Use of Smartphone-based Indoor Positioning for Attendance

Student Name: Geetali Tyagi
Roll Number: IIIT-D-BTech-CSE-2012-037

Student Name: Shweta Sood
Roll Number: IIIT-D-BTech-CSE-2012-064

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Engineering
on November 19th, 2015

BTP Track: Engineering

BTP Advisor
Dr. Vinayak Naik

Indraprastha Institute of Information Technology
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **Use of Smartphone-based Indoor Positioning for Attendance** submitted by me for the partial fulfillment of the requirements for the degree of Bachelor of Technology in Computer Science & Engineering at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of **Dr. Vinayak Naik**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

……………………………... **Place & Date: ....**
(Geetali Tyagi & Shweta Sood)

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

.... **Place & Date: ....**
Dr. Vinayak Naik

Abstract

The problem of indoor positioning has been inquired into from various perspectives. However, most of the perspectives, be it SNMP traps, Bluetooth beacons, involve setting up external devices for gathering Wifi data. This makes them vulnerable to attacks and there is no guarantee of accuracy of location provided to the user and requires admin access to network hardwares. This encourages the need of an approach that allows for feedback and doesn't require setting up any external device. We looked at the current related research to find and compare existing techniques for indoor localization. We collected our dataset by logging Wi-Fi data from Academic Block of IIIT-D campus. The approach discussed allows any phone's Wi-Fi logs to be tested against a common training log by means of calibration. We then applied the proposed approach for smartphone-based indoor positioning and used various techniques to improve accuracy.

Keywords: indoor positioning, smart-phone localization, calibration, RSSI fingerprinting

Acknowledgments

We wish to express our sincere gratitude to Dr. Vinayak Naik , Assistant Professor, IIIT - Delhi, for providing us the opportunity to do our B.Tech project under his guidance. Besides setting a tone for the idea, the lenient, flexible and supportive supervision from him helped vitally in nurturing of the idea and the execution as well.

Work Distribution

This is the second semester of BTP. Last semester, our focus was on face identification and recognition in large group photos. This semester we worked on indoor location detection.
The work distribution among the team members is given below:

- Chapter 2: Geetali Tyagi
- Chapter 4, section 4.1: Shweta Sood
- Chapter 4, section 4.2: Geetali Tyagi
- Chapter 5: section 5.2: Geetali Tyagi
- Chapter 5: section 5.3,5.4,5.5,5.7: Shweta Sood
- Chapter 5: section 5.1, 5.6: Common
- Chapter 1,3,6,7,8: Common

Contents

Chapter 1

Introduction

1.1 Motivation and Problem Statement

Although there has been a lot of research on indoor positioning techniques, real-world deployment of these techniques is not considered. The suggested approaches require sophisticated hardware, manual user input and cost-intensive deployment. Further, the accuracies achieved by them are specific to certain environments and their accuracy reduces greatly when deployed elsewhere.
Our goal is to create a model for indoor-localization within the constraints of mobile-based hardware that can be deployed elsewhere.

A lot of bluetooth and Wi-Fi based techniques use data from wireless hardware such as SNMP traps and Bluetooth beacons. These approaches involve an additional cost of installation if infrastructure is not present in a real-world scenario. Further, they require administrative access to these network hardwares. Not only does this endanger the privacy of the users of the network but prevents accurate feedback of accuracy since the user is not directly involved.
We flip this system and put the onus of localization on the users and their mobile devices. Through a Wi-Fi fingerprint of the building, the users are able to locate their position by mapping the signals from their device. This allows for feedback and subsequent improvement of accuracy through algorithmic changes. Such a system would be useful in attendance systems where inter-classroom localization would need to be done.

The next question is - Keeping in mind these restrictions, how to model a system with appreciable accuracy?

1.2 Users

The users of the smartphone based indoor localization system can be students, faculty, for marking attendance in lectures. This can be used by anyone to know location indoors.

1.3 Constraints

Logging for site-survey on multiple days at multiple times required a lot of effort, is time-consuming and labor-intensive. However, it is inevitable for fingerprinting-based approaches, since the database is constructed by locationally labeled fingerprints from the training survey.

Chapter 2

Related Work

2.1 Aim of Literature Survey

This review aims at finding out the current scenario of mobile-based indoor localization:

- An initial study to find status of prior research in indoor positioning.
- A comparison of the techniques used and their applicability to our chosen scenario for indoor positioning.

2.2 Prior Work in Related Fields

The information related to the conditions of our scenario was found distributed in the literature in multiple fractions.

Outdoor navigation through geo-magnetic compasses using satellite triangulation has achieved high levels of accuracy. It is possible to track movement through the most basic smart phone devices. However, in indoor navigation this technique fails due to obstruction of line-of-sight to the device. It is still possible to map geo-magnetic field inside a building and create a magnetic fingerprint. As done by Chung et al. [1]. However, the fingerprinting and subsequent localization requires use of strong magnetic sensors, tilt sensors and a micro-processing unit whose capabilities cannot be replicated by inbuilt mobile sensors. Thus, indoor localization must rely on other indoor infrastructures such as wireless APs, RFIDs and Bluetooth beacons.

Wi-Fi based techniques include indoor triangulation using wireless access points such as the one followed in [2]. However, this technique is unable to accurately locate positions within the signal radii in AP dense environment since Wi-Fi signals bounce of walls and make line-of-sight triangulation difficult. This approach was refined by Mariakakis et al. [3] by using signals strengths to estimate direct path to APs and subsequently, a more accurate distance value. However, this requires detailed knowledge of the network including positions of APs and admin access to APs for signal strength calculations., whereas, our model can be trained without knowing the wireless AP layout.

A constraint of our approach was the time and effort for site-specific calibration for creation of training data set. Thus, this remains a bottleneck for the portability of our model to a different setup. Rai et al. [5] overcame this by building a system that uses accelerometer, gyroscope and Wi-Fi sensors to passively track the location of contributors through step-counts and log Wi-Fi for tracked location. This reduces the input for a new calibration to just the building layout. While using crowd-sourcing sufficient for mapping the RSSI is out of the scope of our project, we tried to decrease the inputs to our model to simply 31 key points rather than the entire building layout.

2.3 Selection and Inclusion of Literature

Our aim is to locate individuals devices in an indoor environment with maximum precision possible under the following constraints:

- The hardware used for training and testing should be contained within the average mobile device.
- Indoor localization is done in AP dense areas.

Using these constraints, similarities were drawn between the literature and our scenario to pick papers where some or one of these had been tackled.

2.4 Summary of Findings

- The initial study concluded that the research in mobile-based using indoor positioning faces various challenges that have not been fully resolved. The specific constraints of our scenario were not found in any paper.
- Some of the papers for mobile-based localization used some other hardware and sophisticated APs during the training phase to build the fingerprint of of the building.
- Some of the papers had models that required detailed knowledge of the building network and administrative access. Such a system is not portable which is a goal of our project.

Chapter 3

Solution

3.1 Methodology

In order to identify location of a smartphone accurately, without requiring access to an external server that logs data for us, with assurance of performance - certainty of correctness, we propose a solution that relies on user's own smartphone.

- A phone can see various BSSIDs (mac address for an access point), with varying RSSI values.
- We can use these RSSI values for location identification.
- This is achieved by utilizing the information provided to us by all the BSSIDs.
- We create a training dataset from a phone by generating wifi logs of all BSSIDs observed for a location.
- Now, for the phone whose location is to be known, we log data at that position.
- For BSSIDs common to both training and testing logs, we determine location by comparing RSSI values.
- For each common BSSID, we look at the RSSI values that are comparable in both training and testing logs and use the location provided by training log as the location determined for that particular BSSID. We use average of RSSI values for each BSSID.
- Location of the smartphone is the location determined by maximum number of BSSIDs. This increases our confidence in the predicted location and reduces the error as well.

Training BSSID	Location A	Location B	Location C
'74:a2:e6:3f:69:77'	Inf	Inf	Inf
'74:a2:e6:3f:69:78'	Inf	Inf	Inf
'74:a2:e6:3f:69:79'	Inf	Inf	Inf
'74:a2:e6:3f:69:7a'	Inf	Inf	Inf
'74:a2:e6:3f:69:7c'	Inf	Inf	Inf
'74:a2:e6:3f:69:7d'	Inf	Inf	Inf
'78:ba:f9:af:cb:12'	-54.6056	Inf	-83
'78:ba:f9:af:cb:13'	-54.4225	Inf	-85
'78:ba:f9:af:cb:15'	-55.6154	Inf	-82
'78:ba:f9:af:cb:16'	-55.4444	Inf	-83
'78:ba:f9:af:cb:17'	-54.4225	-86	-84
'78:ba:f9:af:cb:18'	-72.3429	Inf	Inf
'78:ba:f9:af:cb:19'	-71.7887	Inf	Inf
'78:ba:f9:af:cb:1a'	-72.9000	Inf	Inf
'78:ba:f9:af:cb:1c'	-74.0423	Inf	Inf
'78:ba:f9:af:cb:1d'	-72.6571	Inf	Inf

Test BSSID	Average RSSI
'00:c0:ca:79:21:66'	-106.4709
'2c:59:e5:da:35:ad'	-111.1981
'38:b1:db:a6:81:b3'	-106.4416
'3c:77:e6:53:2b:02'	-112.4421
'78:ba:f9:af:cb:12'	-45.5446
'78:ba:f9:af:cb:13'	-45.4802
'78:ba:f9:af:cb:15'	-45.7633
'78:ba:f9:af:cb:16'	-45.2003
'78:ba:f9:af:cb:17'	-46.3382
'78:ba:f9:af:cf:92'	-93.7820
'78:ba:f9:af:cf:95'	-91.2940
'78:ba:f9:af:cf:96'	-93.7820
'78:ba:f9:af:cf:97'	-91.2940
'c4:0a:cb:25:b5:a2'	-97.5140
'c4:0a:cb:25:b5:a3'	-97.5140
'c4:0a:cb:25:b5:a7'	-96.2700

Fig. 1. Common Average RSSI values of the training and testing logs for common BSSIDs.

In the figure given above, for the BSSID '78:ba:f9:af:cb:12' we see that the RSSI value is closest to Location A. Hence the predicted location by this BSSID is A. 'inf' denotes that the particular BSSID was not seen at that location. Final prediction is majority voting by all BSSIDs.

3.2 Architecture Diagram

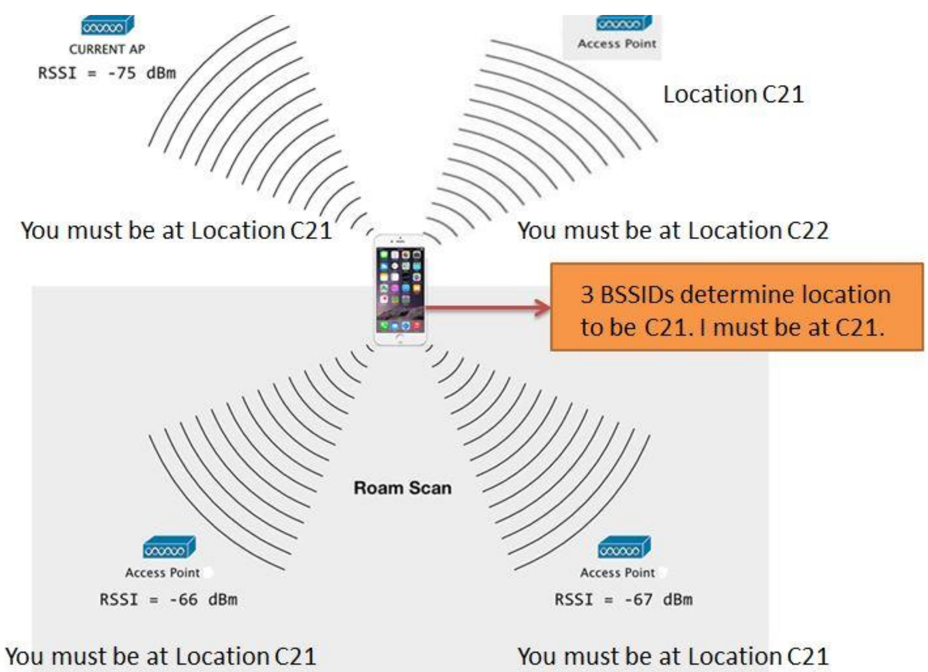


Fig. 2. Explaining Methodology

Chapter 4

Design

4.1 Location Detection

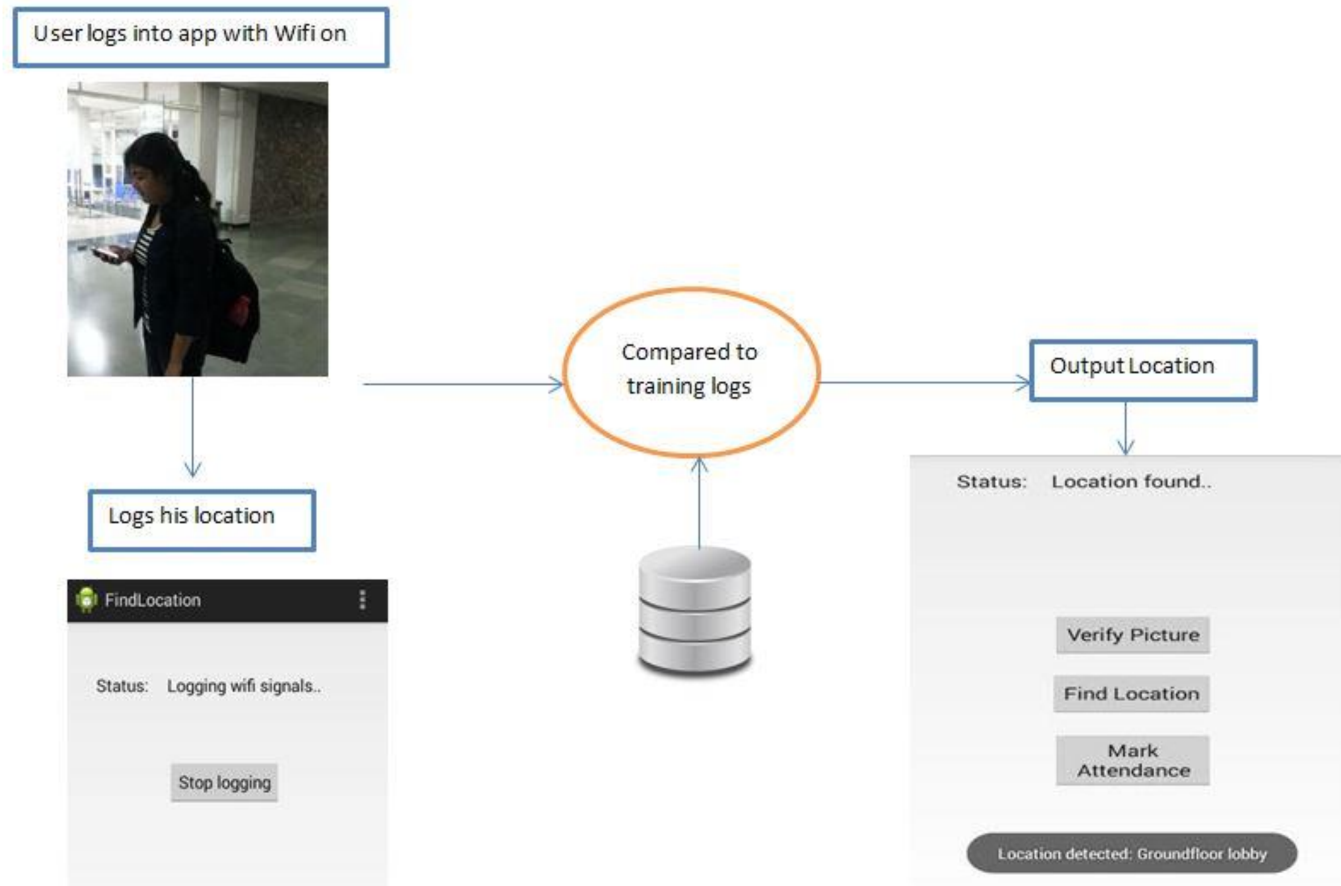


Fig. 3. Explaining Methodology

4.2 Attendance App

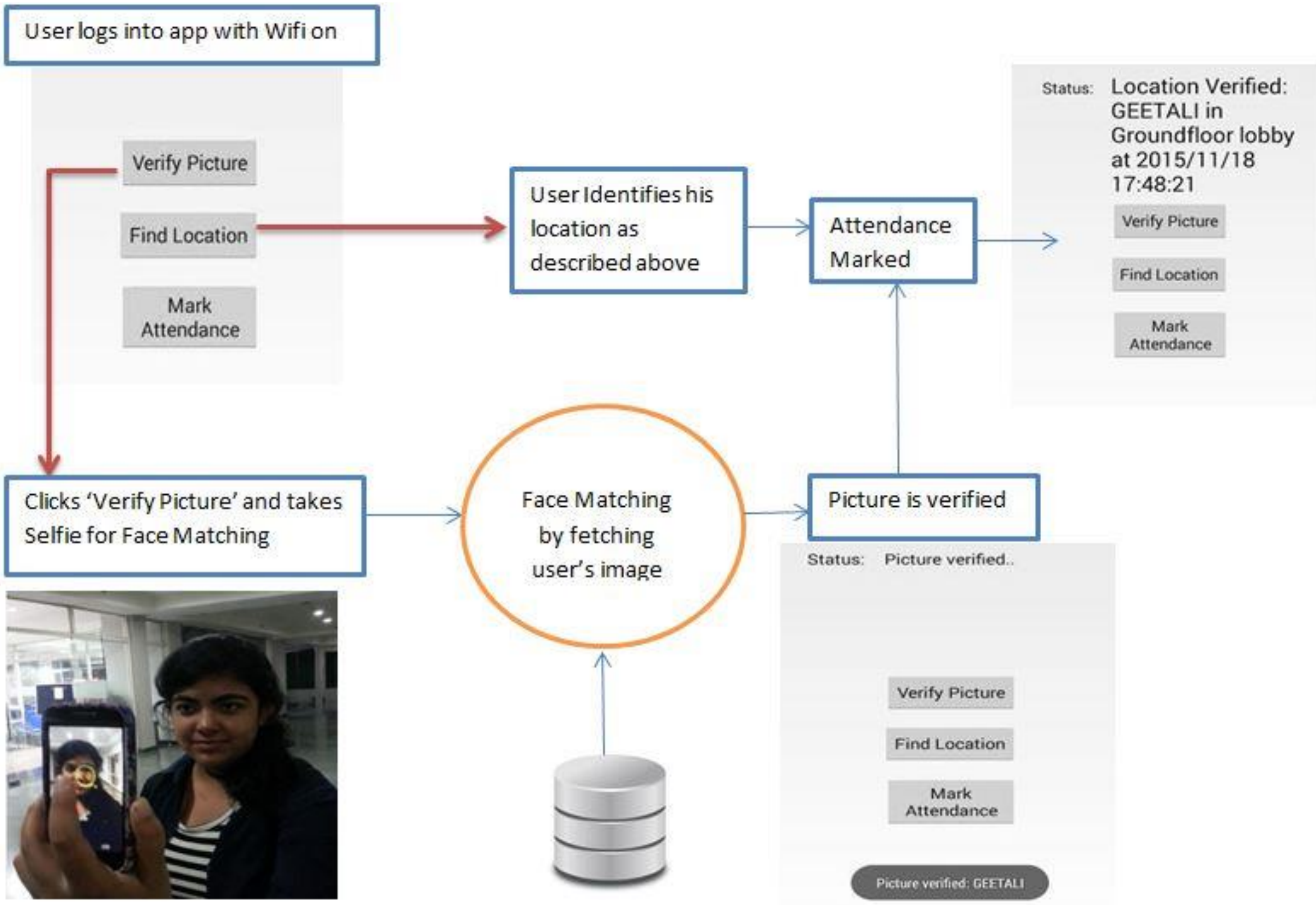


Fig. 4. Explaining Methodology

The above figure explains the attendance app. It combines the location identification module and face matching module to mark the attendance at the current time. Both the photo database for face verification (1:1 matching) and training logs are stored in the phone as of now.

Chapter 5

Implementations

5.1 Data Collection

We collected Wifi data in Academic Block. Logging was done several days at varying times from different smartphones to get a good sample of Wifi data. We stored the following values obtained from all the access points visible to the phone at a location:

- RSSI (Received Signal Strength Indication) - RSSI is an indication of the power level being received. Therefore, the higher the RSSI number, the stronger the signal. RSSI value is given in dBm.
- SSID (Service Set ID of the network) - Unique name for each WLAN.
- BSSID (Basic Service Set Identifier) - Packets bound for devices within the WLAN need to go to the correct destination. There are usually multiple access points within each WLAN. This is achieved through the identifier called BSSID, which is the mac address for the access points.

Time	Block	Mac Address	IP Address	RSSI	SSID	BSSID	Frequency
14-09-2015-07-17-24	1	5C:F8:A1:12:63:F2	192.168.160.137	-79	STUDENTS-M	c4:0a:cb:5c:0b:f7	2462
	1			-73	STUDENTS-M	c4:0a:cb:5c:0b:f7	2462
	1			-75	GUEST-N	c4:0a:cb:5c:0b:f2	2462
	1			-76	SENSOR	c4:0a:cb:5c:0b:f3	2462
	1			-56	SENSOR	c4:0a:cb:2d:97:13	2412
	1			-55	STUDENTS-M	c4:0a:cb:2d:97:17	2412
	1			-55	GUEST-N	c4:0a:cb:2d:97:12	2412
	1			-56	FACULTY-STAFF-N	c4:0a:cb:2d:97:15	2412
	1			-58	STUDENTS-N	c4:0a:cb:2d:97:16	2412
	1			-66	STUDENTS-M	c4:0a:cb:2d:97:18	5180
	1			-66	SENSOR	c4:0a:cb:2d:97:1c	5180
	1			-66	STUDENTS-N	c4:0a:cb:2d:97:19	5180
	1			-68	FACULTY-STAFF-N	c4:0a:cb:2d:97:1a	5180
	1			-71	GUEST-N	c4:0a:cb:2d:97:1d	5180
14-09-2015-07-17-29	1	5C:F8:A1:12:63:F2	192.168.160.137	-74	STUDENTS-M	c4:0a:cb:5c:0b:f7	2462
	1			-63	STUDENTS-M	c4:0a:cb:5c:0b:f7	2462
	1			-71	GUEST-N	c4:0a:cb:5c:0b:f2	2462
	1			-68	SENSOR	c4:0a:cb:5c:0b:f3	2462
	1			-56	SENSOR	c4:0a:cb:2d:97:13	2412
	1			-59	STUDENTS-M	c4:0a:cb:2d:97:17	2412
	1			-59	GUEST-N	c4:0a:cb:2d:97:12	2412
	1			-56	FACULTY-STAFF-N	c4:0a:cb:2d:97:15	2412
	1			-59	STUDENTS-N	c4:0a:cb:2d:97:16	2412

Fig. 5. Wifi Log for a location

The main Android and Java libraries used for creating this Wi-Fi logger app were import android.net.wifi.*, java.util.concurrent.*, android.app.IntentService.

Data collection was done over a period of 3 months. For the purpose of logging we stood still at 31 unique points in Academic Block.

The map of these points is shown in the figure below.

Floor	Place: <Mapped to Numerical value>							
Ground	Cdx: 1	Glassroom: 2	Lobby: 3	Classroom Lobby: 4	C01: 22	C02: 23	C03 :24	
First	A Wing: 5	B Wing: 6	Lobby: 7	Classroom Lobby: 8	C11: 25	C12: 26	C13: 27	
Second	A Wing: 9	B Wing: 10	Lobby: 11	Classroom Lobby: 12	C21: 28	C22: 29	C23: 30	C24 :31
Third	A Wing: 13	B Wing: 14	Lobby :15					
Fourth	A Wing: 16	B Wing: 17	Lobby: 18					
Fifth	A Wing: 19	B Wing: 20	Lobby: 21					

Fig. 6. 31 unique points in Academic Block

1. Training: For the purpose of training, we stood still at all the 31 points for 5 minutes. The logging from access points was done at an interval of 6 sec. We have discussed in section 5.2 the reason to choose a 6 sec interval.
2. Testing: For the purpose of testing, we stood still at all the 31 points for 2 minutes.

We have used a total of 3 smartphones for the logging exercise.

5.2 Identifying frequency of logging

Our training phase involves energy-intensive Wi-Fi scanning. In order to make our model more efficient, we tried to find the optimum rate of logging for which the accuracy while testing is not highly affected by downsampling.
The frequencies for logging tested were $3s^{-1}$, $6s^{-1}$, $9s^{-1}$, $12s^{-1}$, $15s^{-1}$, $18s^{-1}$.

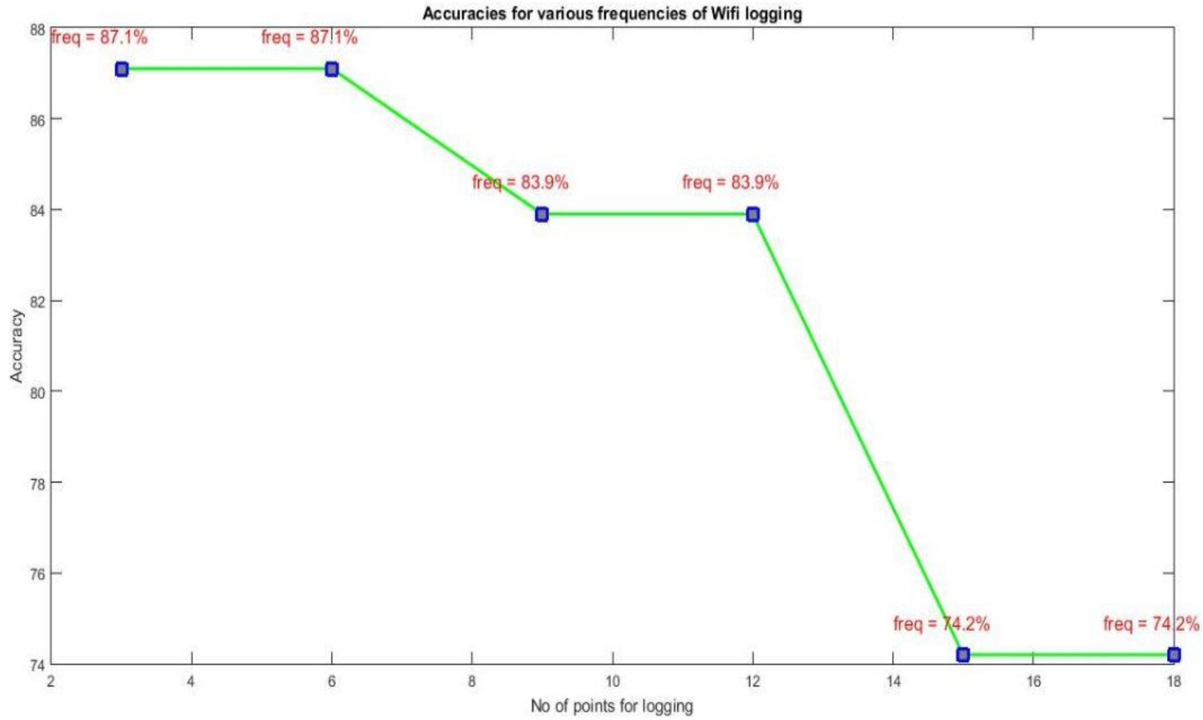


Fig. 7. Accuracies for different frequencies of logging.

As can be seen in the Fig. 8, while downsampling from 3s to 6s, there no change in the accuracies. Whereas, on further downsampling to 9s, accuracy takes a hit. So, the frequency used for final logging was $6s^{-1}$.

5.3 Training and testing from Same Phone

We used three smartphones for logging. After training (logging at 31 points), we tested these phones separately, against their respective testing datasets. Following results were obtained:

S.No	Accuracy
Phone 1 - 123 testing points	88%
Phone 2 - 144 testing points	79%
Phone 3 - 31 testing points	80%

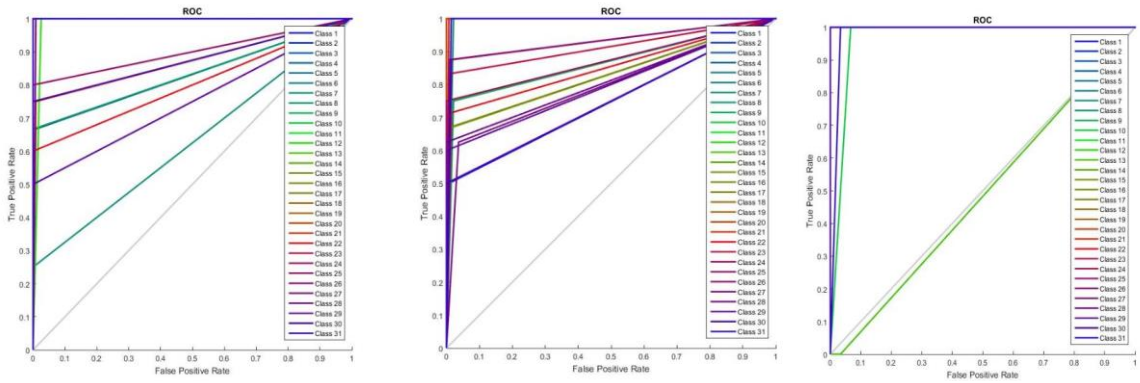


Fig. 8. Left to Right: ROC curves for Phone 1, Phone 2 and Phone 3 respectively for 31 classes (points)

5.4 Training and Testing from different Phones

Testing data of Phone 2 and Phone 3 was ran against training data of Phone 1.

S.No	Accuracy
Phone 2 - 134 testing points	34%
Phone 3 - 31 testing points	48%

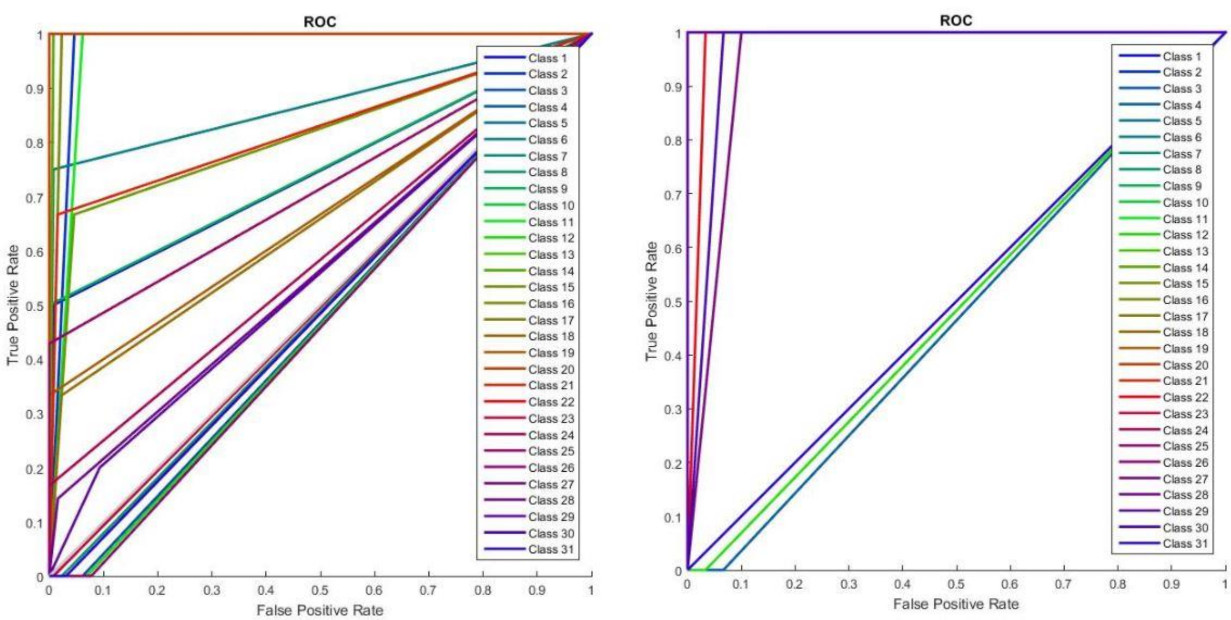


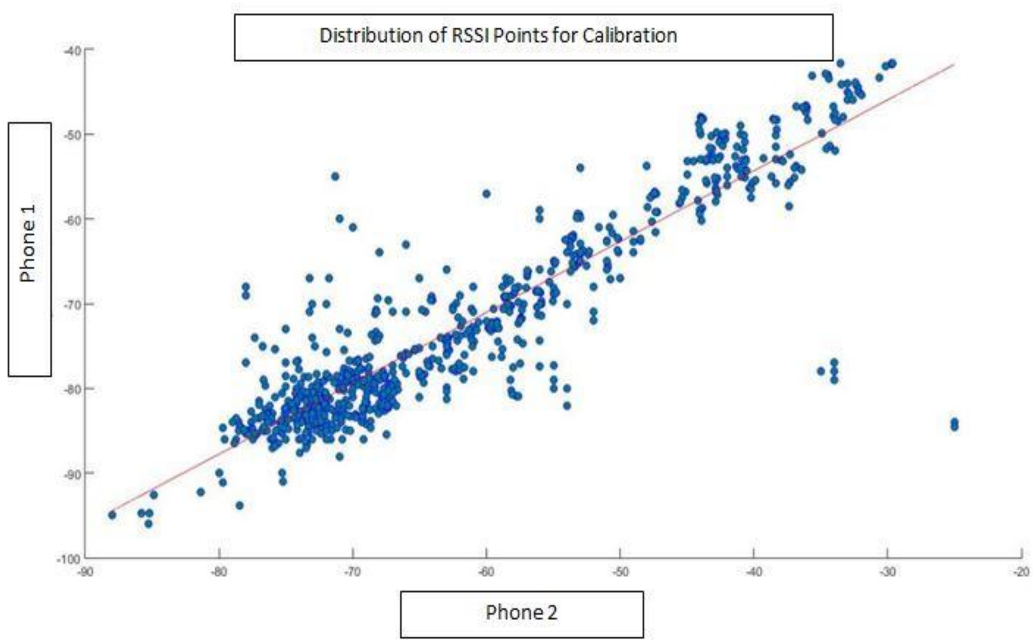
Fig. 9. Left to Right: ROC curves for Phone 2 and Phone 3 respectively for 31 classes (points)

This showed that accuracy reduced as soon as the training data was of a different phone. This was expected because the phones are of a different chip configuration. They are of a different model and make.

This showed the need of a calibration function that removes inconsistency in observed RSSI values due to the make of the phone and allows us to use testing data of a phone against training data of a different phone.

5.5 Calibration: Testing logs of a phone against training logs of another phone

In order to make location detection scalable, it is necessary that we have a common training dataset and use it for detecting location of a different phone. However, as described above, this requires us to map the values logged by a phone to another set of values that can be used against the training data set. To achieve the same, we tried to observe the distribution of RSSI values of phone 1 vs phone 2 and phone 1 vs phone 3 as shown in the figure below.



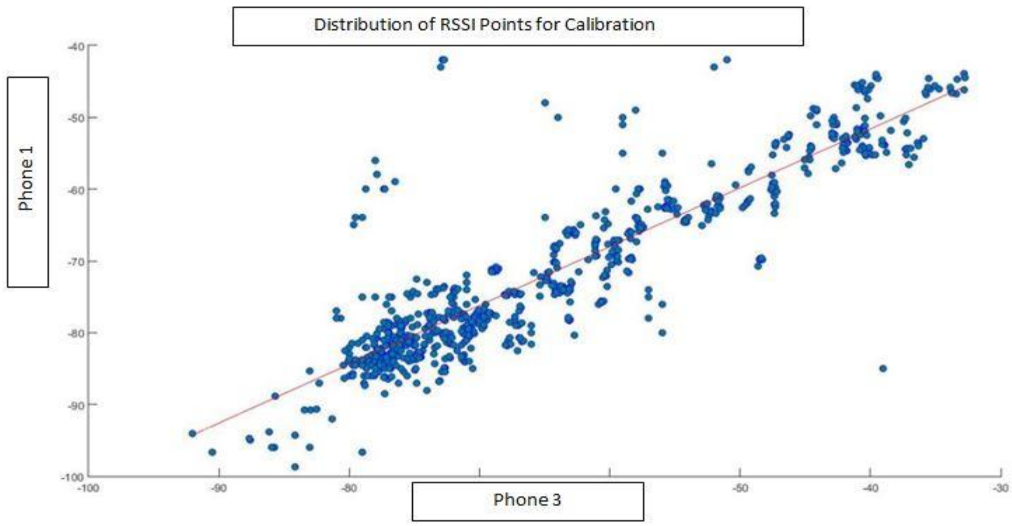


Fig. 10. Top to Bottom: Scatter Plots showing distribution of RSSI values for phone 2 and 3 respectively

As we can see, the distribution of the RSSI values for 2 different phones closely approximates a straight line (shown by red line in the Fig. 9). We know the equation of a straight line is :

$$y=mx+c$$

where m=slope

c=intercept

x=RSSI of phone

y=output RSSI

Once we fit a straight line to the distribution and obtained slope, intercept values, we mapped these values and proceeded the same way as discussed in section 3.1 to obtain the location.

Now the accuracy increased for both the phones as summarized in the table below.

Phone	Experiment	Accuracy
Phone 2	Without Calibration	34%
Phone 2	With Calibration	68%
Phone 3	Without Calibration	48%
Phone 3	With Calibration	71%

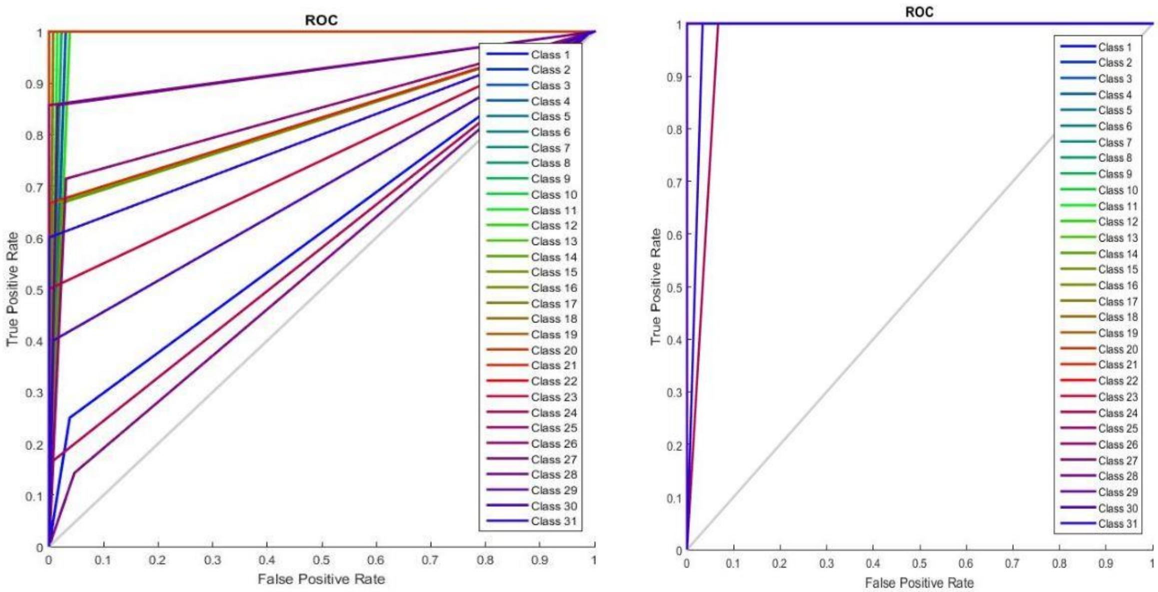


Fig. 11. Left to Right: ROC curves for Phone 2 and 3 respectively after calibration for 31 classes (points)

The table shows that calibration has significantly increased the accuracy for both the phones. Also, it is close to the accuracy obtained when testing against a phone’s own training data. For phone 2, it was 79% and phone 3 it was 80%.

5.6 Downsampling

5.6.1 Downsampling no of points by Intelligence

We tried to reduce the number of points required to obtain slope, intercept values for a line. We wanted to know how many points are good enough to achieve calibration. So, we reduce number of points to be sampled from 31 to 5 at a stepsize of 5. This was done intelligently, by eliminating the points that gave poor accuracy for test points in the following iteration.

The results are summarized in the following figures.

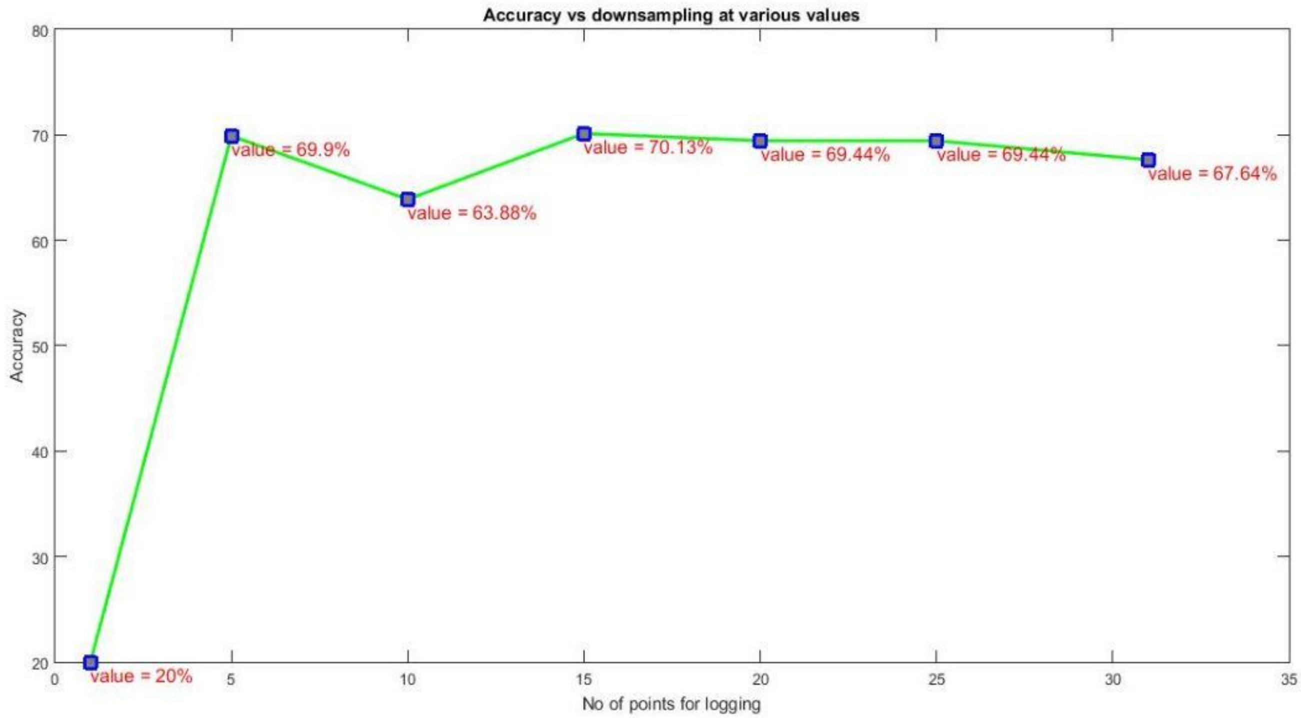


Fig. 12. Phone 2: Line Plot showing accuracy vs varying no. of points for logging

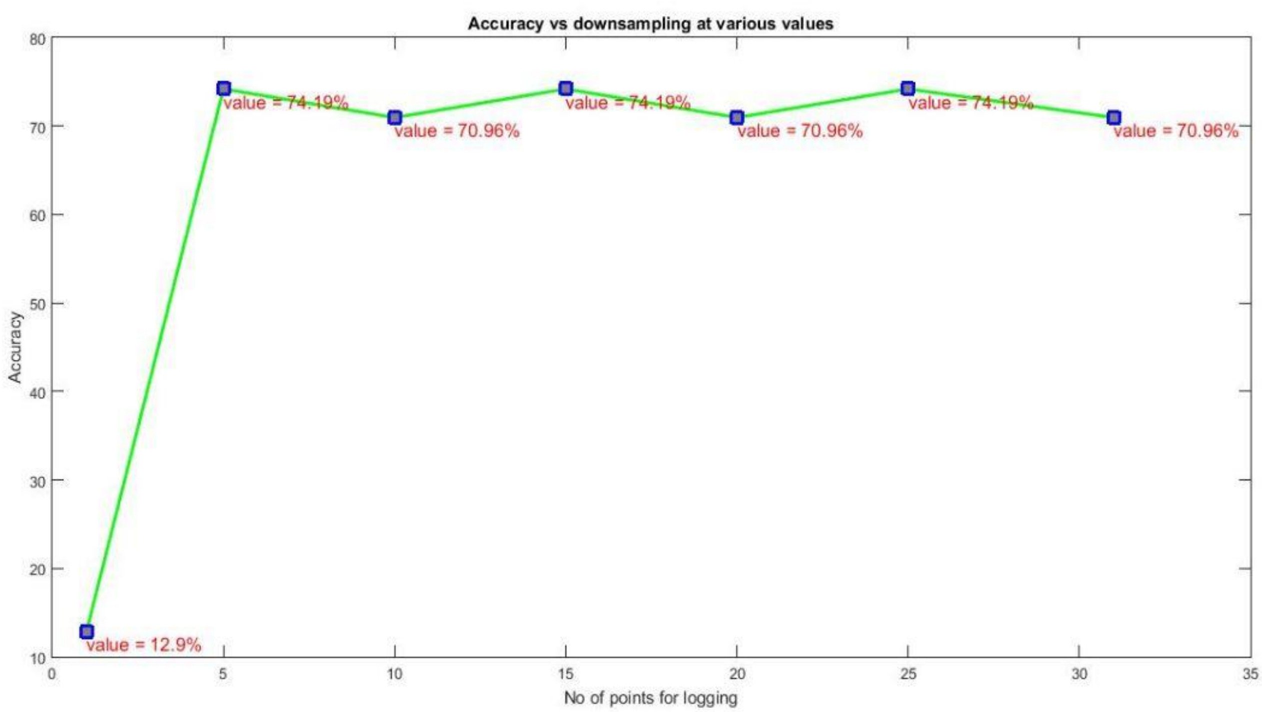


Fig. 13. Phone 3: Line Plot showing accuracy vs varying no. of points for logging

We can see in the above 2 figures that the accuracy observed is significantly good for logging at 5 and 15 points. Infact, it is higher than accuracy for logging at all the 31 points for calibration.
Thus, logging at 5 points should be sufficient to achieve good accuracy. This method looked at 5 points obtained after intelligent elimination of points based on accuracy in each iteration.

5.6.2 Downsampling number of points by Recursive Bagging

Our manual approach to downsampling is not deployable to new environments without user intervention. Thus, an alternate method is proposed to automate search for optimum number and group of points for calibration.
Starting from 31 locations, we randomly make 10 bags of 25 random locations and calculate their accuracy in calibration. The best out of these bags is recursively downsampled to 10 bags of 20 random locations and the same procedure is followed. This is done till 5 points are reached. The best accuracy bag out of all levels is taken for final calibration.
This method though automated, is more time-consuming than the manual scan of accuracy from a single run due to its recursive nature.
15, 5 points as in intelligent downsampling gave higher best accuracy than other point sizes as shown in Fig. 15, 16.

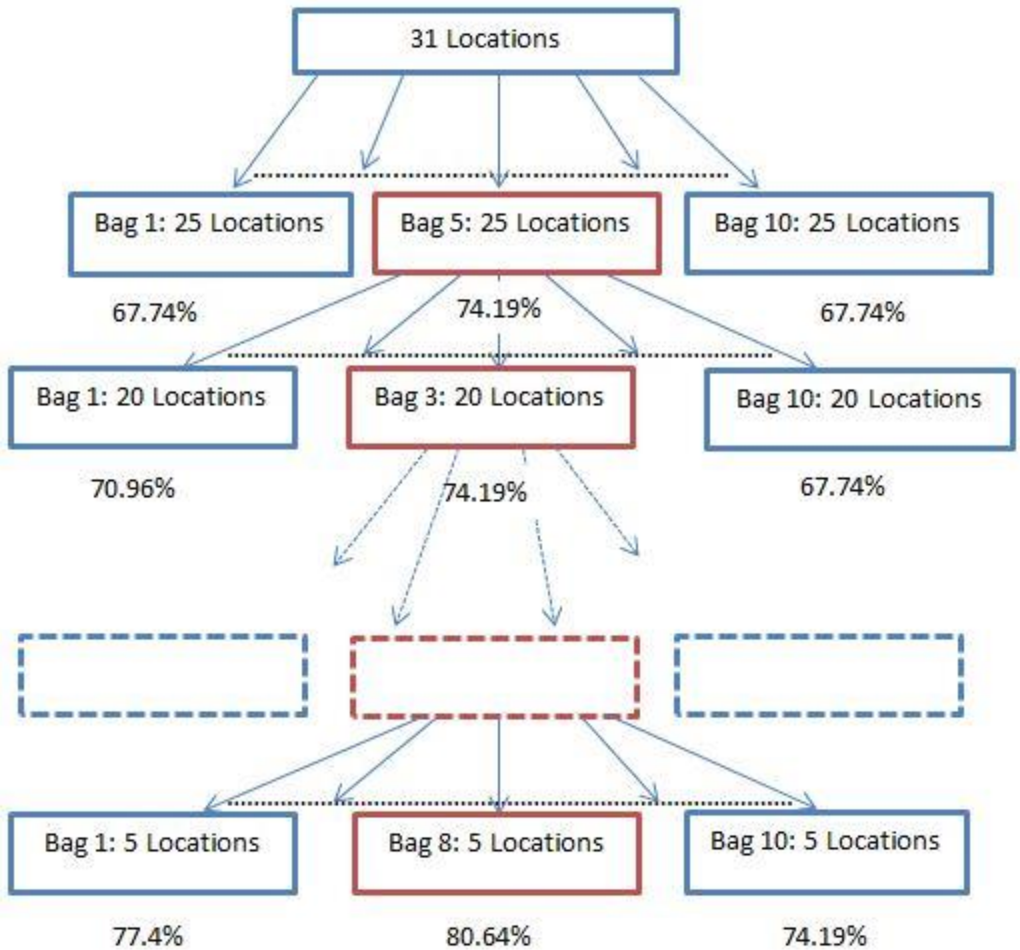


Fig. 14. Recursive random sampling to get best accuracies at each downsampling

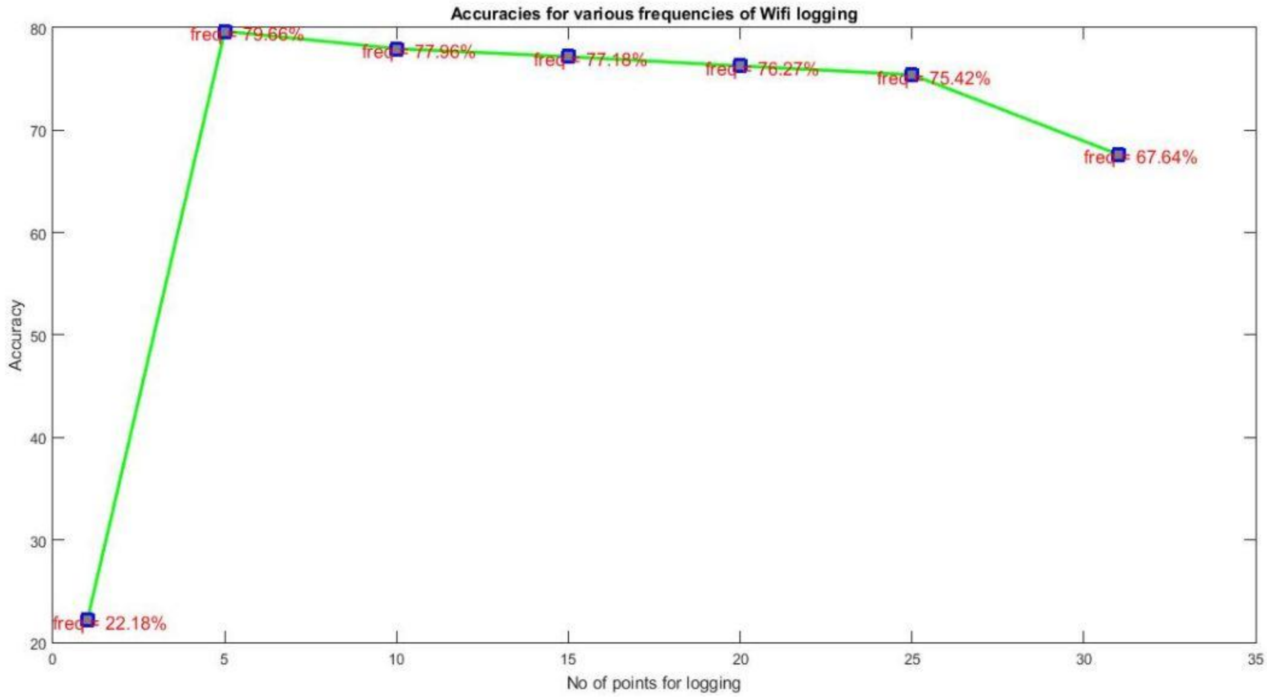


Fig. 15. Phone 2: Line Plot showing accuracy vs varying no. of points for logging

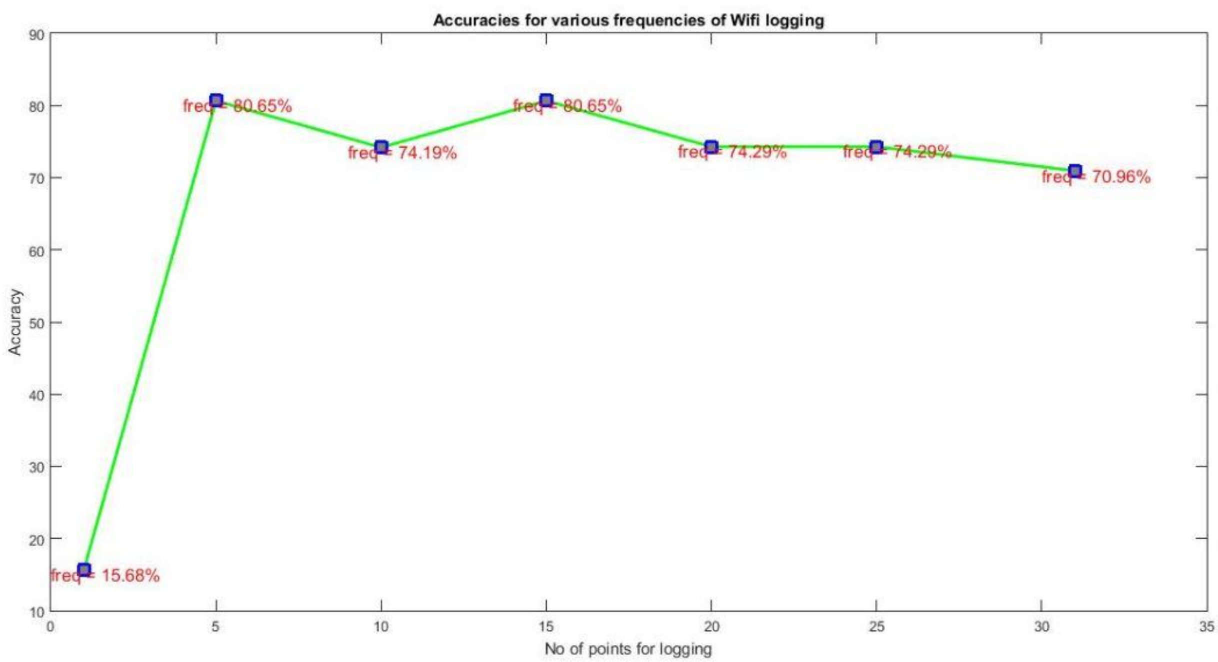


Fig. 16. Phone 3: Line Plot showing accuracy vs varying no. of points for logging

5.6.3 Other Proposed Solutions to identify points for calibration

The following solutions could be employed to reduce number of points for calibration:

a. Logging at points for Maximal Coverage of Academic Block - One may log at certain fixed no. of points on every floor such that they cover the entire floor and are uniformly distributed.
b. Logging at Frequently visited points - One may log at the frequently visited points on each floor. This is because those points become important for determining location.

5.7 Improving Accuracy by Eliminating outliers

For the purpose of calibration, we had fit a line on the distribution of RSSI values of 2 phones. We can see from figure Fig. 10 that there are many points that don't lie on the line and are quite far away from it. These points are outliers that are reducing the accuracy. To further improve the accuracy, we found distance of all the points from the line fitted. We found a cutoff distance for points by finding 80 percentile distance (80% of points are at a larger distance from the fitted line than the cutoff distance). We eliminated all the points with a distance larger than the cutoff distance. We did this for both phone 2 and phone 3 logged at 5 and 15 points using approach discussed in 5.6.1. We didn't try removing outliers after bagging as it is a computationally intensive approach and thus we will try to improve accuracy by intelligently downsampling points.

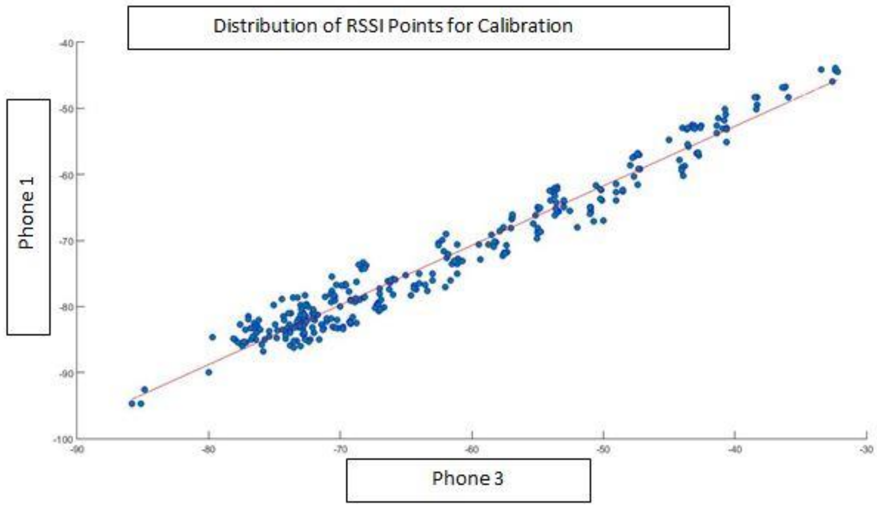


Fig. 17. Phone 3: Scatter Plot showing distribution of RSSI values after eliminating points using cutoff distance at downsample 15

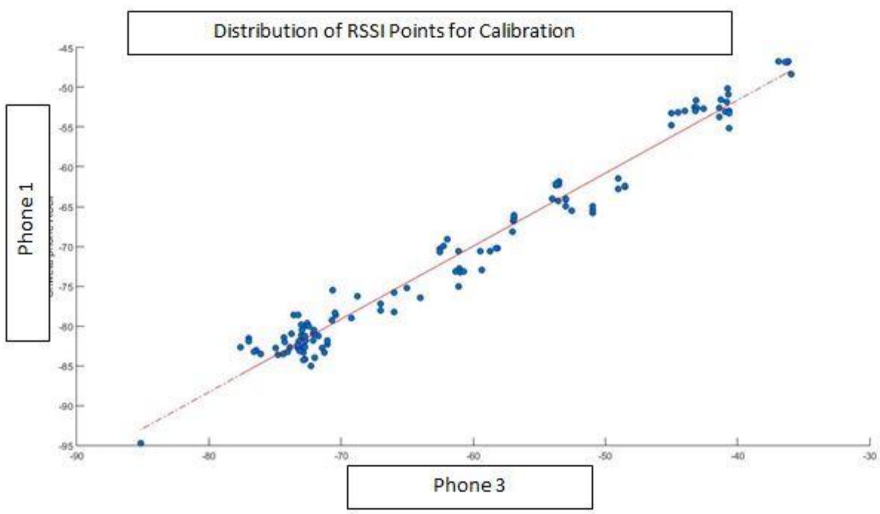


Fig. 18. Phone 3: Scatter Plot showing distribution of RSSI values after eliminating points using cutoff distance at downsample 5

As a result of this exercise, the accuracy did increase for both the phones as summarized in the table below.

Phone	Experiment	Accuracy
Phone 2	Downsample at 15	71%
Phone 2	Downsample at 5	72%

Phone 3	Downsample at 15	74%
Phone 3	Downsample at 5	77%

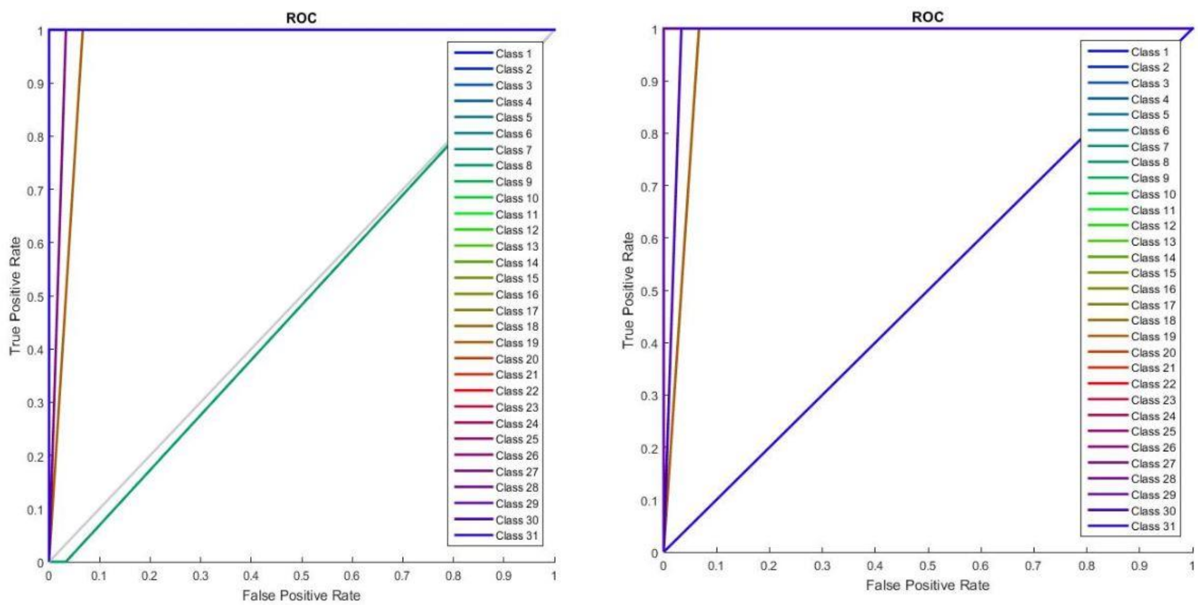


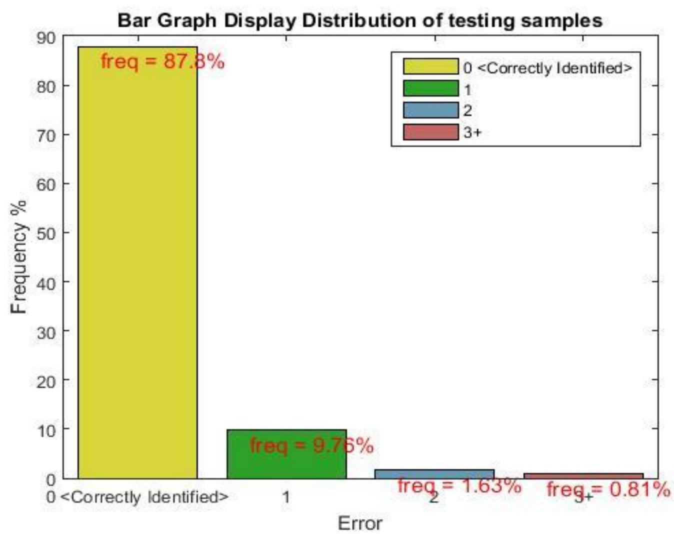
Fig. 19. Left to Right: ROC curves for Phone 3 after using cutoff distance at downsample 15, 5 points

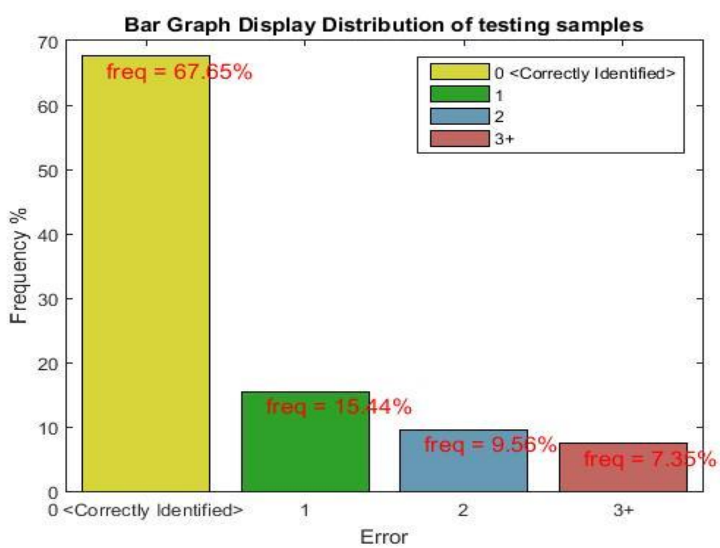
Chapter 6

End User Feedback and Conclusion

Summarizing the results obtained using error graphs. The bins denote error in prediction. If prediction is correct, bin is 0; if prediction is 1 block away from true block, error is 1; and so on; if it's 3 or more blocks away from true block, error is 3+:

- Phone 1, Phone 2 and Phone 3 tested against their respective training sets.





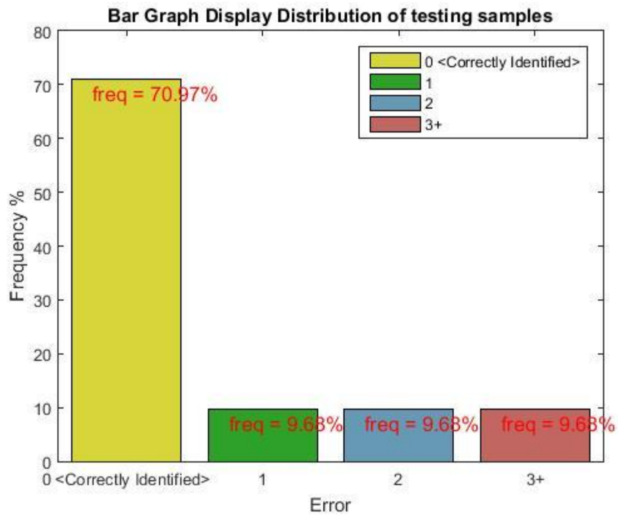
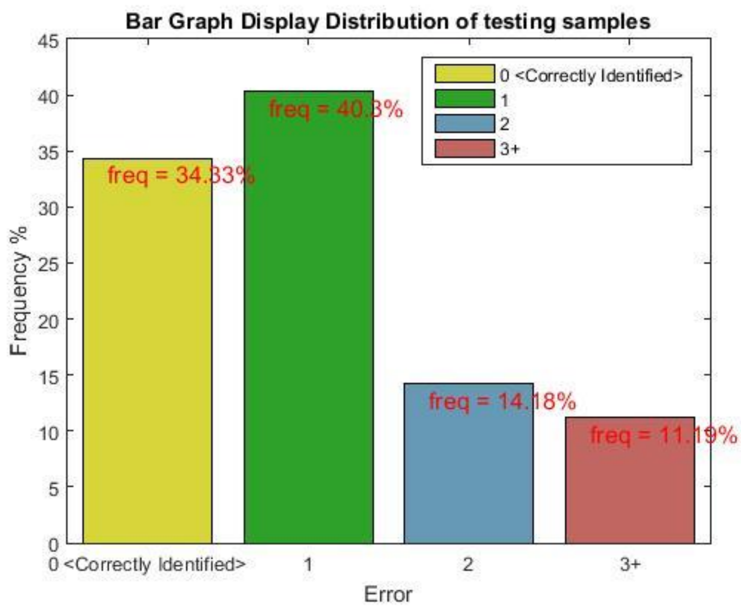


Fig. 20. Top to Bottom: Error in prediction for phone 1, 2, 3 respectively against their training data

- Training and testing from different phones.



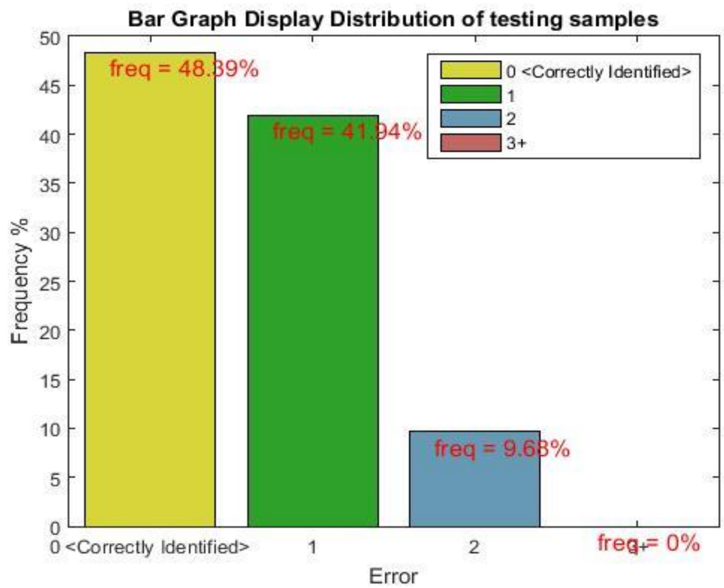
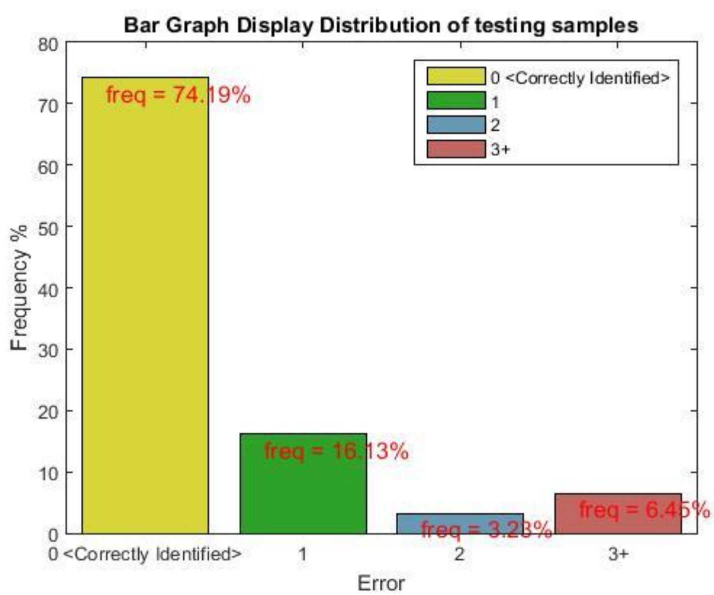


Fig. 21. Top to Bottom: Error in prediction for phone 2,3 respectively against phone 1 training data

- Downsampling no. of points by Intelligence.



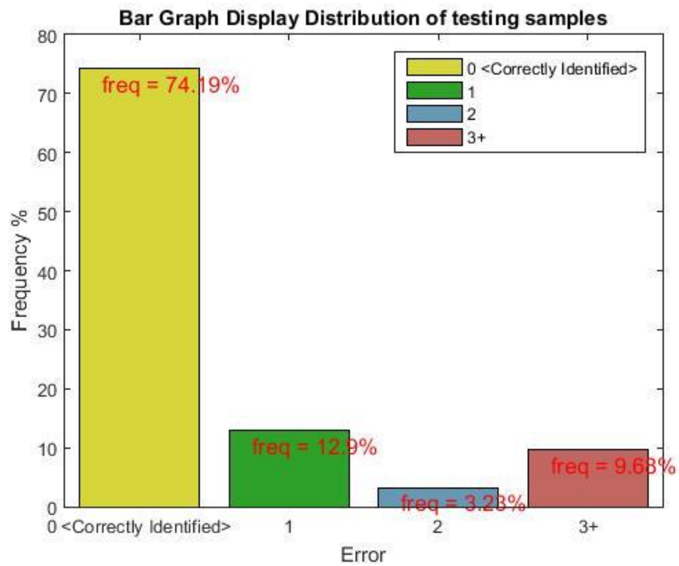
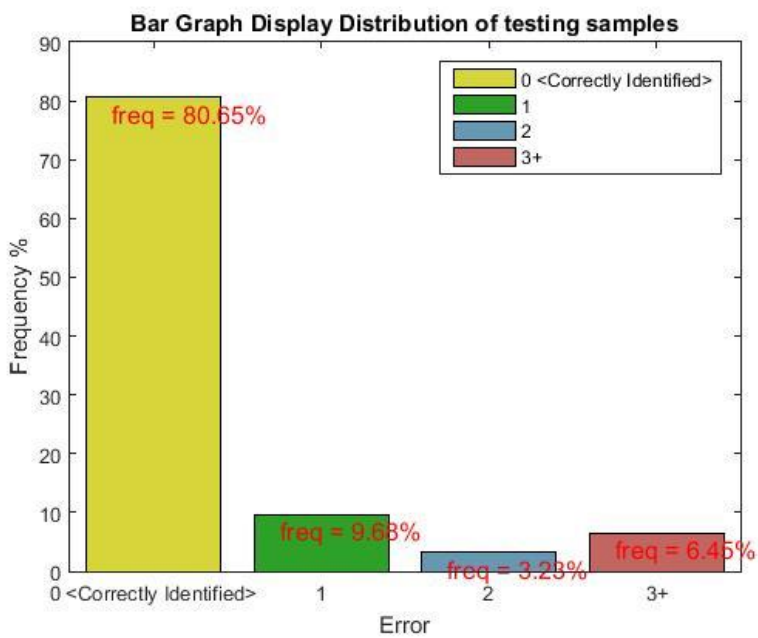


Fig. 22. Top to Bottom: Error in prediction for phone 3 at downsample 15, 5 respectively against phone 1 training data

- Downsampling no. of points by Bagging.



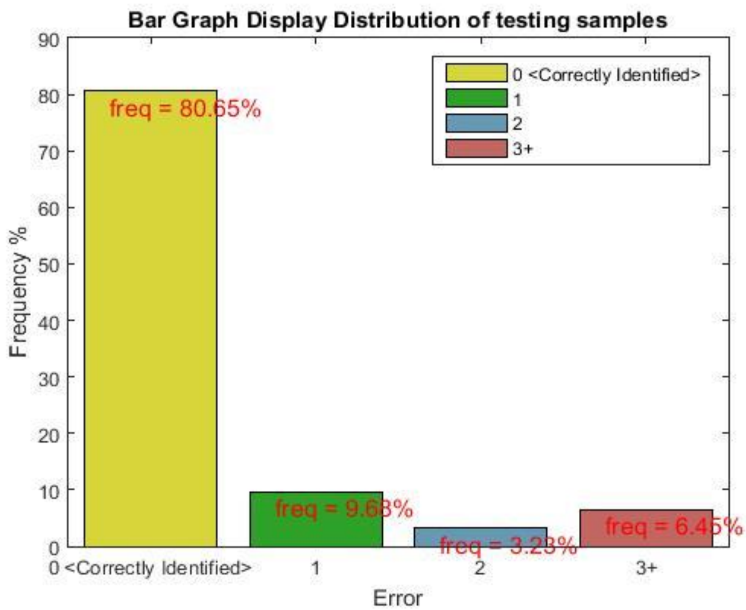
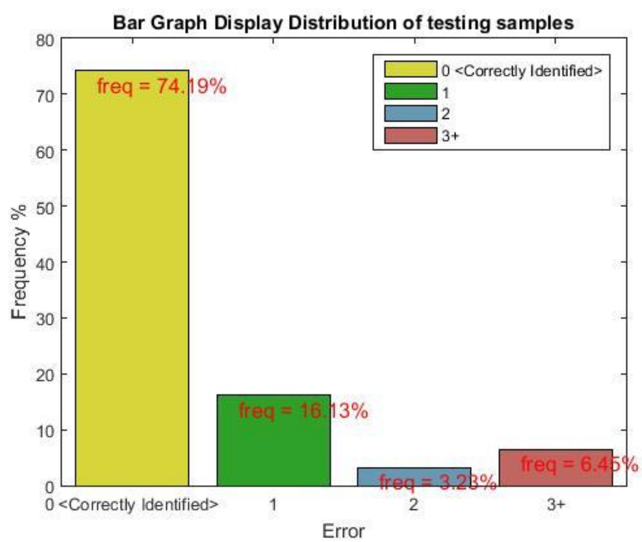


Fig. 23. Top to Bottom: Error in prediction for phone 3 at downsample 15, 5 respectively against phone 1 training data

- Improving Accuracy on intelligent downsampling by eliminating outliers



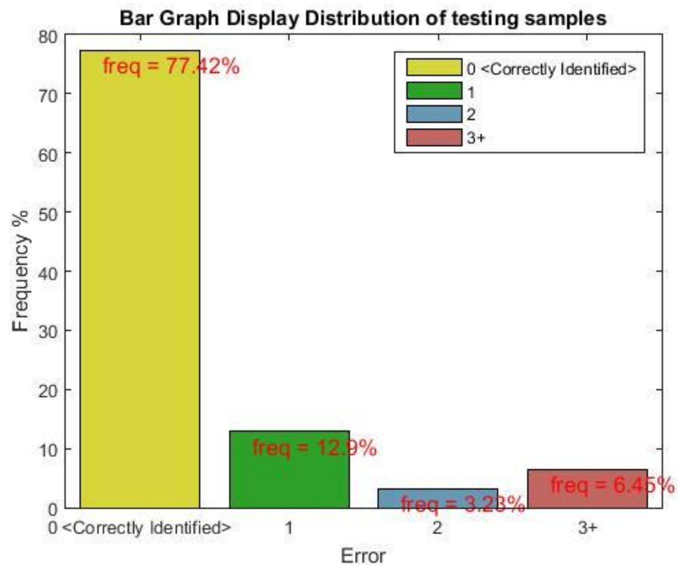


Fig. 24. Top to Bottom: Error in prediction for phone 3 after eliminating outliers at downsample 15, 5 respectively against training data of phone 1

- We can conclude that a good accuracy of indoor localization can be achieved for a phone against a common training dataset constructed by some other phone.
- We can also see that logging at 5 points is sufficient to achieve accuracy close to the one achieved when tested against the same phone training data. The approach of selecting those 5 points can be done by any of the proposed methods discussed in section 5.6
- Comparing the 2 approaches for downsampling, we feel that intelligently downsampling is better computationally as compared to bagging. However, in bagging, we find best accuracy by use of 10 bags at each iteration and use the bag giving best accuracy. Hence, accuracy is better. But it takes a long time to run.
- We tried to improve accuracy of intelligently downsampling by eliminating outlier RSSI values while calculating slope, intercept values for calibration.
- This technique is better than SNMP based techniques which require administrative access to these network hardwares. Since the entire model is run on the network end, accuracy cannot be precisely determined without user feedback. Further, user information may be leaked without the user's knowledge if administrative access is not secured properly.

Bibliography

References

[1] J. Chung, M. Donahoe , C. Schmandt , Ig-Jae Kim1, P. Razavai , M. Wiseman, Indoor Location Sensing Using Geo-Magnetism, MobiSys '11 Proceedings of the 9th international conference on Mobile systems, applications, and services

[2] P. Agarwal, M. Singh, S. Kumar Sharma, Wifi Positioning System Using WLAN Signal Strengths by Block, IJCSIT

[3] A. Mariakakis, S. Sen, J. Lee, Kyu-Han Kim, SAIL: Single Access Point-Based Indoor Localization, MobiSys '14 Proceedings of the 12th annual international conference on Mobile systems, applications, and services

[4] Z. Yang, C. Wu, and Y. Liu, Locating in Fingerprint Space: Wireless Indoor Localization with Little Human Intervention, Mobicom '12 Proceedings of the 18th annual international conference on Mobile computing and networking

[5] Anshul Rai, K. Chintalapudi , V. Padmanabhan, R. Sen, Zee: Zero-Effort Crowdsourcing for Indoor Localization, Mobicom '12 Proceedings of the 18th annual international conference on Mobile computing and networking