                                        test_size = 0.20,
                                        random_state = 42,
                                        shuffle=True)

    # Show the results of the split
    print("Training set has {} samples.".format(X_train.shape[0]))
    print("Testing set has {} samples.".format(X_test.shape[0]))
    return X_train, X_test, y_train, y_test

#Apply split data
X_train, X_test, y_train, y_test = split_data(preprocess_data)

#*****************************************************************************
#KNN Algorithm

start_time = time.time()

#Scale Data with Standard Scaler

scaler = StandardScaler()

#Fit only the training set
#this will help us transform the validation data
scaler.fit(X_train)

# Apply transform to both the training set and the test set.
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)


k = 1
knn = KNeighborsClassifier(n_neighbors=k, p=1)  # p=2 for Euclidean metric
knn.fit(X_train, y_train)

# Assuming 'knn' is your trained KNN classifier
# Assuming 'X_test' and 'y_test' are your test features and labels
y_pred = knn.predict(X_test)

# Calculate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100:.2f}%')

print("--- Run time: %s mins ---" % np.round(((time.time() - start_time)/60),2))

#*****************************************************************************
#Apply PCA while keeping 95% of the variation in the data
pca = PCA(.95)

#Fit only the training set
pca.fit(X_train)

# Apply PCA transform to both the training set and the test set.
X_train_pca = pca.transform(X_train)
X_test_pca = pca.transform(X_test)
```

```
#Create sparse matrices to run the scikit multilearn algorithms
X_train_pca = lil_matrix(X_train_pca).toarray()
y_train = lil_matrix(y_train).toarray()
X_test_pca = lil_matrix(X_test_pca).toarray()
y_test = lil_matrix(y_test).toarray()

#Binary Relevance Multi-Label Classifier
start_time = time.time()

BR_classifier = BinaryRelevance(GaussianNB())
# train
BR_classifier.fit(X_train, y_train)
# predict
predictions = BR_classifier.predict(X_test)
# accuracy
print("Accuracy = ",accuracy_score(y_test,predictions))

print("--- Run time: %s mins ---" % np.round(((time.time() - start_time)/60),2))

#Classifier Chains Multi-Label Classifier
start_time = time.time()

CC_classifier = ClassifierChain(GaussianNB())

# train
CC_classifier.fit(X_train, y_train)

# predict
predictions = CC_classifier.predict(X_test)

accuracy_score(y_test,predictions)
print('Test accuracy is {}'.format(accuracy_score(y_test, predictions)))

print("--- Run time: %s mins ---" % np.round(((time.time() - start_time)/60),2))

#Label Powerset multi-label classifier
start_time = time.time()

LP_classifier = LabelPowerset(GaussianNB())

# train
LP_classifier.fit(X_train, y_train)

# predict
predictions = LP_classifier.predict(X_test)

accuracy_score(y_test,predictions)
print('Test accuracy is {}'.format(accuracy_score(y_test, predictions)))

print("--- Run time: %s mins ---" % np.round(((time.time() - start_time)/60),2))

#MLKNN

start_time = time.time()

MLKNN_classifier = MLkNN(k=1)

# train
MLKNN_classifier.fit(X_train_pca, y_train)

# predict mlknn = 1
predictions = MLKNN_classifier.predict(X_test_pca)

print(predictions)

# accuracy
print("Accuracy = ",accuracy_score(y_test,predictions))

print("--- Run time: %s mins ---" % np.round(((time.time() - start_time)/60),2))

#*****************************************************************************
```

#Hyperparameter Tuning

```
parameters = {'k': range(1,100), 's': [0.5, 0.7, 1.0]}

score = 'accuracy'

clf = GridSearchCV(MLkNN(), parameters, scoring=score, cv=5)
clf.fit(X_train_pca, y_train)

print (clf.best_params_, clf.best_score_)
```

Python Code for Online Determination Phase

```
# necessary Libraries
import numpy as np
import pandas as pd
import time
from datetime import datetime
from itertools import product

# Preprocessing
from sklearn.preprocessing import StandardScaler

# Models
from sklearn.neighbors import KNeighborsClassifier

import os
import mysql.connector
from mysql.connector import Error  # Import the Error module

#*****************************************************************************

# Database configuration
db_config = {
    "host": "localhost",
    "user": "root",
    "password": "",
    "database": "rfid_ips"
}

# Function to connect to the MySQL database
def connect_to_database():
    try:
        connection = mysql.connector.connect(**db_config)
        if connection.is_connected():
            print("Connected to MySQL database")
            return connection
    except Error as e:
        print(f"Error: {e}")
    return None


# Function to save DataFrame to CSV, deleting the existing file if it exists
def save_dataframe_to_csv(df, file_path):
    if os.path.exists(file_path):
        os.remove(file_path)
        print(f"Previous file '{file_path}' deleted.")

    df.to_csv(file_path, index=False)
    print(f"Data extracted and saved to {file_path}")


def clean_data(df):
    # Fill NaN values with specific values
    fill_values = {
        'channel_cap1': 0,
        'channel_cap2': 0,
        'channel_cap3': 0,
        'signal_strength_cap1': 100,
        'signal_strength_cap2': 100,
        'signal_strength_cap3': 100
    }
    df.fillna(value=fill_values, inplace=True)
```

```
    # Create a condition to identify rows where all 'channel_cap' columns are zero
    zero_channel_condition = (df['channel_cap1'] == 0) & \
                             (df['channel_cap2'] == 0) & \
                             (df['channel_cap3'] == 0)

    # Filter the DataFrame to remove rows where the condition is True
    df = df[~zero_channel_condition]

    # Return the cleaned dataframe.
    return df

def preprocess_data(df):

    X = df

    # Extract unique channel values
    unique_channels = sorted(
        set(df['channel_cap1'].unique()) | set(df['channel_cap2'].unique()) | set(df['channel_cap3'].unique()))

    # Create new one-hot encoded columns
    for channel in unique_channels:
        X[f'channel_cap1_{channel}'] = (df['channel_cap1'] == channel).astype(int)
        X[f'channel_cap2_{channel}'] = (df['channel_cap2'] == channel).astype(int)
        X[f'channel_cap3_{channel}'] = (df['channel_cap3'] == channel).astype(int)

    # Drop the original 'channel_cap1', 'channel_cap2', and 'channel_cap3' columns
    X.drop(['channel_cap1', 'channel_cap2', 'channel_cap3'], axis=1, inplace=True)

    # Iterate over signal strength caps and channels to perform multiplication
    signal_columns = ['signal_strength_cap1', 'signal_strength_cap2', 'signal_strength_cap3']

    for signal_col in signal_columns:
        for channel in unique_channels:
            channel_col1 = f'channel_cap1_{channel}'
            channel_col2 = f'channel_cap2_{channel}'
            channel_col3 = f'channel_cap3_{channel}'

            if signal_col.endswith('cap1'):
                X[f'{signal_col}_{channel_col1}'] = df[signal_col] * X[channel_col1]
            elif signal_col.endswith('cap2'):
                X[f'{signal_col}_{channel_col2}'] = df[signal_col] * X[channel_col2]
            elif signal_col.endswith('cap3'):
                X[f'{signal_col}_{channel_col3}'] = df[signal_col] * X[channel_col3]

    # Drop the original 'signal_strength' columns
    X.drop(['signal_strength_cap1', 'signal_strength_cap2', 'signal_strength_cap3'], axis=1, inplace=True)

    # Drop unwanted columns
    unwanted_columns = [f'channel_cap{i}_{cap}' for i in range(1, 4) for cap in unique_channels]
    X.drop(unwanted_columns, axis=1, inplace=True)

    return X

def preprocess_data_dataset(df):
    # split the data set into features and targets(Floor and BuildingID)
    X = df.drop(['floorid', 'roomid'], axis=1)
    y = df[['floorid']]

    # Extract unique channel values
    unique_channels = sorted(
        set(df['channel_cap1'].unique()) | set(df['channel_cap2'].unique()) | set(df['channel_cap3'].unique()))

    # Create new one-hot encoded columns
    for channel in unique_channels:
        X[f'channel_cap1_{channel}'] = (df['channel_cap1'] == channel).astype(int)
        X[f'channel_cap2_{channel}'] = (df['channel_cap2'] == channel).astype(int)
        X[f'channel_cap3_{channel}'] = (df['channel_cap3'] == channel).astype(int)

    # Drop the original 'channel_cap1', 'channel_cap2', and 'channel_cap3' columns
    X.drop(['channel_cap1', 'channel_cap2', 'channel_cap3'], axis=1, inplace=True)

    # Iterate over signal strength caps and channels to perform multiplication
    signal_columns = ['signal_strength_cap1', 'signal_strength_cap2', 'signal_strength_cap3']

    for signal_col in signal_columns:
```

```
        for channel in unique_channels:
            channel_col1 = f'channel_cap1_{channel}'
            channel_col2 = f'channel_cap2_{channel}'
            channel_col3 = f'channel_cap3_{channel}'

            if signal_col.endswith('cap1'):
                X[f'{signal_col}_{channel_col1}'] = df[signal_col] * X[channel_col1]
            elif signal_col.endswith('cap2'):
                X[f'{signal_col}_{channel_col2}'] = df[signal_col] * X[channel_col2]
            elif signal_col.endswith('cap3'):
                X[f'{signal_col}_{channel_col3}'] = df[signal_col] * X[channel_col3]

    # Drop the original 'signal_strength' columns
    X.drop(['signal_strength_cap1', 'signal_strength_cap2', 'signal_strength_cap3'], axis=1, inplace=True)

    # Drop unwanted columns
    unwanted_columns = [f'channel_cap{i}_{cap}' for i in range(1, 4) for cap in unique_channels]
    X.drop(unwanted_columns, axis=1, inplace=True)

    # create Dummies for the targets to feed into the model
    y = pd.get_dummies(data=y, columns=['floorid'])

    return X, y

def add_unique_channels(X_processed, X_train):
    # Get the columns of X_train
    X_train_cols = X_train.columns

    # Add missing columns from X_train to X_processed with a default value
    for col in X_train_cols:
        if col not in X_processed.columns:
            X_processed[col] = 0  # Fill missing columns with default value

    # Rearrange columns to match X_train
    X_processed = X_processed[X_train_cols]

    return X_processed

# Main function
def main():

    # Load the dataset
    #file_path = r'C:\Users\Thesis2.0\django_thesis\KNN Algorithm\ap_data_processed.csv'
    #trainingData = pd.read_csv(file_path)


    while True:  # Run forever

    # SCANNED DATA*********************************************************************************

        # Load scanned data
        scanned_data_cap1 = r'C:\Users\Thesis2.0\django_thesis\Position Scanner\scanned_aps_cap1.csv'
        scanned_data_cap2 = r'C:\Users\Thesis2.0\django_thesis\Position Scanner\scanned_aps_cap2.csv'
        scanned_data_cap3 = r'C:\Users\Thesis2.0\django_thesis\Position Scanner\scanned_aps_cap3.csv'

        data_cap1 = pd.read_csv(scanned_data_cap1)
        data_cap2 = pd.read_csv(scanned_data_cap2)
        data_cap3 = pd.read_csv(scanned_data_cap3)

        #Calibration Coefficients
        coefficients = {
            "cap1": {
                0: {"1": (0, 0), "2": (0, 0), "3": (0, 0), "5": (0, 0)},
                2412: {"1": (-0.024, -61.023), "2": (0.2827, -36.555), "5": (0.3878, -9.6743)},
                2417: {"1": (0.04, -56.859), "2": (-0.0036, -53.377), "5": (0.021, -36.732)},
                2422: {"1": (-0.0778, -70.582), "2": (-0.0465, -57.052), "3": (0.6694, -33.635), "5": (-0.1676, -82.611)},
                2427: {"3": (-0.8452, 9.2976), "1": (0.6731, -25.269), "2": (0.2202, -47.943)},
                2432: {"1": (0.6207, -23.241), "2": (0.3929, -36.143), "3": (0.2655, -37.651)},
                2437: {"3": (0.0444, -50.416)},
                2442: {"1": (-0.1255, -71.997), "2": (0.1047, -53.554), "3": (-0.2599, -69.391)},
                2447: {"1": (-0.6667, -124.17), "2": (-0.119, -83.807), "3": (-0.1774, -67.56), "5": (-0.1727, -79.866)},
                2452: {"1": (0.2067, -41.894), "2": (0.149, -44.829), "3": (-0.3546, -91.039), "5": (-0.7143, -114.57)},
                2457: {"1": (0.08, -53.266), "2": (0.425, -28.729), "3": (0.1188, -44.925), "5": (-0.0498, -72.19)},
                2462: {"1": (-0.1609, -64.342), "2": (0.2363, -38.416), "3": (-0.2958, -106.97)}
            },
```

```
    "cap2": {
        0: {"1": (0, 0), "2": (0, 0), "3": (0, 0), "5": (0, 0)},
        2412: {"1": (0.2101, -57.413), "2": (-0.0482, -80.835), "5": (0.225, -62.581)},
        2417: {"1": (0.2417, -51.294), "2": (0.1959, -59.09), "5": (0.0288, -68.802)},
        2422: {"1": (0.3673, -44.935), "2": (0.3049, -51.899), "3": (0.6694, -33.635), "5": (0.0153, -73.517)},
        2432: {"1": (0.6207, -23.241), "2": (0.3929, -36.143), "3": (-0.0883, -65.034)},
        2437: {"3": (0.7726, -21.769)},
        2442: {"1": (-0.1255, -71.997), "2": (0.1047, -53.554), "3": (0.2563, -54.682)},
        2447: {"1": (-0.0812, -53.422), "2": (-0.119, -83.807), "3": (0.3658, -49.46), "5": (-0.1727, -79.866)},
        2452: {"1": (0.132, -60.967), "2": (0.9795, 5.2032), "3": (-0.7243, -137.61), "5": (-0.7143, -114.57)},
        2457: {"1": (0.2378, -50.732), "2": (0.0334, -63.397), "3": (0.0499, -64.936), "5": (-0.0498, -72.19)},
        2462: {"1": (0.1852, -56.06), "2": (0.1738, -50.895), "3": (0.2272, -53.008)}
    },
    "cap3": {
        0: {"1": (0, 0), "2": (0, 0), "3": (0, 0), "5": (0, 0)},
        2412: {"1": (0.7291, -23.486), "2": (-0.0186, -87.503), "5": (0.3878, -9.6743)},
        2417: {"1": (0.39, -47.237), "2": (-0.2875, -108.68), "5": (0.021, -36.732)},
        2422: {"1": (0.3685, -54.482), "2": (0.324, -60.304), "3": (0.6694, -33.635), "5": (0.0396, -37.967)},
        2427: {"5": (-1.0054, -97.567)},
        2432: {"1": (0.6207, -23.241), "2": (0.3929, -36.143), "3": (0.7924, -6.1627)},
        2437: {"3": (0.7078, -16.978)},
        2442: {"1": (-0.1255, -71.997), "2": (0.1047, -53.554)},
        2447: {"1": (-0.0812, -53.422), "2": (0.0601, -43.927), "3": (-0.0129, -52.832), "5": (-0.1727, -79.866)},
        2452: {"1": (0.0747, -70.878), "2": (0.3107, -56.937), "3": (-0.7243, -137.61), "5": (-0.7143, -114.57)},
        2457: {"1": (0.1002, -31.957), "2": (-0.0519, -82.597), "3": (0.0073, -38.155), "5": (-0.0498, -72.19)},
        2462: {"1": (0.1534, -65.258), "2": (0.2363, -38.416), "3": (0.2018, -59.922)}
    }
}

# Define the mapping of Device ID to SSID
device_ssid_mapping = {
    1: "Realme_6Pro",
    2: "RedmiNote_11",
    3: "TechnoCamon_20",
    4: "Vivo_Y21T",
    5: "Xiaomi_10T",
    6: "Xiaomipoco_X3"
}

# Function to calculate new signal strength and add valid rows to new DataFrame
def update_signal_strength(row):
    source = row['source']
    channel = row['channel']
    signal_strength = row['signal_strength']
    ssid_name = row['ssid']  # Name of the SSID

    # Find the correct device_id based on the SSID's name
    device_id = None
    for id, ssid in device_ssid_mapping.items():
        if ssid == ssid_name:
            device_id = id
            break

    # Check if calibration data is available for the source, channel, and device_id
    if (
            coefficients.get(source)
            and coefficients[source].get(channel)
            and coefficients[source][channel].get(str(device_id))
    ):
        a, b = coefficients[source][channel][str(device_id)]
        new_signal_strength = a * signal_strength + b
        row['signal_strength'] = new_signal_strength
        row['ssid'] = device_id  # Set SSID to the correct Device_ID
        return row
    else:
        # Log missing coefficients but retain original row data
        if not coefficients.get(source):
            print(f"No coefficients found for source '{source}'")
        elif not coefficients[source].get(channel):
            print(f"No coefficients found for channel '{channel}' in source '{source}'")
        elif not coefficients[source][channel].get(str(device_id)):
            print(
                f"No coefficients found for device ID '{device_id}' in channel '{channel}', source '{source}'")
        # Retain the original signal strength and other data
```

```
        # Update the 'ssid' column to reflect device ID
    row['ssid'] = str(device_id)  # Change SSID to device ID

    return row

# Function to process a DataFrame and remove duplicates
def process_dataframe(ap_data, updated_df):
    for _, row in ap_data.iterrows():
        new_row = update_signal_strength(row)
        if new_row is not None:
            updated_df = updated_df._append(new_row, ignore_index=True)

    # Remove duplicates based on key columns
    updated_df = updated_df.drop_duplicates(subset=['source', 'channel', 'ssid', 'signal_strength'], keep='first')
    return updated_df

    # Remove duplicates based on key columns
    updated_df = updated_df.drop_duplicates(subset=['source', 'channel', 'ssid', 'signal_strength'], keep='first')
    return updated_df

# Create a dictionary to store the updated DataFrames
updated_data = {
    "cap1": pd.DataFrame(),
    "cap2": pd.DataFrame(),
    "cap3": pd.DataFrame()
}

# Process each DataFrame and update the corresponding entry in 'updated_data'
updated_data["cap1"] = process_dataframe(data_cap1, updated_data["cap1"])
updated_data["cap2"] = process_dataframe(data_cap2, updated_data["cap2"])
updated_data["cap3"] = process_dataframe(data_cap3, updated_data["cap3"])

# Combine data from cap1, cap2, and cap3
combined_data = pd.concat([updated_data["cap1"], updated_data["cap2"], updated_data["cap3"]], ignore_index=True)
validationData = pd.concat([updated_data["cap1"], updated_data["cap2"], updated_data["cap3"]], ignore_index=True)

# ARRANGED DATA*******************************************************************************

# Assuming 'combined_data' is a DataFrame containing your data

# Find unique SSIDs in the dataset
unique_ssids = combined_data['ssid'].unique()

# Initialize an empty list to store dictionaries
combinations_list = []

# Iterate over unique SSIDs
for ssid in unique_ssids:
    # Filter data for the specific SSID
    specific_ssid_data = combined_data[combined_data['ssid'] == ssid]

    # Get unique 'cap' categories for the specific SSID
    unique_caps = specific_ssid_data['source'].unique()

    # Initialize lists to store unique channels and signal strengths for each 'cap'
    unique_channels = {}
    signal_strengths = {}

    # Iterate over unique 'cap' categories
    for cap in unique_caps:
        # Filter data for each 'cap' category within the specific SSID
        cap_data = specific_ssid_data[specific_ssid_data['source'] == cap]

        # Extract unique channels and signal strengths for this 'cap'
        unique_channels[cap] = cap_data['channel'].unique()
        signal_strengths[cap] = cap_data.set_index('channel')['signal_strength'].to_dict()

    # Generate all combinations of unique channels
    all_combinations = product(*unique_channels.values())

    # Append combinations to the list
    for combination in all_combinations:
        combination_dict = {
            'mac_address': specific_ssid_data['mac_address'].iloc[0],
            'ssid': ssid,
```

```
            'timestamp': specific_ssid_data['timestamp'].iloc[0],
        }
        for i, cap in enumerate(unique_caps):
            combination_dict[f'channel_cap{i + 1}'] = combination[i]
            combination_dict[f'signal_strength_cap{i + 1}'] = signal_strengths[cap].get(combination[i], None)
        combinations_list.append(combination_dict)

    # Create the DataFrame outside the loop
    combinations_df = pd.DataFrame(combinations_list, columns=[
        'mac_address', 'ssid', 'timestamp',
        'channel_cap1', 'channel_cap2', 'channel_cap3',
        'signal_strength_cap1', 'signal_strength_cap2', 'signal_strength_cap3'
    ])

# CLEANED DATA ******************************************************************************

    # Apply Cleaning

    cleaned_data = clean_data(combinations_df)

# PREPROCESSED DATA*************************************************************************

    # Apply preprocessing

    X_processed = preprocess_data(cleaned_data)

# TRAINED DATA*******************************************************************************

    # Load datasets for each device
    datasets = {}
    for ssid in range(1, 7):
        file_path = f'C:/Users/Thesis2.0/django_thesis/KNN Algorithm/trainingdataset/device{ssid}_trainingdataset_300.csv'
        training_data = pd.read_csv(file_path)
        cleaned_data = clean_data_dataset(training_data)  # Apply your data cleaning function
        datasets[f'device{ssid}'] = cleaned_data

    # Identify unique SSIDs from 'X_processed'
    relevant_ssids = X_processed['ssid'].unique()  # Identify relevant SSIDs for training

    # Filter the datasets based on relevant SSIDs
    filtered_datasets = {f'device{ssid}': datasets[f'device{ssid}'] for ssid in relevant_ssids}

    # Apply preprocessing to the filtered datasets and create training data
    X_train_sets = {}
    y_train_sets = {}

    for device, data in filtered_datasets.items():
        X_train, y_train = preprocess_data_dataset(data)  # Apply your preprocessing function
        X_train_sets[device] = X_train
        y_train_sets[device] = y_train

    #y_train = y_train.sample(frac=1).reset_index(drop=True)

# PREPROCESSED DATA *************************************************************************

    # Create X_test_sets for each relevant SSID
    X_test_sets = {}

    # Extract the relevant data from 'X_processed' based on unique SSIDs
    for ssid in relevant_ssids:
        # Filter X_processed to include only the rows with the current SSID
        X_test_subset = X_processed[X_processed['ssid'] == ssid]

        # Add unique channels to align with the corresponding training set
        # You need to determine which X_train set to use for adding unique channels
        corresponding_device = f'device{ssid}'
        if corresponding_device in X_train_sets:
            X_train = X_train_sets[corresponding_device]
            # Add unique channels to X_test_subset using X_train's columns
            X_test = add_unique_channels(X_test_subset, X_train)  # This function should align channels

            # Store the resulting X_test in the dictionary
            X_test_sets[corresponding_device] = X_test
        else:
            print(f"Warning: No X_train found for device {ssid}")
```

```
# PREDICTED DATA*********************************************************************************

    # Dictionary to store KNN classifiers for each relevant SSID
    knn_classifiers = {}

    # Dictionary to store predictions for each relevant SSID
    predictions = {}

    # Define the value of k for KNN
    k = 1  # Adjustable

    # Iterate over relevant SSIDs and get corresponding X_train, y_train, and X_test
    for ssid in relevant_ssids:
        # Ensure we're dealing with the right SSID key
        ssid_key = f'device{ssid}'

        # Get the training data and labels for the current SSID
        X_train = X_train_sets.get(ssid_key, pd.DataFrame())  # Default to empty DataFrame if not found
        y_train = y_train_sets.get(ssid_key, pd.DataFrame())

        # Get the test data for the current SSID
        X_test = X_test_sets.get(ssid_key, pd.DataFrame())

        # Check for empty datasets
        if X_train.empty or X_test.empty:
            print(f"Warning: No data for SSID {ssid}. Skipping...")
            continue  # Skip if there's no data

        # Scale the data
        scaler = StandardScaler()
        scaler.fit(X_train)  # Fit only on the training data
        X_train_scaled = scaler.transform(X_train)  # Transform the training set
        X_test_scaled = scaler.transform(X_test)  # Transform the test set

        # Train the KNN classifier
        knn = KNeighborsClassifier(n_neighbors=k, p=2)  # Euclidean metric (p=2)
        #time.sleep(2)

        knn.fit(X_train_scaled, y_train)  # Fit the classifier
        time.sleep(10)

        # Store the trained KNN classifier in the dictionary
        knn_classifiers[ssid_key] = knn

        # Make predictions on the test data
        y_test_predictions = knn.predict(X_test_scaled)

        # Store the predictions in the dictionary
        predictions[ssid_key] = y_test_predictions

# See the predictions and translate them

    # Create a new DataFrame to store SSID, predicted floor ID, and predicted room ID
    predictions_summary = pd.DataFrame(columns=['ssid', 'predicted_floorid', 'predicted_roomid'])

    # Loop through each SSID and its corresponding predictions
    for ssid_key, y_test in predictions.items():
        # Get column names from 'y_train' to map the predictions
        column_names = y_train.columns

        # Separate 'floor_id' column names
        floor_column_names = [col for col in column_names if col.startswith('floorid_')]



        # Ensure the SSID is stored in the new DataFrame
        for i, y_pred in enumerate(y_test):
            print(f"\nPrediction {i + 1}:")

            # Find indices where the prediction is True for floor_id
            floor_true_indices = np.where(y_pred[:len(floor_column_names)] == 1)[0]

            # Get the predicted floor ID
            predicted_floorid = floor_column_names[floor_true_indices[0]] if floor_true_indices else None
```

```
        # Add information to the new DataFrame
        ssid_value = ssid_key.replace('device', '')  # Extract the SSID number

        # Determine the predicted room ID based on the predicted floor ID
        if predicted_floorid:
            floor_number = int(predicted_floorid.replace("floorid_", ""))
            if floor_number <= 100:
                predicted_roomid = 216
            elif floor_number > 189:
                predicted_roomid = 111
            else:
                predicted_roomid = 213
        else:
            predicted_roomid = None

        # Add to 'predictions_summary'
        predictions_summary = predictions_summary._append({
            'ssid': ssid_value,
            'predicted_floorid': predicted_floorid.replace("floorid_", "") if predicted_floorid else None,
            'predicted_roomid': predicted_roomid,
        }, ignore_index=True)

        # Add a timestamp for reference
        current_timestamp = datetime.now()
        print("Timestamp:", current_timestamp)

    # Extract columns for 'final_predictions'
    selected_columns = predictions_summary[['ssid', 'predicted_floorid', 'predicted_roomid']]

    # Create a new DataFrame named 'final_predictions'
    final_predictions = pd.DataFrame(selected_columns)

    print('Final Predictions: ')
    print(final_predictions)

    # Count occurrences of each room ID
    roomid_counts = final_predictions['predicted_roomid'].value_counts()

    # Count occurrences of specific rooms and floors
    be111_count = roomid_counts.get('111', 0)
    be213_count = roomid_counts.get('213', 0)
    be216_count = roomid_counts.get('216', 0)
    first_floor_count = be111_count
    second_floor_count = be213_count + be216_count
    total_persons = roomid_counts.sum()

    # Create a DataFrame for the counts
    counts_df = pd.DataFrame({
        'RoomID': ['BE111', 'BE213', 'BE216', 'First Floor', 'Second Floor', 'Total'],
        'Count': [be111_count, be213_count, be216_count , first_floor_count, second_floor_count, total_persons]
    })

# CSV FILE ******************************************************************************

    connection = connect_to_database()

    cursor = connection.cursor()

    # Query to fetch latitude and longitude from the "markers" table
    query = "SELECT title, lat, lng FROM markers_combined"
    # Execute the query and fetch the results

    cursor.execute(query)
    markers_data = cursor.fetchall()

    # Create a DataFrame from the results
    markers_df = pd.DataFrame(markers_data, columns=['title', 'lat', 'lng'])

    # Convert 'predicted_floorid' to object type in the aggregated_data DataFrame
    final_predictions['predicted_floorid'] = final_predictions['predicted_floorid'].astype(str)

    # Merge the aggregated_data DataFrame with markers_df based on 'predicted_floorid'
    result_df = pd.merge(final_predictions, markers_df, left_on='predicted_floorid', right_on='title',
                         how='left')
```

```
    # Drop the duplicate 'title' column
    result_df = result_df.drop(columns=['title'])

    # Create empty DataFrames to accumulate rows based on the condition
    df_map1 = pd.DataFrame()
    df_map2 = pd.DataFrame()

    for i, row in result_df.iterrows():
        # Convert 'predicted_roomid' column to integers
        row['predicted_roomid'] = int(row['predicted_roomid'])

        # Check if the value in the 'predicted_roomid' column is equal to 111
        if row['predicted_roomid'] == 111:
            # Append the current row to df_map2
            df_map2 = pd.concat([df_map2, row.to_frame().T], ignore_index=True)
        else:
            # Append the current row to df_map1
            df_map1 = pd.concat([df_map1, row.to_frame().T], ignore_index=True)

    # Specify the full paths for the CSV files
    csv_file_path_map1 = 'C:/Users/Thesis2.0/django_thesis/rfid_ips/css/final_predicted_values_aggregated_map1.csv'
    csv_file_path_map2 = 'C:/Users/Thesis2.0/django_thesis/rfid_ips/css/final_predicted_values_aggregated_map2.csv'

    # Save the accumulated DataFrames to the specified CSV files
    df_map1.to_csv(csv_file_path_map1, index=False)
    df_map2.to_csv(csv_file_path_map2, index=False)

    time.sleep(1)

# Run the main function
if __name__ == "__main__":
    main()
```

TABLE XVI. BINARY RELEVANCE CONFUSION MATRIX FOR 100 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	7761	111
Negative	153	8635

BE 213

	Positive	Negative
Positive	8765	3740
Negative	19	4136

BE 214

	Positive	Negative
Positive	11455	1488
Negative	0	3717

TABLE XVII. CLASSIFIER CHAIN CONFUSION MATRIX FOR 100 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	7761	111
Negative	153	8635

BE 213

	Positive	Negative
Positive	9140	3365
Negative	111	4044

BE 214

	Positive	Negative
Positive	12943	0
Negative	3237	480

TABLE XVIII. LABEL POWERSET CONFUSION MATRIX FOR 100 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	7852	20
Negative	375	8413

BE 213

	Positive	Negative
Positive	12161	344
Negative	1473	2682

BE 214

	Positive	Negative
Positive	11459	1484
Negative	0	3717

TABLE XIX. KNN ALGORITHM CONFUSION MATRIX FOR 100 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	1430	3
Negative	0	2196

BE 213

	Positive	Negative
Positive	1061	3
Negative	3	1135

BE 214

	Positive	Negative
Positive	1907	0
Negative	3	292

TABLE XX. MLKNN ALGORITHM CONFUSION MATRIX FOR 100 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	7761	111
Negative	153	8635

BE 213

	Positive	Negative
Positive	8765	3740
Negative	19	4136

BE 214

	Positive	Negative
Positive	11455	1488
Negative	0	3717

TABLE XXI. BINARY RELEVANCE CONFUSION MATRIX FOR 200 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	8906	64
Negative	250	11420

BE 213

	Positive	Negative
Positive	11604	4188
Negative	11	4837

BE 214

	Positive	Negative
Positive	14764	1754
Negative	0	4122

TABLE XXII. CLASSIFIER CHAIN CONFUSION MATRIX FOR 200 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	8906	64
Negative	250	11420

BE 213

	Positive	Negative
Positive	12111	3681
Negative	74	4774

BE 214

	Positive	Negative
Positive	16518	0
Negative	3464	658

TABLE XXIII. LABEL POWERSET CONFUSION MATRIX FOR 200 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	8961	9
Negative	74	10946

BE 213

	Positive	Negative
Positive	15119	673
Negative	1712	3136

BE 214

	Positive	Negative
Positive	14764	1754
Negative	0	4122

TABLE XXIV. KNN ALGORITHM CONFUSION MATRIX FOR 200 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	1994	2
Negative	2	1458

BE 213

	Positive	Negative
Positive	1853	2
Negative	2	1599

BE 214

	Positive	Negative
Positive	3061	0
Negative	0	395

TABLE XXV. MLKNN ALGORITHM CONFUSION MATRIX FOR 200 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	8970	0
Negative	0	11670

BE 213

	Positive	Negative
Positive	15792	0
Negative	0	4848

BE 214

	Positive	Negative
Positive	16518	0
Negative	0	4122

TABLE XXVI. BINARY RELEVANCE CONFUSION MATRIX FOR 300 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	23435	0
Negative	0	26265

BE 213

	Positive	Negative
Positive	37220	2
Negative	2	12476

BE 214

	Positive	Negative
Positive	38741	2
Negative	2	10955

TABLE XXVII. CLASSIFIER CHAIN CONFUSION MATRIX FOR 300 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	23435	0
Negative	0	26265

BE 213

	Positive	Negative
Positive	37220	2
Negative	2	12476

BE 214

	Positive	Negative
Positive	38741	2
Negative	2	10955

TABLE XXVIII. LABEL POWERSET CONFUSION MATRIX FOR 300 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	23435	0
Negative	0	26265

BE 213

	Positive	Negative
Positive	37220	2
Negative	2	12476

BE 214

	Positive	Negative
Positive	38741	2
Negative	2	10955

TABLE XXIX. KNN ALGORITHM CONFUSION MATRIX FOR 300 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	3994	0
Negative	0	2107

BE 213

	Positive	Negative
Positive	2845	1
Negative	3	3252

BE 214

	Positive	Negative
Positive	5359	3
Negative	1	738

TABLE XXX. MLKNN ALGORITHM CONFUSION MATRIX FOR 300 REFERENCE POINTS

BE 111

	Positive	Negative
Positive	23435	0
Negative	0	26265

BE 213

	Positive	Negative
Positive	37220	2
Negative	2	12476

BE 214

	Positive	Negative
Positive	38741	2
Negative	2	10955

TABLE XXXI. SUMMARY OF THE ACCURACY AND RUN TIME OF DIFFERENT POSITION ALGORITHMS

Number of Reference Points	**Position Algorithm**									
	Binary Relevance		**Classifier Chain**		**Label Powerset**		**KNN**		**MLKNN**	
	Accuracy (%)	***Run Time (sec)***	***Accuracy (%)***	***Run Time (sec)***	***Accuracy (%)***	***Run Time (sec)***	***Accuracy (%)***	***Run Time (sec)***	***Accuracy (%)***	***Run Time (sec)***
100	67.97	0.01	78.99	0.01	88.91	0.07	99.73	0.03	99.67	0.08
200	70.94	0.01	81.65	0.01	88.20	0.07	99.88	0.06	99.88	0.14
300	67.25	0.02	79.36	0.05	87.01	0.18	99.93	0.2	99.92	0.23

APPENDIX I

User Manual

Table of Contents

1. Introduction
2. System Overview
3. Installation
4. XAMPP Control Panel
5. User Interface
    5.1. Login Administrator
    5.2. Devices
        5.2.1. Device Manager
        5.2.2. Enrollment
        5.2.3. Attendance
    5.3. Register Users
    5.4. User
    5.5. User Log
        5.5.1. Log Filter
    5.6. Map
6. Troubleshooting
    6.1. Common Issues
        6.1.1. Login
        6.1.2. Attendance
        6.1.3. Register User
        6.1.4. Add User

USER MANUAL

1. Introduction

The SMART-TRACE system integrates the RFID database and WiFi-RSSI map tracking inside a building to upgrade disaster management. SMART-TRACE system's initial purpose is to locate potential casualties in aiding effective and quick emergency response. Also, this system is a website interface that is locally available to a network server. This manual will guide the user on how to activate the system, the registration process of the RFID tags, and providing information on all the website functionalities.

2. System Overview

The SMART-TRACE consists of hardware and software components. The hardware includes UHF RFID Tags for the beneficiaries, while the software manages the registration processes and provides visualization such as registered users in the system, user logs, and an interactive map for the location of its beneficiaries. The SMART-TRACE system provides user information, user logs, and the location of the beneficiaries with utmost security, privacy, and accuracy.

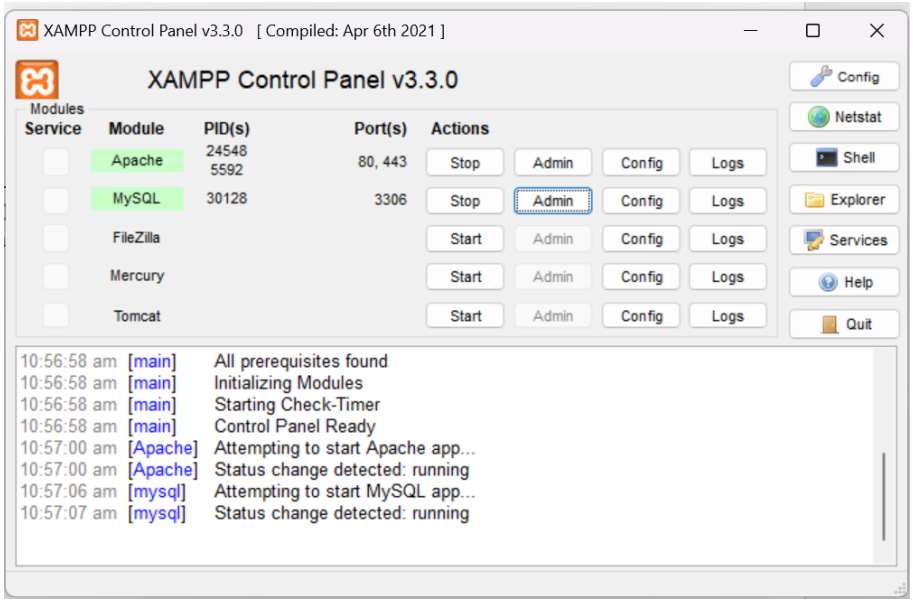
3. Installation

For the system to work, these are the following significant software:

- XAMPP Control Panel – Database management
- Python Compiler (VS Code or PyCharm) – Running machine learning tools
- Arduino Integrated Development Environment (Arduino IDE) – Flashing programs to microcontroller

4. XAMPP Control Panel

XAMPP is a web server developed by Apache Friends, which has a MariaDB database and programming languages Perl and PHP. It is an open-source application that is easy to install and use.

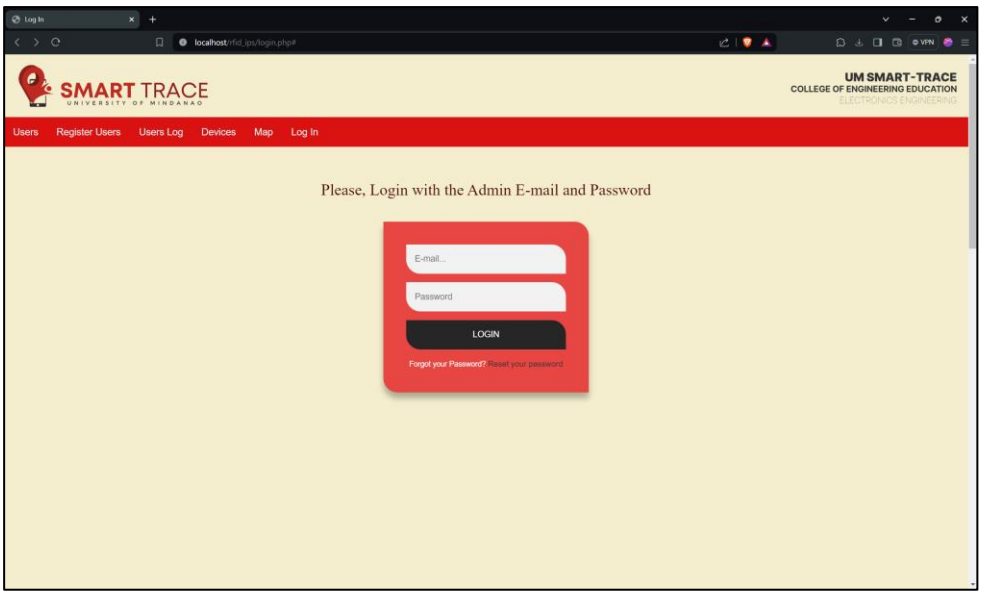


a. APACHE serves as a web server, presenting the system interface on a website accessible via the internet.
b. MySQL functions as a data management system, facilitating the storage and retrieval of data through database creation. Typically, PHP is employed as the server-side scripting language for developing dynamic web applications.

5. User Interface

The user interface or the web interface can be accessed once the XAMPP Database is activated. This can be accessible through the local host network of the computer. The local host network is the default IP address of the computer, therefore, only that is connected to the network of the computer that will be able to access the website. Also, the specific site for the website is shown in the figure below.

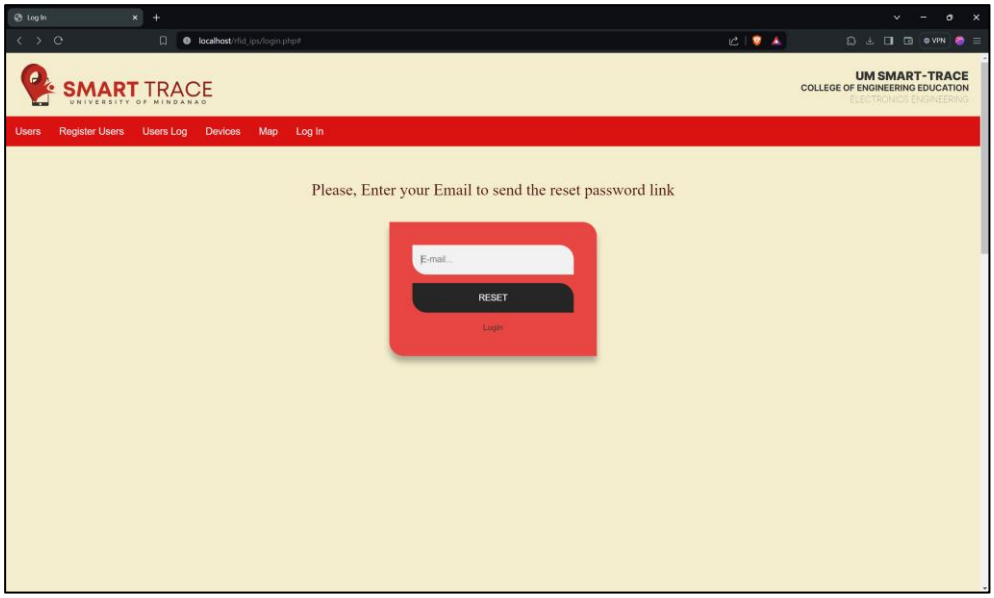
5.1. Login Administrator



a. Login

- The default email address is “admin@gmail.com” and the default password is “123”.
- Then, the default email address and password can then be changed afterwards.

b. Reset Password



Note: Only the administrator can access to change or reset the password of the system.

5.2. Devices

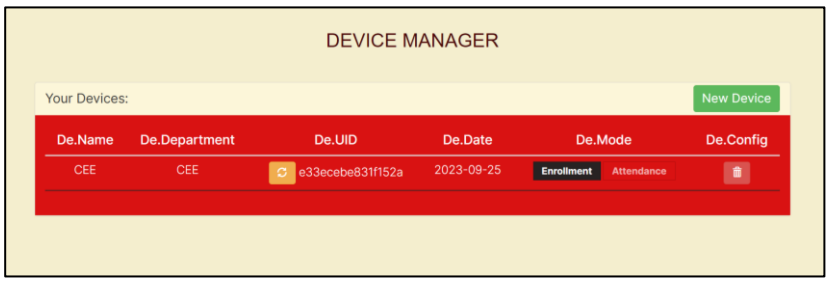
The Devices section manages UHF RFID tags for users, such as students or staff. It enables administrators to organize and secure device information, ensuring each RFID tag is properly registered, categorized, and maintained for effective system operation.

5.2.1. Device Manager

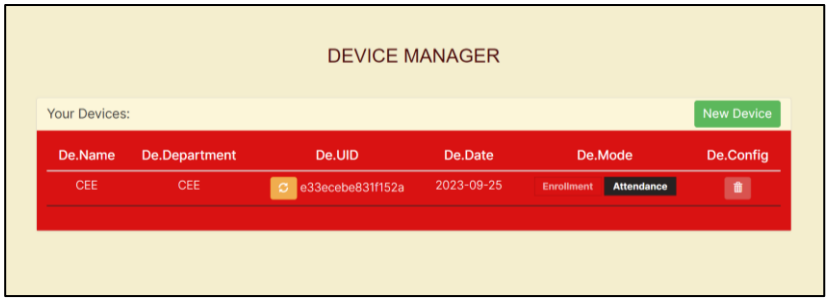
a. **Device Name (De. Name)**: This is the general name for the user's UHF RFID Tag, such as a student, professor, or service crew working or present inside the building.

b. **Device Department (De. Department)**: The administrator can use this field to allow the administrator to separate building names.
c. **Device UID (De. UID)**: This random ID generator adds security to the system. It is a unique ID for registering UHF RFID tags. Without this ID, the system cannot receive other data.
d. **Device Date (De. Date)**: The date the administrator recorded the device.
e. **Device mode**: To distinguish between "registration" and "attendance", the system can select:

a. **Enrollment**: This specifies that the device is in "enrollment mode", allowing the administrator to add more data to the database, such as students who register in the system.



b. **Attendance**: This records the time in and out of beneficiaries who enter and leave the specified area.



f. **Device Configuration (De. Config)**: This allows the administrator to delete devices in the system.

5.2.2. Enrollment

The enrollment function is for the system to register another UHF RFID Tag into the system. The administrator cannot enroll another tag without this operation. A detailed preview of the registration is found in section 5.3.

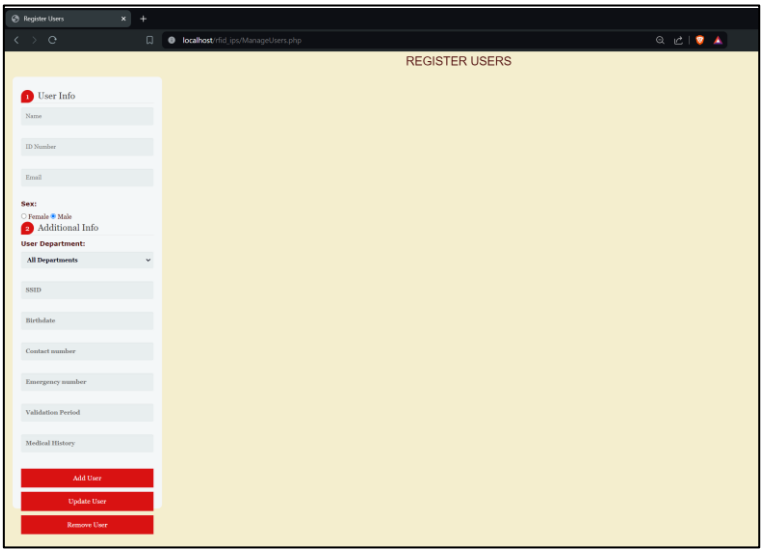
5.2.3. Attendance

The attendance function is for the system to record the registered user, which can be found in section 5.5, the User Log of the system.

If this is not turned on, the system will not record the user even if the UHF RFID reader has read it.

5.3. Register Users

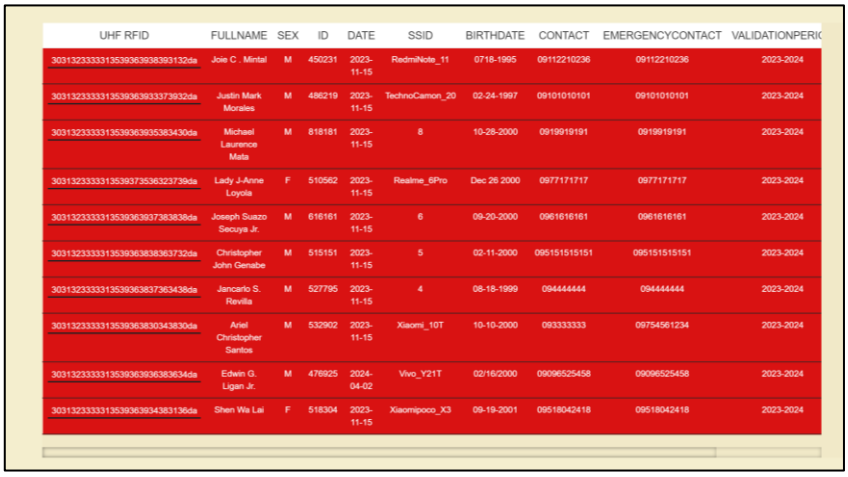
This is the registration interface of the SMART-TRACE system. This is where the administrator can add, update, or remove users in the system.



Note: The administrator can register user once the "enrolment" function is turned on.

Steps for the registration:

1. Use a UHF RFID Tag for enrollment/registration. An MFRC522 RFID tag will not be scanned by the UHF RFID reader because it has a different frequency band.
2. Scan the UHF RFID Tag using the UHF RFID reader/scanner for the unregistered ID to appear in the system.
3. Click the unregistered ID (the underlined UHF RFID) and fill up the necessary information.
4. Finally, click the "Add User" for the user to be registered in the XAMPP Database. If the user was successfully registered, it should show the necessary information as shown in the figure below.



UHF RFID	FULLNAME	SEX	ID	DATE	SSID	BIRTHDATE	CONTACT	EMERGENCYCONTACT	VALIDATIONPERIO
303132333331353936393839313 2da	Joie C . Mintal	M	450231	2023-11-15	RedmiNote_11	0718-1995	09112210236	09112210236	2023-2024
303132333331353936393337393 2da	Justin Mark Morales	M	486219	2023-11-15	TechnoCamon_20	02-24-1997	09101010101	09101010101	2023-2024
303132333331353936393535383 430da	Michael Laurence Mata	M	818181	2023-11-15	8	10-28-2000	0919919191	0919919191	2023-2024
303132333331353937353632373 9da	Lady J-Anne Loyola	F	510562	2023-11-15	Realme_6Pro	Dec 26 2000	0977171717	0977171717	2023-2024
303132333331353936393738383 8da	Joseph Suazo Secuya Jr.	M	616161	2023-11-15	6	09-20-2000	0961616161	0961616161	2023-2024
303132333331353936383836373 2da	Christopher John Genabe	M	515151	2023-11-15	5	02-11-2000	095151515151	095151515151	2023-2024
303132333331353936383736343 8da	Jancarlo S. Revilla	M	527795	2023-11-15	4	08-18-1999	094444444	094444444	2023-2024
303132333331353936383034383 0da	Ariel Christopher Santos	M	532902	2023-11-15	Xiaomi_10T	10-10-2000	093333333	09754561234	2023-2024
303132333331353936393638363 4da	Edwin G. Ligan Jr.	M	476925	2024-04-02	Vivo_Y21T	02/16/2000	09096525458	09096525458	2023-2024
303132333331353936393438313 6da	Shen Wa Lai	F	518304	2023-11-15	Xiaomipoco_X3	09-19-2001	09518042418	09518042418	2023-2024

a. **Add User**: Add an unregistered ID into the system
b. **Update User**: Update a registered user's information into the system.
c. **Remover User**: Removed a registered user from the system.

Primary Information

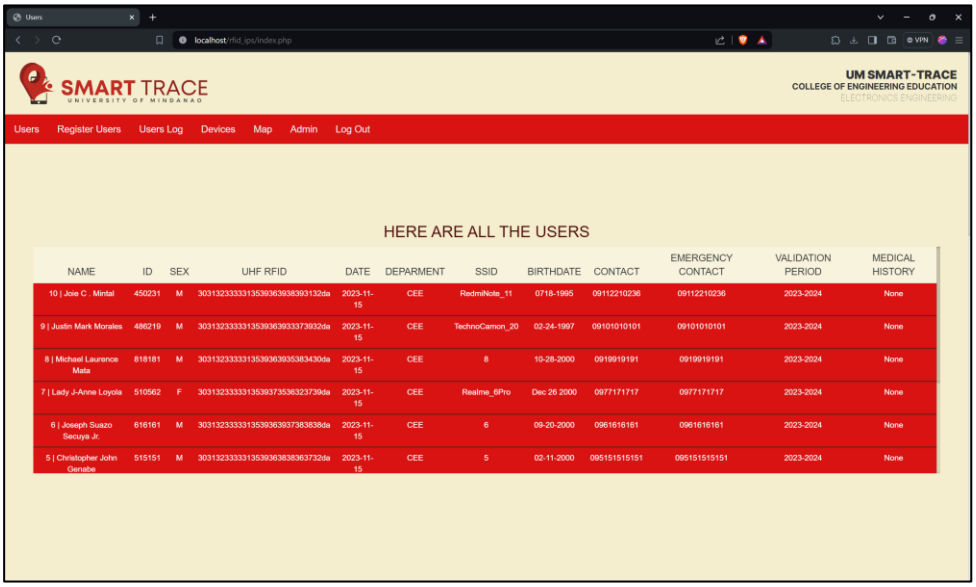
1. Name: First Name Middle Name Surname Suffix Name
2. ID Number: ID number of the company
3. E-mail: Company E-mail
4. Sex: Male or Female

Additional Information

1. Department: User Device Department (De. Department). A detailed preview of the Device Department is further explained in section 5.2.1.
2. Service Set Identifier (SSID): The SSID or the name of the mobile hotspot of the device.
    a. To know the SSID of the phone go to Settings > Hotspot > then the Name of the Hotspot.
3. Birthdate
4. Contact number
5. Emergency Number of the student: In case of emergency contact number
6. Validation Period: Validity period of the ID
7. Medical History of the student: Important information for the respondents

5.4. User

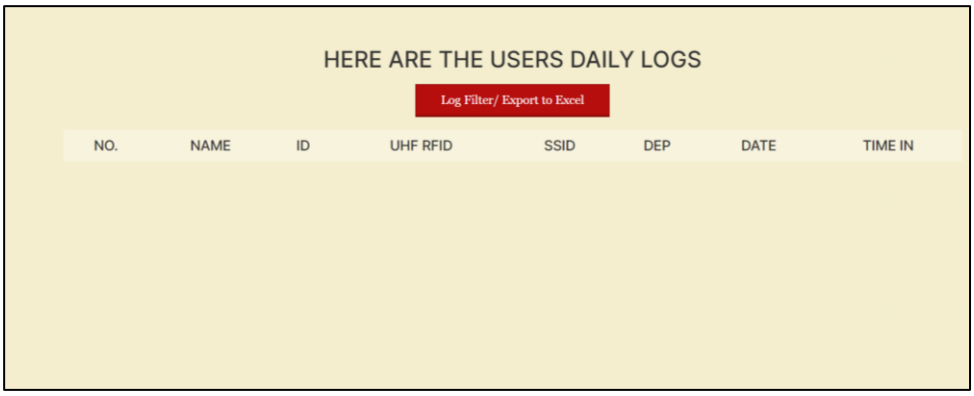
The User provides the list of the names of the registered user in the system. The necessary information that was recorded during the registration process is also shown in the User toggle. This is the part where we can now see all the Users being Registered.



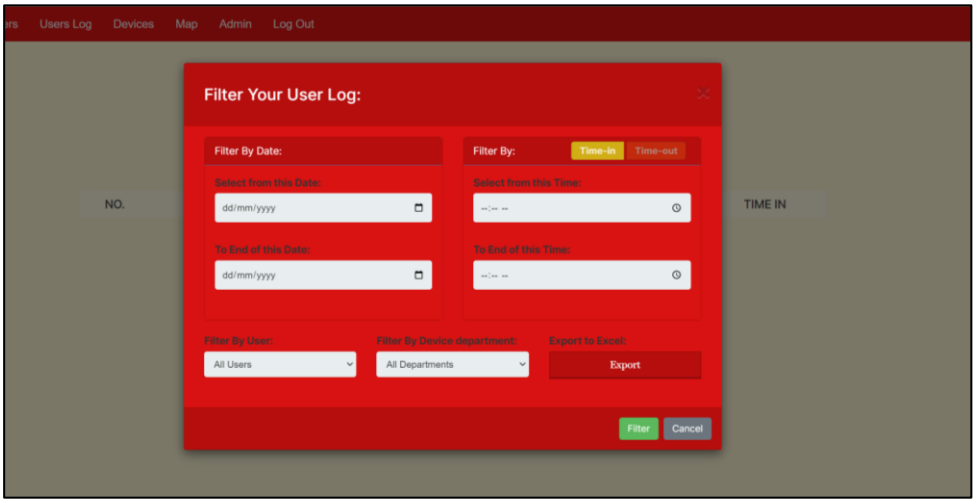
NAME	ID	SEX	UHF RFID	DATE	DEPARMENT	SSID	BIRTHDATE	CONTACT	EMERGENCY CONTACT	VALIDATION PERIOD	MEDICAL HISTORY
10 | Jose C . Mintal	450231	M	303132333331353936393839313 2da	2023-11-15	CEE	RedmiNote_11	0718-1995	09112210236	09112210236	2023-2024	None
9 | Justin Mark Morales	486219	M	303132333331353936393337393 2da	2023-11-15	CEE	TechnoCamon_20	02-24-1997	09101010101	09101010101	2023-2024	None
8 | Michael Laurence Mata	618181	M	303132333331353936393538343 0da	2023-11-15	CEE	8	10-28-2000	0919919191	0919919191	2023-2024	None
7 | Lady J-Anne Loyola	510562	F	303132333331353937353632373 9da	2023-11-15	CEE	Realme_6Pro	Dec 26 2000	0977171717	0977171717	2023-2024	None
6 | Joseph Suazo Secuya Jr.	616161	M	303132333331353936393738383 8da	2023-11-15	CEE	6	09-20-2000	0961616161	0961616161	2023-2024	None
5 | Christopher John Genabe	515151	M	303132333331353936383836373 2da	2023-11-15	CEE	5	02-11-2000	095151515151	095151515151	2023-2024	None

5.5. User Log

The user log provides the user activities and captures the details of the user such as its name, ID, ssid, department, and the time in and time out. In this way, the administrator can track the behavior of the user to acquire valuable information prior to an emergency.



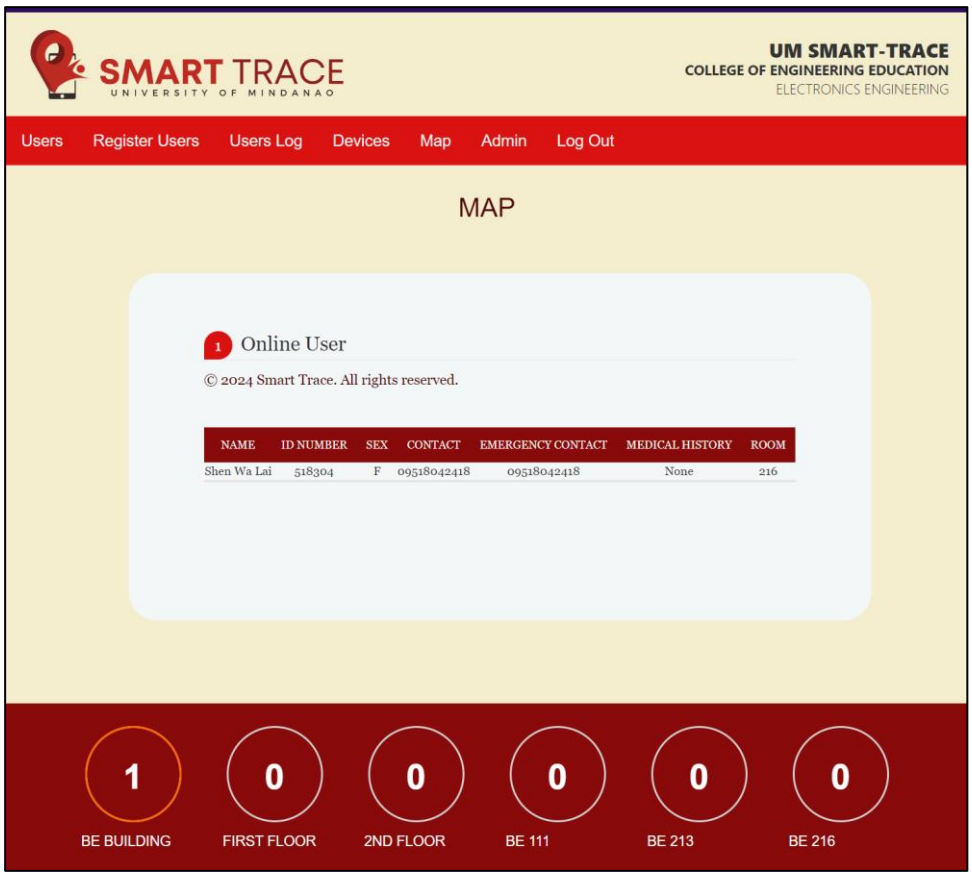
The User's Daily Logs or User Log interface displays the number of students present on the building, the phone they use, the department they belong to, and the date and time they entered the building.

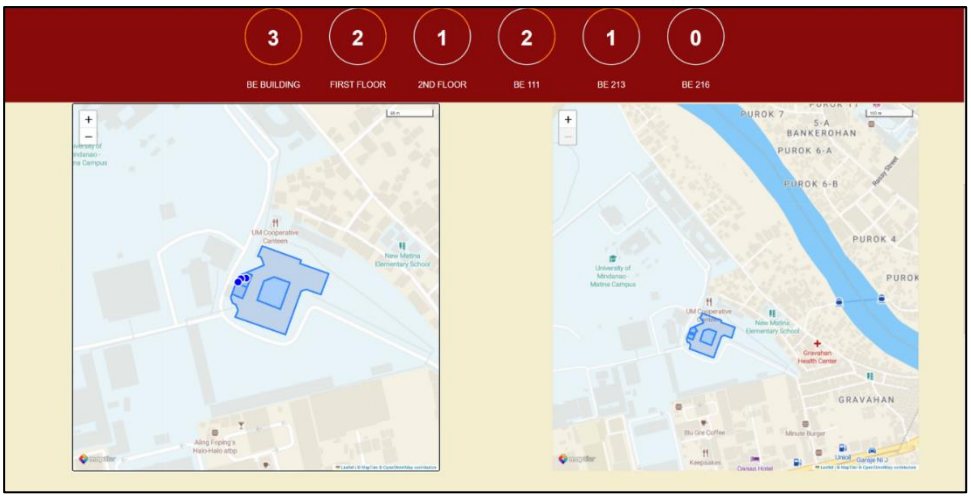


Log Filter interface allows the system to export data on a daily, weekly, or monthly basis to maintain database storage efficiency.

5.6. Map

This section of the SMART-TRACE system provides a 2D visual representation of the building, integrating RFID and WiFi-RSSI data to track the real-time location of users. This feature is essential for monitoring movement within the building, allowing for quick identification of potential casualties during emergencies. The map aids in disaster management by offering precise tracking, enhancing the effectiveness of emergency responses.





The following figure above illustrates a sample map with a tracked device, demonstrating how the SMART-TRACE system visually represents user locations in real-time within the building.

6. Troubleshooting

The troubleshooting section provides solutions to common issues that may arise while using the SMART-TRACE system. This section ensures that users can quickly identify and fix issues, maintaining the system's efficiency and reliability during critical operations.

6.1. Common Issues

6.1.1. Login. When logging in into the website, it is important to use "@gmail.com" accounts. These are also necessary when changing password in the future.

6.1.2. Attendance. There are instances where the website cannot track devices, this might be because the Device Mode, as shown in Section 5.2.1., is set to Enrollment mode. It is worth noting the difference between the Attendance and Enrollment Mode, as discussed in Section 5.2.2 and Section 5.2.3., respectively.

6.1.3. Register User. In contrast to the issue described in Section 6.1.2, if the system cannot detect an unregistered ID, it may be because the Device Mode, as shown in Section 5.2.1, is set to Attendance mode.

6.1.4. Add User. To add users or unregistered IDs, it is important to first click on the underlined UHF RFID, as shown in Section 5.3, before filling in the necessary information. The system will not overwrite the data in the UHF RFID if it is not clicked first.

APPENDIX J

Informed Consent

Title of Research: SMART-TRACE: A Combined RFID Databasing and WiFi-RSSI Fingerprinting Indoor Positioning for Disaster Management using Smartphone

Investigators, Affiliations, and Contact Information:

Reyden Jenn Cagata	**Shen Wa Lai**	**Edwin Ligan Jr.**
University of Mindanao	University of Mindanao	University of Mindanao
College of Engineering Education	College of Engineering Education	College of Engineering Education
Electronics Engineering Program	Electronics Engineering Program	Electronics Engineering Program
Davao City, Philippines	Davao City, Philippines	Davao City, Philippines
r.cagata.503515@umindanao.edu.ph	s.lai.518304@umindanao.edu.ph	j.ligan.476925@umindanao.edu.ph

Introduction and Purpose of the Study

This study proposes a SMART-TRACE system that can track the user. The proponent used two devices for the user to achieve its objective, an RFID Tag and a smartphone, to provide effective and quick emergency response during an emergency. Integrating both technologies, the UHF RFID databasing and WiFi-RSSI Indoor Positioning System (IPS), will help disaster management.

Description of the Research

By agreeing to participate in our study, you will be asked to register your basic personal information, such as your name. Furthermore, you will then be asked to register your mobile phone's SSID and MAC Address to our database. After the registration process, you and your smartphone will be asked to participate in our accuracy testing, wherein we will check through several trials if our system recognizes the registered user. Finally, you will be asked to attend a mock election to test the functionality of our device.

Subject Participation

There will be six (6) participants needed for this study. They must be a bona fide student at the University of Mindanao for the second semester of the academic year 2023-2024, enrolled in the BS Electronics Engineering program.

Potential Risks and Discomforts

No known risks will be apparent throughout the participation of our study.

Confidentiality

All information collected from the study will be coded to protect each subject's name. Although your information will be stored in our database as needed for the authentication processes, no names or other identifying information will be used when discussing or reporting data. The researchers will safely keep all data gathered from the said database, and only they can access it. Once the data has been fully analyzed, it will be destroyed.

Authorization

By signing this form, you authorize the use and disclosure of the following information for this research:

I authorize the use of the data collected from me during this study for the sole purpose of the said research.

Voluntary Participation and Authorization
Your decision to participate in this study is entirely voluntary. If you decide not to participate in this study, it will not affect you in any way.

Withdrawal from the Study and/or Withdrawal of Authorization
If you choose to participate in this study, you may withdraw from your participation at any time without penalty by informing us, the researchers. However, any data collected prior to the withdrawal will still be included in the study.

Cost
There is no cost for participating in this study.

I voluntarily agree to participate in this research study.

- ☐ **Yes**
- ☐ **No**

I understand that I will be given a copy of this signed Consent Form.

Name of Participant

Signature: Date Signed:

Name of Witness

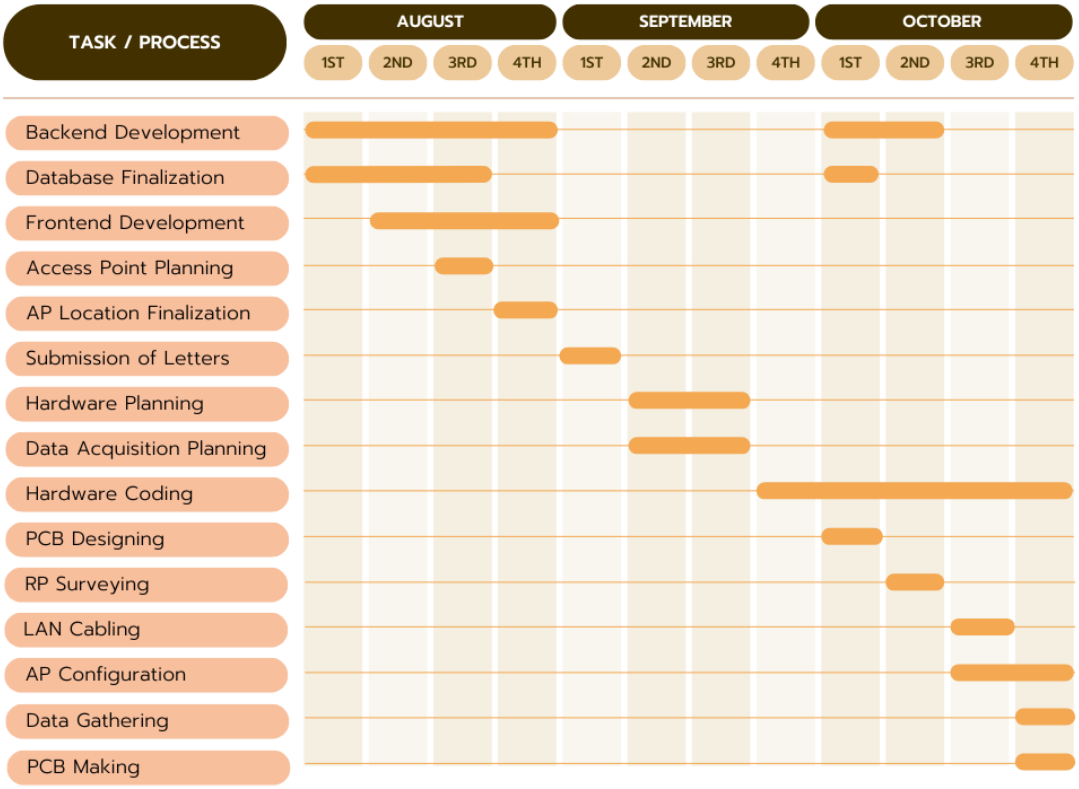
Signature: Date Signed:

Person Obtaining Consent

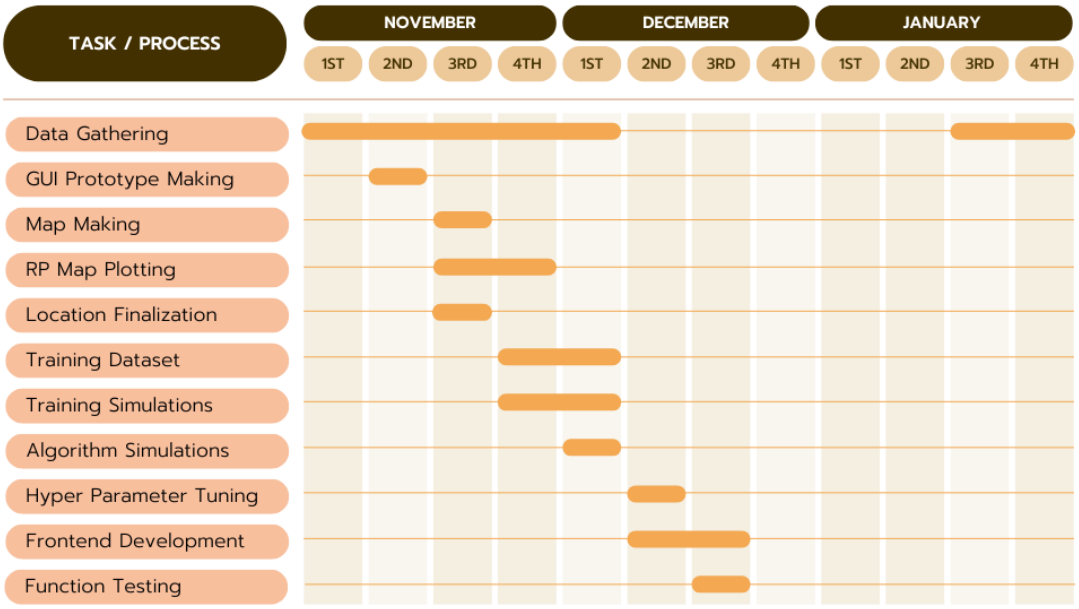
Signature: Date Signed:

APPENDIX K

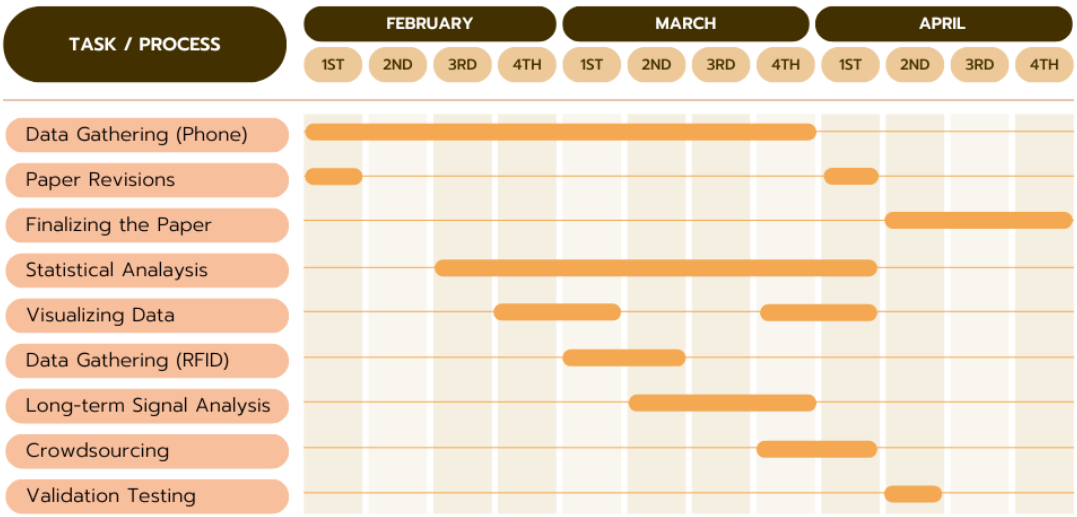
Gantt Chart



SMART-TRACE: A Combined RFID Databasing and WiFi-RSSI Fingerprinting Indoor Positioning for Disaster Management using Smartphone



SMART-TRACE: A Combined RFID Databasing and WiFi-RSSI Fingerprinting Indoor Positioning for Disaster Management using Smartphone



SMART-TRACE: A Combined RFID Databasing and WiFi-RSSI Fingerprinting Indoor Positioning for Disaster Management using Smartphone

APPENDIX L

Production Cost Summary

Item	Quantity	Cost	Total
UHF RFID Reader ISO18000-6C/6B RS232/RS485/Wiegand26 (902-928 MHZ) with RFID tags	2	₱7,000	₱14,000
RJ-45	10	₱5	₱50
MIKROTIK RouterBOARD hAP lite (RB941-2nD-TC)	3	₱1,450	₱4,350
UTP LAN Cable (CAT 6)	150 meters	₱1,500	₱1,500
Single-sided Universal PCB board	2 (3x4 inches)	28	₱56
ESP8266 Wireless Module Based	2	₱150	₱300
MAX2323 (TTL UART To RS232 Bus Converter Db9 Male Socket)	2	₱320	₱640
OVERALL COST			**₱20,896**

APPENDIX M

Research Documents

College of Engineering Education
2nd Floor, BE Building
Matina Campus, Davao City
Telefax: (082)296-1084
Phone No.: (082)300-5456/300-0647 Local 131

March 2, 2023

ENGR. RANDY E. ANGELIA
Faculty Member
This College

Dear **Engr. Angelia**:

Greetings!

I am pleased to appoint you as the adviser of **Reyden Jenn Cagata, Shen Wa Lai, and Edwin Ligan Jr.** of the undergraduate thesis entitled **Smart-Trace: A Combined RFID Databasing and WiFi-RSSI Fingerprinting Indoor Positioning for Disaster Management Using Smartphone** of the **Electronics Engineering** program of the College of Engineering Education. This appointment is in reference to your field of specialization and your qualification as a thesis adviser reflected in OPM 14.03 (Conduct of Undergraduate Thesis).

Please affix your signature hereunder to signify your acceptance of the appointment. Looking forward to your continued support in the research endeavor of the college.

Thank you, and have a nice day ahead!

Sincerely yours,

JETRON J. ADTOON
College Research Coordinator

Noted:

Dr. CHARLITO L. CAÑESARES
Dean of College

Accepted by:

RANDY E. ANGELIA
Faculty Member

College of Engineering Education
2nd Floor, BE Building
Matina Campus, Davao City
Telefax: (082)296-1084
Phone No.: (082)300-5456/300-0647 Local 131

March 2, 2023

ENGR. ROLIEVEN CAÑIZARES
Program Head, ECE Program
This College

Dear **Engr. Cañizares:**

Greetings!

I am pleased to appoint you as the statistician of **Reyden Jenn Cagata, Shen Wa Lai, and Edwin Ligan Jr.** of the undergraduate thesis entitled **Smart-Trace: A Combined RFID Databasing and WiFi-RSSI Fingerprinting Indoor Positioning for Disaster Management Using Smartphone** of the **Electronics Engineering** program of the College of Engineering Education. This appointment is in reference to your field of specialization and your qualification as a thesis statistician reflected in OPM 14.03 (Conduct of Undergraduate Thesis).

Please affix your signature hereunder to signify your acceptance of the appointment. Looking forward to your continued support in the research endeavor of the college.

Thank you, and have a nice day ahead!

Sincerely yours,

JETRON J. ADTOON
College Research Coordinator

Noted:

Dr. CHARLITO L. CAÑESARES
Dean of College

Accepted by:

ROLIEVEN CAÑIZARES
Program Head, ECE Program

UM
The University of Mindanao

RESEARCH AND PUBLICATION CENTER

[/] Main [ ] Branch _____

UNDERGRADUATE RESEARCH AGREEMENT

As a research student, I hereby declare my acceptance and adherence to the policies, rules, and guidelines in the conduct of my thesis/capstone/FS/creative work as follows:

1. The researcher/s shall be required to attend the research orientation and shall submit this signed agreement to the subject teacher. Failure to do so shall render the researcher/s ineligible for title defense.

2. The researcher/s shall group themselves by three, of their own choice and at their own risk. They shall work as a team from the initiation to completion of the project. They shall choose a research topic/theme/subject that is aligned with the agenda of the College and the University.

3. With advisement from the Research Coordinator(RC) and approval of the Dean of College, the researcher/s shall be provided with qualified research personnel (adviser and statistician) to help them out in the conduct of the project.

4. The researcher/s shall exercise due diligence and adhere to established standards/rules of data gathering, experimentation, field work that ensure safety, security and ethical practice. Research activities in remote and secluded areas are highly prohibited.

5. The researcher/s shall seek guidance, assistance and approval from their adviser in all research-related activities, documented in a journal/record book which is to be inspected by the subject teacher from time to time for purposes of monitoring.

6. The researcher/s shall always secure the endorsement of the adviser before submitting the outline/final manuscripts for defense to the RC.

7. The researcher/s shall be responsible to get updates about schedules for outline/final defense (one month before finals) and on thesis binding as announced/posted. Two (2) copies of the outlines/manuscripts shall be submitted to the RC one week before the scheduled defense.

8. The researcher/s shall pay P2700 per group as defense fee. Lone-research or two-member group are allowed only when there is no other possible way to divide the class. Anything beyond, needs to obtain RPC consent through writing. No additional payment shall be made to panel members. Researcher/s are strongly enjoined to report to the Dean/RPC any extra fee solicited by any party.

9. The researcher/s shall deliver oral defense as scheduled by the RC before the final exam. Failure to submit themselves for oral examination forfeits their right to obtain a passing grade. Likewise, failure to submit bounded duly approved final manuscript on time shall cause a grade of 7.2 which should be completed within the standard completion days as prescribed by the UM registrar. No extension shall be entertained.

10. In the event that the researcher/s failed to meet and/or satisfy the requirements and criteria for title, outline and final defenses, they shall be given another chance through redefense. This redefense is granted only once. Should the researcher/s failed during the redefense, they shall be given a failing grade and are required to reenroll the course.

11. The researcher/s shall bring all the raw data and evidences of data gathering procedures during the oral defense as indicator of data trustworthiness and reliability.

12. All transactions with research personnel during the processing/critiquing of the final manuscript shall require the use of the standard routing form. This routing form is to be submitted to the RC for documentation and future reference.

13. The researcher/s shall maintain the right for three (3) free plagiarism checks: one during Methods of Research or its equivalent, and two during Research Proper after final oral presentation. Five hundred pesos shall be charged for the fourth and succeeding checks. Threshold of similarity is 25% and lower.

14. Manuscripts shall follow the institutional format and referencing style; the study must be a product of exercise of intellect, free of dishonesty and fraud. Also, the papers shall be subjected for Grammarly check to ensure adherence to rules of language and writing conventions. Threshold of grammar accuracy is 95% and above.

15. The researcher/s shall acknowledge the right of the College to disallow or reject researches found to be deficient, unsatisfactory or plagiarized in form or substance.

16. The researcher/s shall take precautionary measures to ensure that their paper will not be copied, replicated or duplicated by others.

17. The researcher/s shall transfer the ownership of the study to the College once the paper has been successfully defended and bound. After which no paper shall be published or presented without consent/approval from the College. Sharing and/or furnishing copies of the research paper to pertinent offices or parties is subject to the approval of the College and shall require a signed Non-Disclosure Agreement from the receiving parties.

18. The researcher/s shall present their research projects strictly during their scheduled oral defense. Failure to present as scheduled, the researcher/s will have to wait for another cycle of defense.

I affix my signature to signify that I read, understand, and conform to the items enumerated above.

Conformed:

REYDEN JENN R. CAGATA
Signature over printed name of student

LETECIA UMBRIA
Signature over printed name of parent/guardian

ENGR. RANDY ANGELA
Signature over printed name of adviser

UM
The University of Mindanao

RESEARCH AND PUBLICATION CENTER

[/] Main [ ] Branch _____

UNDERGRADUATE RESEARCH AGREEMENT

As a research student, I hereby declare my acceptance and adherence to the policies, rules, and guidelines in the conduct of my thesis/capstone/FS/creative work as follows:

1. The researcher/s shall be required to attend the research orientation and shall submit this signed agreement to the subject teacher. Failure to do so shall render the researcher/s ineligible for title defense.

2. The researcher/s shall group themselves by three, of their own choice and at their own risk. They shall work as a team from the initiation to completion of the project. They shall choose a research topic/theme/subject that is aligned with the agenda of the College and the University.

3. With advisement from the Research Coordinator(RC) and approval of the Dean of College, the researcher/s shall be provided with qualified research personnel (adviser and statistician) to help them out in the conduct of the project.

4. The researcher/s shall exercise due diligence and adhere to established standards/rules of data gathering, experimentation, field work that ensure safety, security and ethical practice. Research activities in remote and secluded areas are highly prohibited.

5. The researcher/s shall seek guidance, assistance and approval from their adviser in all research- related activities, documented in a journal/record book which is to be inspected by the subject teacherfrom time to time for purposes of monitoring.

6. The researcher/s shall always secure the endorsement of the adviser before submitting the outline/final manuscripts for defense to the RC.

7. The researcher/s shall be responsible to get updates about schedules for outline/final defense (one month before finals) and on thesis binding as announced/posted. Two (2) copies of the outlines/manuscripts shall be submitted to the RC one week before the scheduled defense.

8. The researcher/s shall pay P2700 per group as defense fee. Lone-research or two-member group are allowed only when there is no other possible way to divide the class. Anything beyond, needs to obtain RPC consent through writing. No additional payment shall be made to panel members. Researcher/s are strongly enjoined to report to the Dean/RPC any extra fee solicited by any party.

9. The researcher/s shall deliver oral defense as scheduled by the RC before the final exam. Failure to submit themselves for oral examination forfeits their right to obtain a passing grade. Likewise, failure to submit bounded duly approved final manuscript on time shall cause a grade of 7.2 which should be completed within the standard completion days as prescribed by the UM registrar. No extension shall be entertained.

10. In the event that the researcher/s failed to meet and/or satisfy the requirements and criteria for title, outline and final defenses, they shall be given another chance through redefense. This redefense is granted only once. Should the researcher/s failed during the redefense, they shall be given a failing grade and are required to reenroll the course.

11. The researcher/s shall bring all the raw data and evidences of data gathering procedures during the oral defense as indicator of data trustworthiness and reliability.

12. All transactions with research personnel during the processing/critiquing of the final manuscript shall require the use of the standard routing form. This routing form is to be submitted to the RC for documentation and future reference.

13. The researcher/s shall maintain the right for three (3) free plagiarism checks: one during Methods of Research or its equivalent, and two during Research Proper after final oral presentation. Five hundred pesos shall be charged for the fourth and succeeding checks. Threshold of similarity is 25% and lower.

14. Manuscripts shall follow the institutional format and referencing style; the study must be a product of exercise of intellect, free of dishonesty and fraud. Also, the papers shall be subjected for Grammarly check to ensure adherence to rules of language and writing conventions. Threshold of grammar accuracy is 95% and above.

15. The researcher/s shall acknowledge the right of the College to disallow or reject researches found to be deficient, unsatisfactory or plagiarized in form or substance.

16. The researcher/s shall take precautionary measures to ensure that their paper will not be copied, replicated or duplicated by others.

17. The researcher/s shall transfer the ownership of the study to the College once the paper has been successfully defended and bound. After which no paper shall be published or presented without consent/approval from the College. Sharing and/or furnishing copies of the research paper to pertinent offices or parties is subject to the approval of the College and shall require a signed Non- Disclosure Agreement from the receiving parties.

18. The researcher/s shall present their research projects strictly during their scheduled oral defense. Failure to present as scheduled, the researcher/s will have to wait for another cycle of defense.

I affix my signature to signify that I read, understand, and conform to the items enumerated above.

Conformed:

SHEN NA S. LAI
Signature over printed name of student

MARY ROSE Q. MAMAC
Signature over printed name of parent/guardian

ENGR. RANDY ANGELA
Signature over printed name of adviser

UM The University of Mindanao	RESEARCH AND PUBLICATION CENTER [✓] Main [ ] Branch _____ UNDERGRADUATE RESEARCH AGREEMENT

As a research student, I hereby declare my acceptance and adherence to the policies, rules, and guidelines in the conduct of my thesis/capstone/FS/creative work as follows:

1. The researcher/s shall be required to attend the research orientation and shall submit this signed agreement to the subject teacher. Failure to do so shall render the researcher/s ineligible for title defense.

2. The researcher/s shall group themselves by three, of their own choice and at their own risk. They shall work as a team from the initiation to completion of the project. They shall choose a research topic/theme/subject that is aligned with the agenda of the College and the University.

3. With advisement from the Research Coordinator(RC) and approval of the Dean of College, the researcher/s shall be provided with qualified research personnel (adviser and statistician) to help them out in the conduct of the project.

4. The researcher/s shall exercise due diligence and adhere to established standards/rules of data gathering, experimentation, field work that ensure safety, security and ethical practice. Research activities in remote and secluded areas are highly prohibited.

5. The researcher/s shall seek guidance, assistance and approval from their adviser in all research-related activities, documented in a journal/record book which is to be inspected by the subject teacher from time to time for purposes of monitoring.

6. The researcher/s shall always secure the endorsement of the adviser before submitting the outline/final manuscripts for defense to the RC.

7. The researcher/s shall be responsible to get updates about schedules for outline/final defense (one month before finals) and on thesis binding as announced/posted. Two (2) copies of the outlines/manuscripts shall be submitted to the RC one week before the scheduled defense.

8. The researcher/s shall pay P2700 per group as defense fee. Lone-research or two-member group are allowed only when there is no other possible way to divide the class. Anything beyond, needs to obtain RPC consent through writing. No additional payment shall be made to panel members. Researcher/s are strongly enjoined to report to the Dean/RPC any extra fee solicited by any party.

9. The researcher/s shall deliver oral defense as scheduled by the RC before the final exam. Failure to submit themselves for oral examination forfeits their right to obtain a passing grade. Likewise, failure to submit bounded duly approved final manuscript on time shall cause a grade of 7.2 which should be completed within the standard completion days as prescribed by the UM registrar. No extension shall be entertained.

10. In the event that the researcher/s failed to meet and/or satisfy the requirements and criteria for title, outline and final defenses, they shall be given another chance through redefense. This redefense is granted only once. Should the researcher/s failed during the redefense, they shall be given a failing grade and are required to reenroll the course.

11. The researcher/s shall bring all the raw data and evidences of data gathering procedures during the oral defense as indicator of data trustworthiness and reliability.

12. All transactions with research personnel during the processing/critiquing of the final manuscript shall require the use of the standard routing form. This routing form is to be submitted to the RC for documentation and future reference.

13. The researcher/s shall maintain the right for three (3) free plagiarism checks: one during Methods of Research or its equivalent, and two during Research Proper after final oral presentation. Five hundred pesos shall be charged for the fourth and succeeding checks. Threshold of similarity is 25% and lower.

14. Manuscripts shall follow the institutional format and referencing style; the study must be a product of exercise of intellect, free of dishonesty and fraud. Also, the papers shall be subjected for Grammarly check to ensure adherence to rules of language and writing conventions. Threshold of grammar accuracy is 95% and above.

15. The researcher/s shall acknowledge the right of the College to disallow or reject researches found to be deficient, unsatisfactory or plagiarized in form or substance.

16. The researcher/s shall take precautionary measures to ensure that their paper will not be copied, replicated or duplicated by others.

17. The researcher/s shall transfer the ownership of the study to the College once the paper has been successfully defended and bound. After which no paper shall be published or presented without consent/approval from the College. Sharing and/or furnishing copies of the research paper to pertinent offices or parties is subject to the approval of the College and shall require a signed Non-Disclosure Agreement from the receiving parties.

18. The researcher/s shall present their research projects strictly during their scheduled oral defense. Failure to present as scheduled, the researcher/s will have to wait for another cycle of defense.

I affix my signature to signify that I read, understand, and conform to the items enumerated above.

Conformed:

EDWIN G. LIGAN JR
Signature over printed name of student

EDWIN P. LIGAN
Signature over printed name of parent/guardian

ENGR. KANOY ANGGA
Signature over printed name of adviser

The University of Mindanao

College of Engineering Education
2nd Floor, BE Building
Matina Campus, Davao City
Telefax: (082)296-1084
Phone No.: (082)300-5456/300-0647 Local 131

ENDORSEMENT FOR FINAL DEFENSE

Date: **April 23, 2024**

This is to endorse the research manuscript entitled: "**SMART-TRACE: A Combined RFID Databasing and WiFi-RSSI Fingerprinting Indoor Positioning for Disaster Management using Smartphone**" prepared and submitted by **Edwin Ligan Jr., Reyden Jenn Cagata,** and **Shen Wa Lai,** for Final Defense. The manuscript has been evaluated by the research personnel listed below and was found to be compliant with the quality standards as provided in the UM Research Manual.

	NAME OF RESEARCH PERSONNEL	SIGNATURE
Adviser	**ENGR. RANDY ANGELIA**	
Statistician	**ENGR. ROLIEVEN CAÑIZARES**	

Endorsed by:

ENGR. RANDY ANGELIA
Research Adviser

Noted by:

ENGR. JETRON J. ADTOON
College Research Coordinator

Approved by:

DR. CHARLITO L. CAÑESARES, PME
Dean, College of Engineering Education

The University of Mindanao

College of Engineering Education
2nd Floor, BE Building
Matina Campus, Davao City
Telefax: (082)296-1084
Phone No.: (082)300-5456/300-0647 Local 131

UNDERGRADUATE THESIS / RESEARCH / CAPSTONE
APPROVAL OF FINAL MANUSCRIPT

Date : June 14, 2024

Title : **SMART-TRACE: A Combined RFID Databasing and WiFi-RSSI Fingerprinting Indoor Positioning for Disaster Management using Smartphone**

Student-Proponents	Program
1. Reyden Jenn Cagata	ECE
2. Edwin Ligan Jr.	ECE
3. Shen Wa Lai	ECE

Panel Comments/ Recommendations	Previous Status	Actions Taken / Revisions	Page Reflected
Sequential References	The references' numbers are not in order	Corrected the order of the reference	1
5 Keywords in the Abstract	8 Keywords in the Abstract	Deleted 3 keyword in the Abstract	1
Abstract should be easily understood, use layman's term	The abstract is hard to understand	Used more common words so that it could be easily understand	1
Remove database as an objective because it is unnecessary to mention	There is an objective (a)	Removed objective (a)	1
Past tense in Methodology	Using "will" in the methodology	Used past tense to address that the study was already conducted	2-3
Mention specifications	Less specifications	Added more specifications	2
Trade-off analysis should be at the beginning of the Methodology	The trade-off analysis was placed at the end of the Methodology	Transferred the trade-off analysis to the beginning of the Methodology	2
Remove the figure for database at Chapter III	Figure 6 shows the database of the system	Removed figure 6	4
Show the RFID Process on how it been used in the study and combine the RFID reader section and Access Point Configuration section at Chapter II	RFID reader section and Access Point Configuration section at Chapter II were separated	At Chapter II, RFID readers section and Access Point Configuration section were combined and named it Design	2
Transfer all equations from Chapter III to Chapter II	Some of the equations are placed at Chapter III	Transferred all of the equations from Chapter III to Chapter II	4
Removed Table VI	Table VI shows the summary of the different Algorithms' Accuracy and run time	Removed the Table VI	7
Use IEEE Format for reference	Used APA format	Corrected the format and used the IEEE format	8
Transfer the flowchart to the appendix	Figure 5 shows the flowchart of the system	Transferred the flowchart to the appendix	4
Same font size throughout the paper	There is a different font size found on page 5	Corrected the font size	5
Add Chapter III: Simultaneous device with 10 trials using a confusion matrix	Not found on the paper	Add data by conducting another data gathering for six simultaneous devices in seven scenarios with a total of 14 trials, as shown in Table VI	7

The University of Mindanao

College of Engineering Education
2nd Floor, BE Building
Matina Campus, Davao City
Telefax: (082)296-1084
Phone No.: (082)300-5456/300-0647 Local 131

UNDERGRADUATE THESIS / RESEARCH / CAPSTONE
APPROVAL OF FINAL MANUSCRIPT

APPROVALS:

C Complied NC Not Complied	Thesis Adviser / Editor	Signature	Date
C	Engr. Randy E. Angelia		JUNE 20, 2024

C Complied NC Not Complied	Statistician	Signature	Date
C	Engr. Rolieven P. Cañizares		JUNE 21, 2024

C Complied NC Not Complied	Panel Members	Signature	Date
C	Engr. Jetron J. Adtoon		08-07-24
C	Engr. Elena Matillano-Perez		

The University of Mindanao

Research and Publication Center
Room H1B, DPT Building
Matina Campus, Davao City
Phone No.: (082)300-0640 loc. 111

Certificate of Plagiarism Check with turnitin

Type of paper:

- [x] Bachelor's thesis/capstone
- [ ] Master's thesis
- [ ] Doctoral thesis

Title of thesis:

SMART-TRACE: A Combined RFID Databasing and WiFi-RSSI Fingerprinting Indoor Positioning for Disaster Management using Smartphone

Name of author/s:

Reyden Jenn Cagata, Shen Wa Lai, Edwin Ligan Jr.

Summary of report:

turnitin Originality Report

Processed on:	August 16, 2024
ID number:	2432745986
Word count:	4,060
Submitted:	August 14, 2024

Similarity Index

8%

Examiner's statement:

I hereby certify that this thesis has been evaluated using the Turnitin Originality Check system. And I have analysed the report produced by the system. Based on it, I certify that the references in this thesis are in accordance with our University policy and good scientific practice.

Name of examiner:

Jetron J. Adtoon

Signature:

Date signed:

08/19/2024

Note: **Not valid without attachment**. *Please print the 1st page of Turnitin report for validation purposes*

Digital Receipt

This receipt acknowledges that Turnitin received your paper. Below you will find the receipt information regarding your submission.

The first page of your submissions is displayed below.

Submission author:	Jetron Adtoon
Assignment title:	Submission 1
Submission title:	SMART-TRACE: A Combined RFID Databasing and WiFi-RSSI F...
File name:	l-Manuscript-not-for-binding-IEEE-from-4-Without-Referenc...
File size:	7.43M
Page count:	7
Word count:	4,060
Character count:	21,684
Submission date:	16-Aug-2024 10:50AM (UTC+0800)
Submission ID:	2432745986

SMART-TRACE: A Combined RFID Databasing and WiFi-RSSI Fingerprinting Indoor Positioning for Disaster Management using Smartphone

Reyden Jem Cagata
Electronics Engineering Program
College of Engineering Education
University of Mindanao, Davao City,
Philippines
reydencagata@gmail.com

Shen Wa Lai
Electronics Engineering Program
College of Engineering Education
University of Mindanao, Davao City,
Philippines
shenwa.lai0910@gmail.com

Edwin Ligan Jr.
Electronics Engineering Program
College of Engineering Education
University of Mindanao, Davao City,
Philippines
edgaliasmilligan@gmail.com

***Abstract*—As a widely used device, smartphones have never been integral to disaster management. GPS might be a solution, but it is not accurate for indoors. Hence, SMART-TRACE used smartphone mobile hotspots to locate potential victims inside a building using an interactive map through a website. This study combined radio frequency identification (RFID) and WiFi technology to locate positions inside buildings, called the indoor positioning system (IPS). The parameter used in the study to predict the users is the Received Signal Strength Indicator (RSSI), commonly emitted by smartphones. Thus, these signal strengths were surveyed ahead of time, called fingerprinting, to create a database of signals and RFID tags from smartphone users. Additionally, this study evaluated SMART-TRACE's capability to identify possible victim locations on the map through several machine-learning algorithms. Among other position algorithms used in the study, the k-nearest neighbours (KNN) algorithm prevails in terms of its accuracy of 99.93% with its calculation time of 0.2s. Results also show a 100% accuracy of the UHF RFID reader with an average time reading interval of 0.02s. At the same time, WiFi-RSSI IPS covering three rooms and two floors has an overall accuracy of 97.22% in locating smartphones on the map with an average position error of 4.80m.**

Keywords—Indoor Positioning System (IPS), KNN algorithm, Controlled Access Point system Manager (CAPsMAN), crowdsourcing, multivariate linear regression analysis

I. Introduction

Knowing the whereabouts of every person inside a premises is essential during an emergency. GPS might be a solution; however, it is primarily applicable to outdoor environments as its signals cannot penetrate buildings due to their low signal strength and accuracy [1]. Also, rescuers must locate potential casualties before it is too late to provide an effective, quick emergency response. Hence, the researcher designed an indoor positioning system (IPS) using smartphone WiFi to address this discrepancy.

IPS is an approach that can locate positions inside a building. According to [2], there are several types of IPS technology in the radio frequency spectrum, such as WiFi, Bluetooth Low Energy (BLE), Radio Frequency Identification (RFID), and Ultrawideband (UWB). Based on the study of [3], UHF RFID has been reported to track users with low power consumption easily. However, RFID and BLE used for IPS have a short range, which is unsuitable for long-range applications [2, 4]. Also, UWB has a broader range; however, it requires higher costs due to its complex circuitry and availability [2, 5]. Among these, WiFi prevails in terms of its low cost, more comprehensive range than RFID and BLE, and the most accessible technology among other types of IPS [2, 6]. Furthermore, most of the research that studied IPS [7, 9, 11] utilizes Received Signal Strength Indication (RSSI) as the representative range-based positioning because of its simplicity and effectiveness. Other indicators used for IPS are time of arrival (TOA) [8], time difference of arrival (TDOA) [9], and angle of arrival (AOA) [10].

Consequently, three primary WLAN RSS techniques for locating clients are the Strongest Base Station, Propagation Model, and Fingerprinting-based method. The Strongest Base Station is the simplest method that estimates the user's location nearest to the Access Points (AP) [11]. Also, the Propagation Model, which includes trilateration and triangulation methods, requires at least three APs to locate an object location [2, 11]. This method is low-cost and straightforward; however, it has low accuracy in a large area and proved unreliable and unpredictable [4, 11-12]. In addition, these studies [4, 13-14] find that the fingerprinting-based method has more accurate results as it cannot be affected by path loss due to its ability to record long-term RSS data. In addition, these algorithms use two basic equations: the Bayes Rule [15], which estimates the location through a probabilistic method, and the Euclidean equation. Other relevant studies [6-7, 9, 11] show that applying matching algorithm increases the accuracy of the IPS such as Binary Relevance [16], Classifier Chain and Label Powerset [17], k-nearest Neighbor (KNN) [6-7, 9, 11], and Multilabel KNN (MLKNN) [18]. IPS can also be improved through a statistical technique such as Principal Component Analysis (PCA) through noise dimension reduction [19].

Despite the growing interest in IPS using WiFi-RSSI and RFID databasing, more research is needed on applying indoor localization in disaster management using WiFi integrated with RFID technology. Most studies only record attendance using RFID and do not account for indoor localization. Also, other studies account for IPS but do not account for users' information.

The study's general objective is to combine RFID databasing and WiFi-RSSI indoor positioning for disaster management using smartphones. Specifically, the objectives of this study are (a) to create an indoor positioning system using fingerprinting-based WiFi-RSSI, (b) to integrate both the RFID database and WiFi-RSSI indoor positioning for disaster management using a smartphone, and (c) to test its accuracy by conducting a functionality test using the appropriate statistical tool.

Moreover, this research study provides valuable data, especially for government agencies specializing in disaster risk reduction and disaster management. Also, private or commercial buildings such as schools, hospitals, or malls will benefit from the technology for enhancing disaster management. Through this, users will gain support to improve their safety during a disaster. Also, future researchers will benefit from the study that the researchers conducted.

Copyright 2024 Turnitin. All rights reserved.

grammarly

Report: 123SMART TRACE_Without GRAPH_Cagata,Lai, Ligan

123SMART TRACE_Without GRAPH_Cagata,Lai, Ligan

by Randy Angelia

General metrics

20,858	3,225	198	12 min 54 sec	24 min 48 sec
characters	words	sentences	reading time	speaking time

Score

97

This text scores better than 97% of all texts checked by Grammarly

Writing Issues

48	8	40
Issues left	Critical	Advanced

Plagiarism

This text hasn't been checked for plagiarism

Report was generated on Tuesday, Jul 23, 2024, 01:17 PM

Page 1 of 20

The University of Mindanao

College of Engineering Education
2nd Floor, BE Building
Matina Campus, Davao City
Telefax: (082)296-1084
Phone No.: (082)300-5456/300-0647 Local 131

Date: August 26, 2024

AUTHORIZATION LETTER

This is to authorize the University of Mindanao and adviser/co-author, **PROF. RANDY E. ANGELIA,** of the study entitled **SMART-TRACE: A COMBINED RFID DATABASING AND WIFI-RSSI FINGERPRINTING INDOOR POSITIONING FOR DISASTER MANAGEMENT USING SMARTPHONE** to present the paper in local, national or international research conferences; publish the paper in local, national or international research journals; and/or submit the paper for national or international intellectual property protection. It is therefore the responsibility of the adviser to ensure that the primary authors/inventors/makers/designers are given due recognition.

The Researchers

Edwin Ligan Jr.

Reyden Jenn Cagata

Shen Wa Lai