School of
Engineering &
Computing Sciences

Indoor Localization Framework with WiFi Fingerprinting

Project Member	Rajan Khullar
Project Advisor	Dr. Ziqian Dong
Semester	Fall 2016

Introduction

Abstract

As with most studies, the results of WiFi fingerprinting are much more meaningful if there is a large sample size. For this project I created an Android app that helps researchers in this field build their dataset efficiently. After collecting two weeks worth of data I studied how many access points are ideal to predict a user's location from a single WiFi scan.

Background

Global Positioning Systems (GPS) have been the standard technology used to obtain location of electronic devices. GPS devices are essential for the functionality of many of today's devices, appearing in vehicles, drones, and smart phones. They have a wide range of applications, from helping people navigate the roads to assisting the military in navigating planes and drones. Moreover, the popularity of smartphones with their inbuilt GPS devices has made GPS easily available.

Unfortunately localization using GPS is ineffective indoors and in highly metropolitan environments because of GPS signal fading. Signal fading occurs when signals penetrate building materials causing a decrease in intensity. This causes a decrease in signal-to-noise ratio making it difficult for GPS devices to differentiate between signals and noise. Signal fading is also a result of multipath phenomenon, which is caused by reflection and refraction of signals when they encounter walls [1]. Because of these reasons GPS signal is often lost entirely on smartphones in indoor environments, causing a pressing need for indoor localization.

Indoor Localization may be utilized similarly to outdoor localization. This includes commercial applications such as real-time maps to help people navigate within malls or museums, and more technical applications such as guiding drones through indoor environments or locating cars inside tunnels. An indoor positioning system will open a new market of applications.
Indoor Localization may be utilized similarly to outdoor localization. This includes commercial applications such as real-time maps to help people navigate within malls or museums, and more technical applications such as guiding drones through indoor environments or locating cars inside tunnels. An indoor positioning system will open a new market of applications.

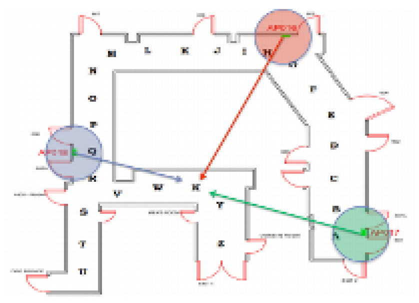
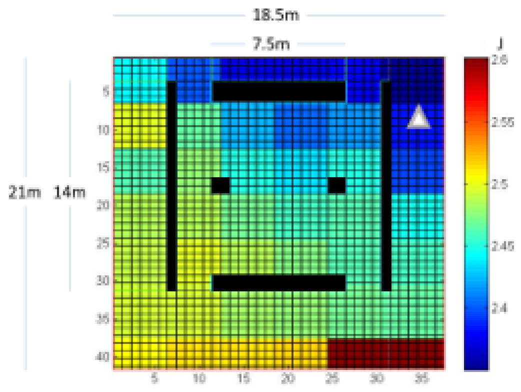
	

Figure 1.1	***Figure 2.1***

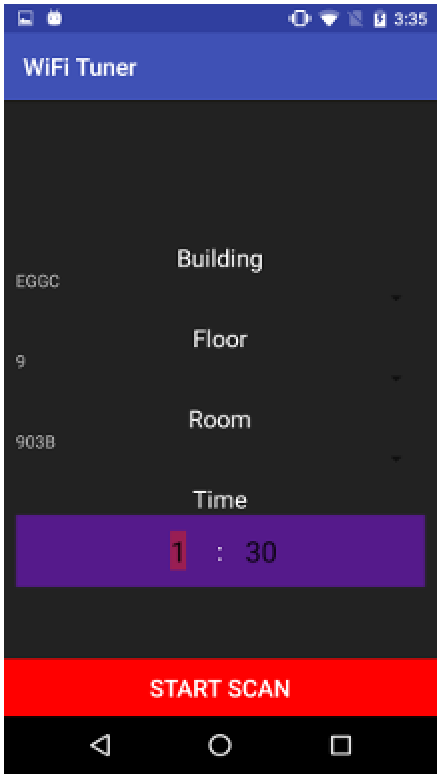
Related Work

Indoor Localization has been attempted through the use of Wi-Fi signal strength and Sensor fusion. Kothari et al (2012) [2] have successfully used dead reckoning and Wi-Fi signal strength fingerprinting to find the location of a smartphone. Dead reckoning was their method of using the accelerometer, gyroscope, compass and a particle filter in order to track walking and thereby track location. Both of these methods are prone to large errors. Wi-Fi signal strength is affected by obstacles and by the myriad of other Wi-Fi signals in an urban environment, while the accelerometer and gyroscope sensors are likely to generate random noise in the data.

Thanks to the National Science Foundation's Research Experience Undergraduate (REU) program, research fellows have been able to study indoor localization at NYIT. As shown in Figure 1.1, students in the summer of 2013 chose three access points on a single floor and measured signal strengths from twenty six spots evenly distributed in the hallways.

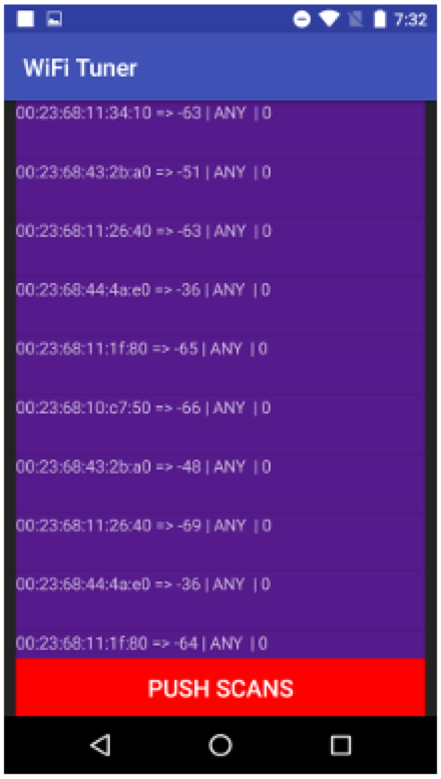
In Summer 2015 a student from Cooper Union and I were two of the REU fellows. We attempted to correlate energy consumption of a Nexus 5 to physical distance from an access point. We setup one router as a fixed access point and designed an Android application that records the current and voltage of the phone while pinging the router. We took ten samples at sixty points in a large classroom, however we were unable to find any significant correlation as shown in Figure 1.2.

In Summer 2016 two fellows studied multifloor localization with four consecutive floors in NYIT's main building. They were able to distinguish between thirty locations in their dataset with high accuracy. In all three REU studies the process of data collection proved difficult. The third project had a sample size of around 300 WiFi scans. After realizing this I was inspired to create a framework to help automate the process of gathering samples.



Building
EGGC
Floor
9
Room
903B
Time
1 : 30

Figure 2.1



```
00:23:68:11:34:10 => -63 | ANY  | 0
00:23:68:43:2b:a0 => -51 | ANY  | 0
00:23:68:11:26:40 => -63 | ANY  | 0
00:23:68:44:4a:e0 => -36 | ANY  | 0
00:23:68:11:1f:80 => -65 | ANY  | 0
00:23:68:10:c7:50 => -66 | ANY  | 0
00:23:68:43:2b:a0 => -48 | ANY  | 0
00:23:68:11:26:40 => -69 | ANY  | 0
00:23:68:44:4a:e0 => -36 | ANY  | 0
00:23:68:11:1f:80 => -64 | ANY  | 0
```

Figure 2.2

Implementation

The Digital Ocean server has Apache and PostgreSQL installed. A python library called Flask was used to create a REST api. Java was used to create the Android application. As shown in Figure 2.1, once users sign up and login they can choose their classroom and setup a scan for the duration of that class. The scans will occur in the background so the users can close the app and use their phone normally. The scans can be paused or canceled in case the users needs to change their location. Finally once the scans are complete, then each user can upload their local dataset to the server.

Database

One WiFi scan or sample results in one or more scan records. Each scan record contains information about the mac address to one access point, the signal strength to that access point, the location (building, floor, room), the unix time stamp, and the user who performed the scan.

In order to reduce redundancy a table of unique access points is maintained as well as one for unique locations. The actor table contains information for all people in the system including normal users, administrators, and new unverified signups. Figure 3 shows the simplified entity relation diagram.

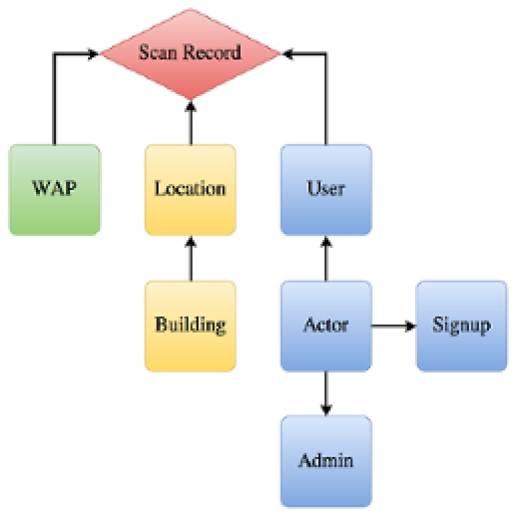


Figure 3

Preprocessing

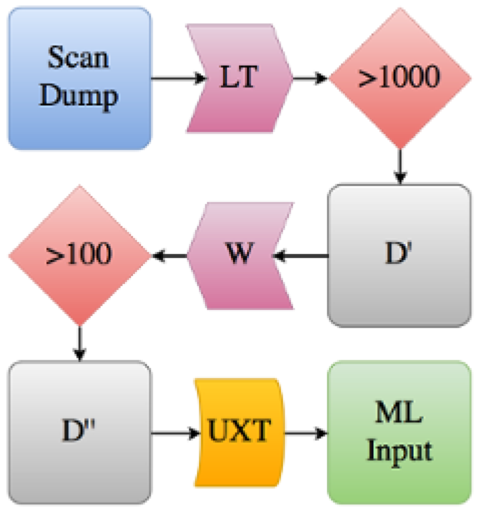


Figure 4

First the table of scan records is downloaded from the server by an administrator. The program groups each record by location and hour. Groups that do not have at least 1000 records are ignored. Each passing group is further grouped by the WiFi access point into blocks. Each block must contain at least 100 records. Then all the passing access points become columns and the passing records are combined by their timestamps into complete scans. The day of week and hour are also extracted from each timestamp. The new table serves as input for the machine learning algorithms.

Input

UXT	BSSID	Signal Strength	Location
T1	W1	*W*	L1
T1	W2	*X*	L1
T2	W2	*Y*	L2
T2	W3	*Z*	L2

Output

DoW	Hour	W1	W2	W3	Location
dow(T1)	hour(T1)	*W*	*X*	-	L1
dow(T2)	hour(T2)	-	*Y*	*Z*	L2

Results and Analysis

Group Cardinality

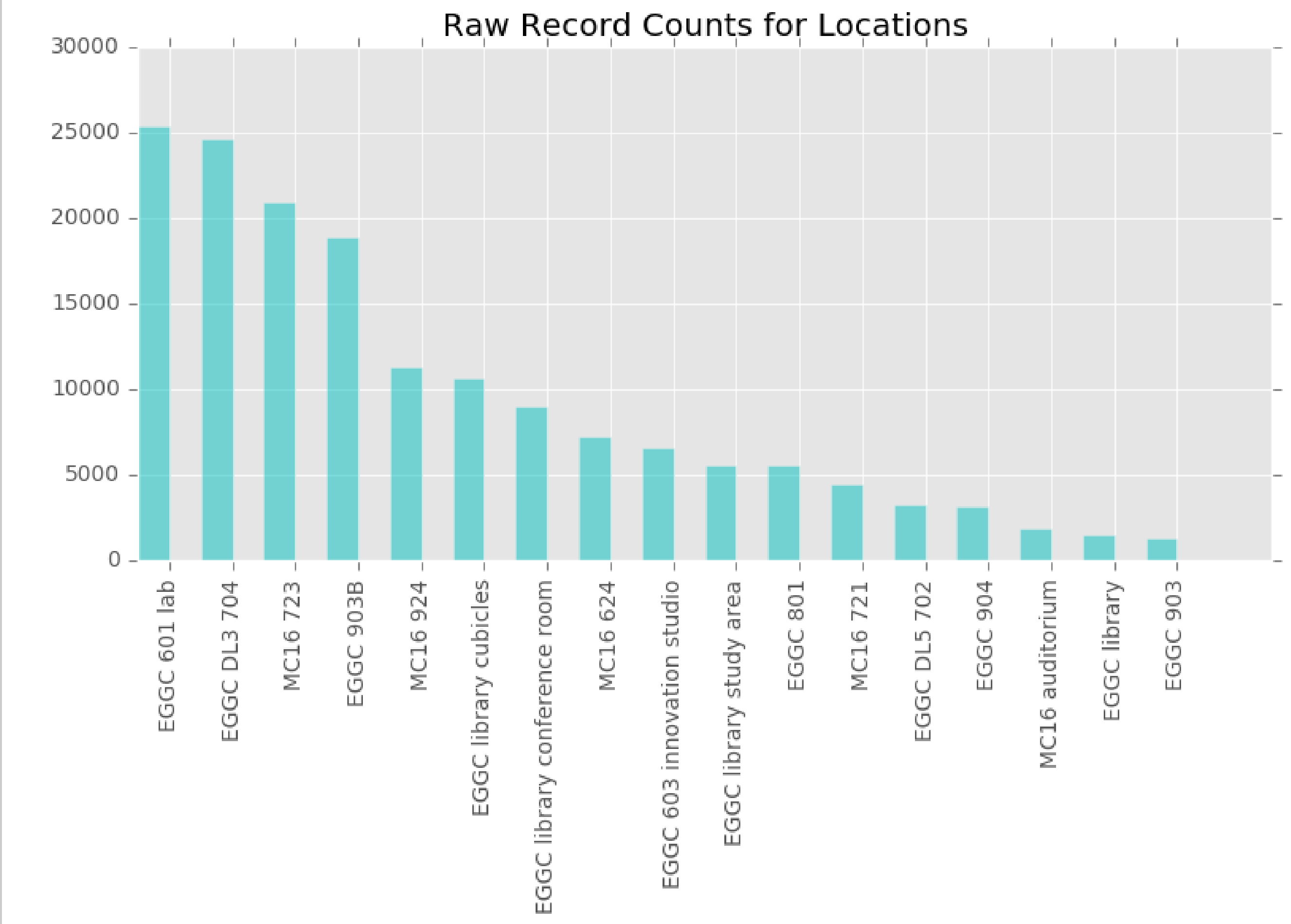


Figure 5.1

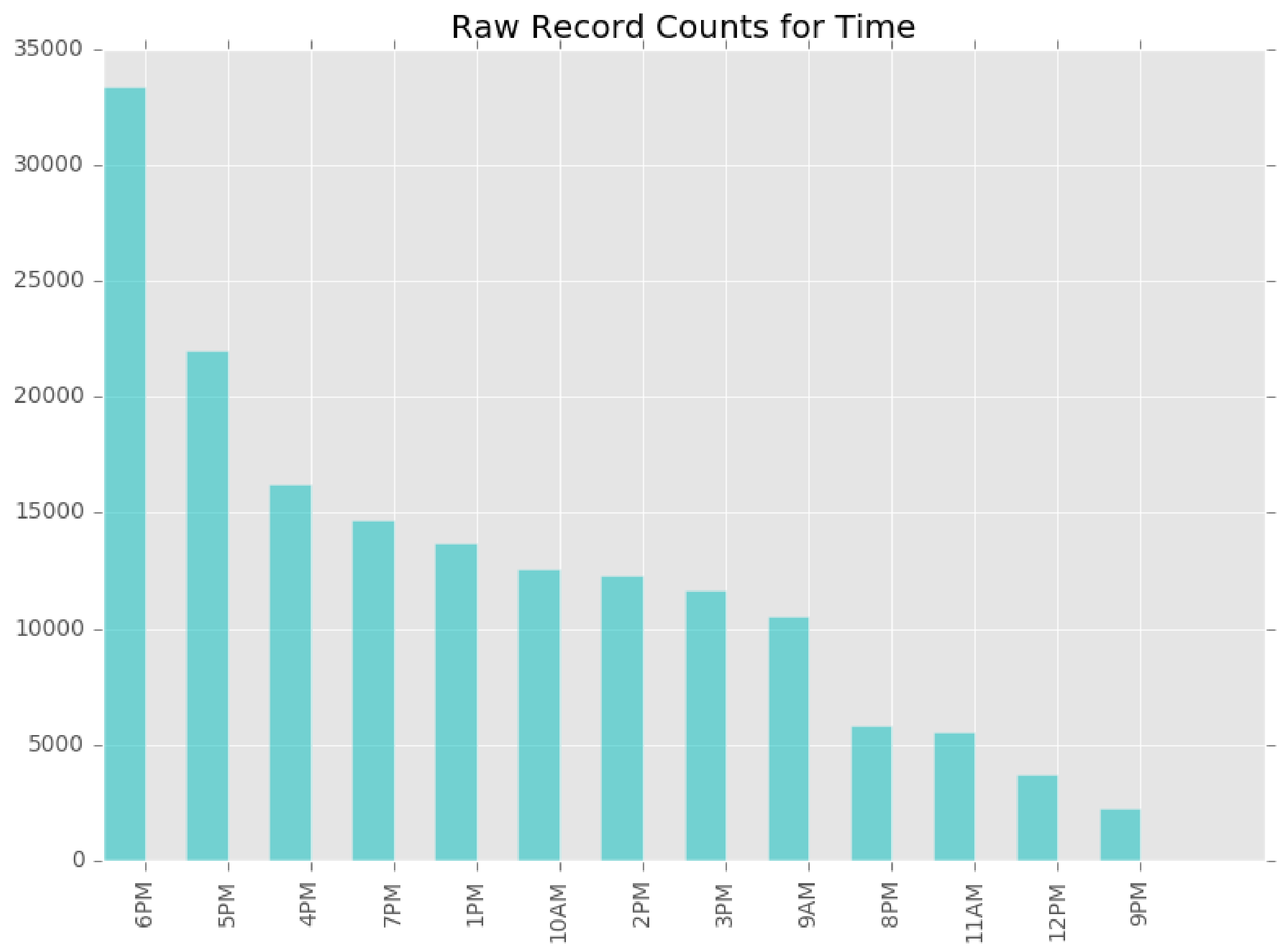

Raw Record Counts for Time
35000
30000
25000
20000
15000
10000
5000
0
6PM
5PM
4PM
7PM
1PM
10AM
2PM
3PM
9AM
8PM
11AM
12PM
9PM

Figure 5.2

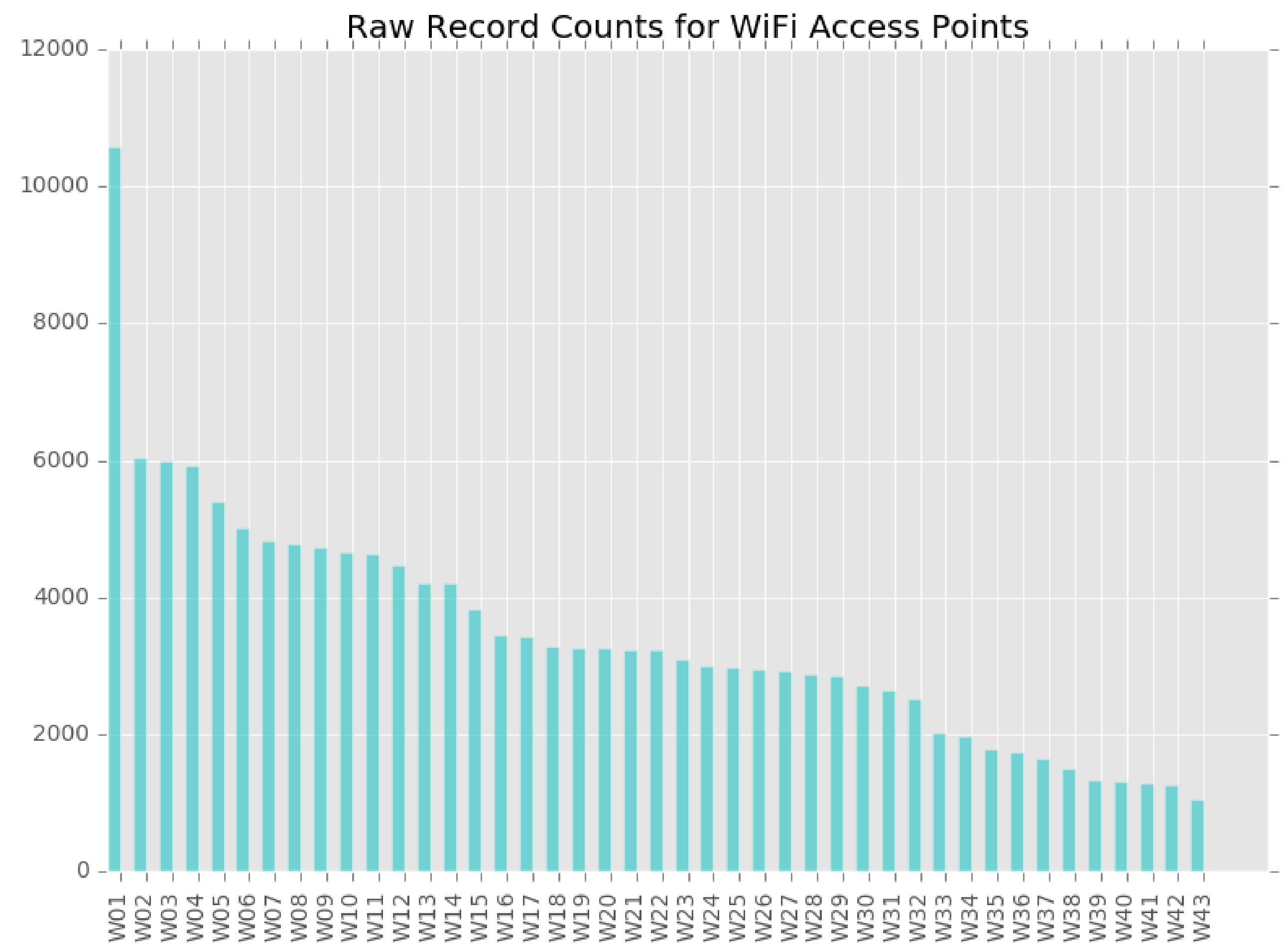


Figure 5.3

Forty access points passed the intitial filter which means their are forty three features for access points.

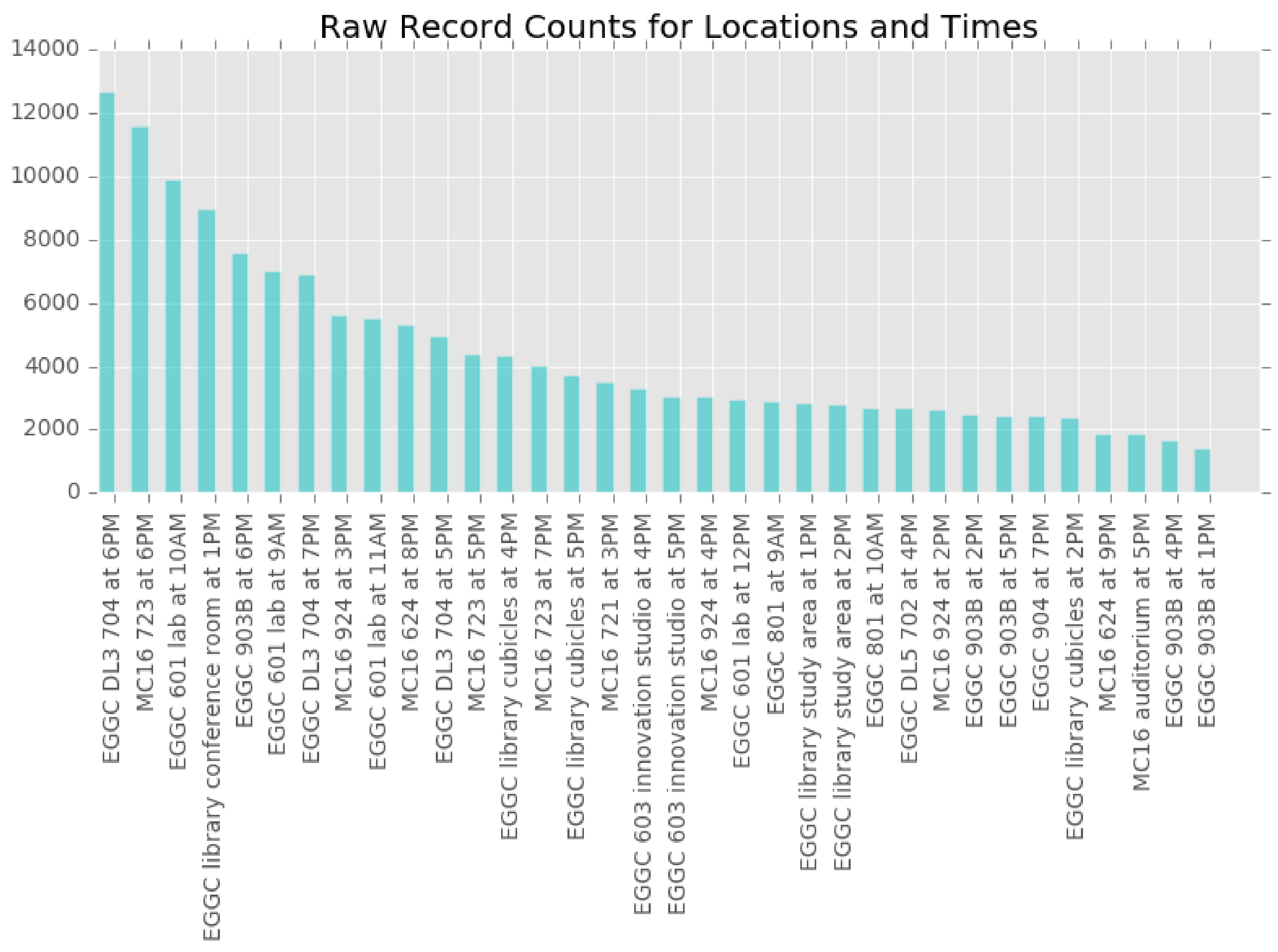

Raw Record Counts for Locations and Times
14000
12000
10000
8000
6000
4000
2000
0
EGGC DL3 704 at 6PM
MC16 723 at 6PM
EGGC 601 lab at 10AM
EGGC library conference room at 1PM
EGGC 903B at 6PM
EGGC 601 lab at 9AM
EGGC DL3 704 at 7PM
MC16 924 at 3PM
EGGC 601 lab at 11AM
MC16 624 at 8PM
EGGC DL3 704 at 5PM
MC16 723 at 5PM
EGGC library cubicles at 4PM
MC16 723 at 7PM
EGGC library cubicles at 5PM
MC16 721 at 3PM
EGGC 603 innovation studio at 4PM
EGGC 603 innovation studio at 5PM
MC16 924 at 4PM
EGGC 601 lab at 12PM
EGGC 801 at 9AM
EGGC library study area at 1PM
EGGC library study area at 2PM
EGGC 801 at 10AM
EGGC DL5 702 at 4PM
MC16 924 at 2PM
EGGC 903B at 2PM
EGGC 903B at 5PM
EGGC 904 at 7PM
EGGC library cubicles at 2PM
MC16 624 at 9PM
MC16 auditorium at 5PM
EGGC 903B at 4PM
EGGC 903B at 1PM

Figure 5.4

Signal Strength

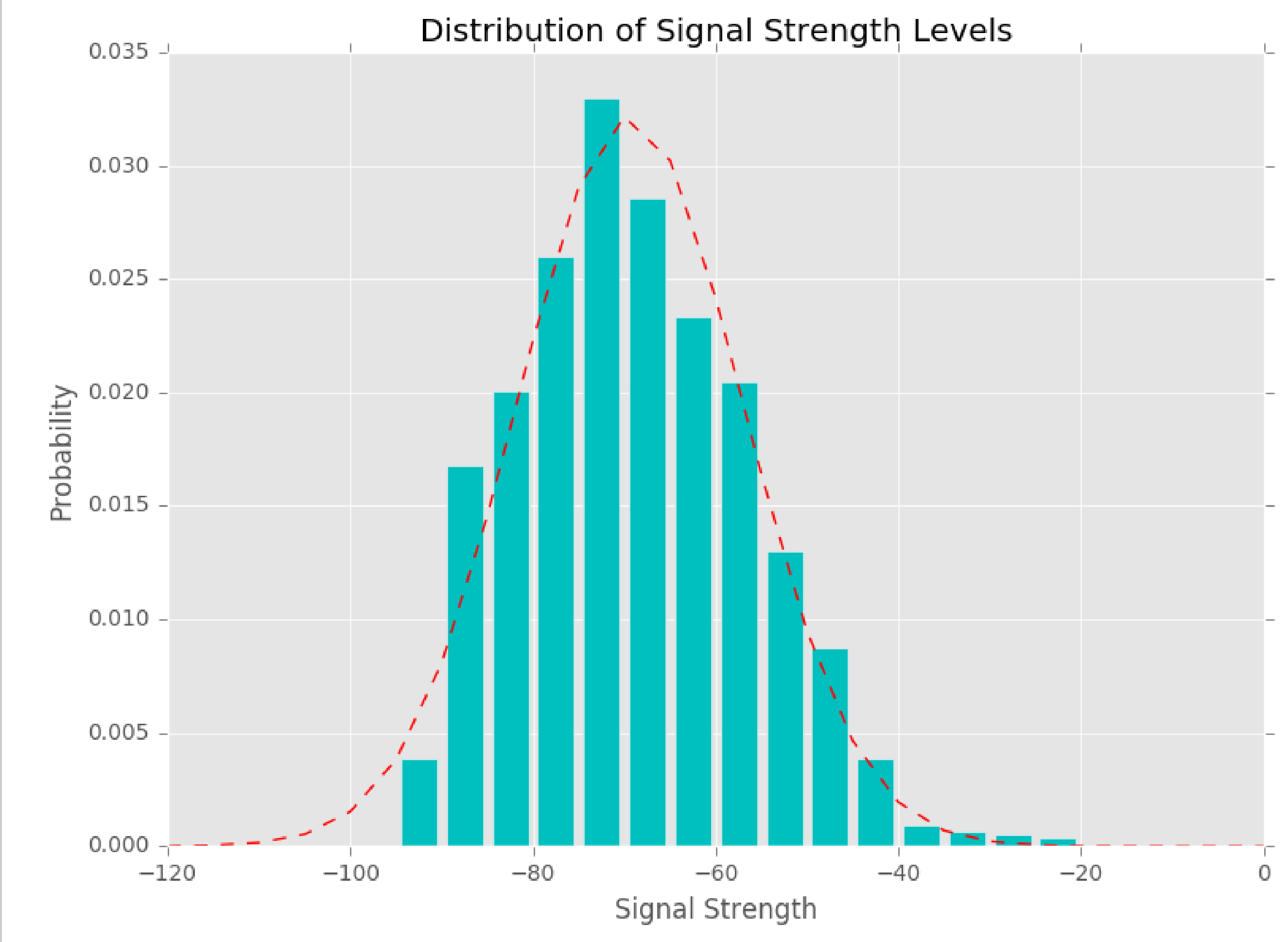


Figure 6

The best and worst signal strength's recorded in my dataset were -20 dB and -95 dB respectively. The signal strength is normally distributed.

Fingerprints

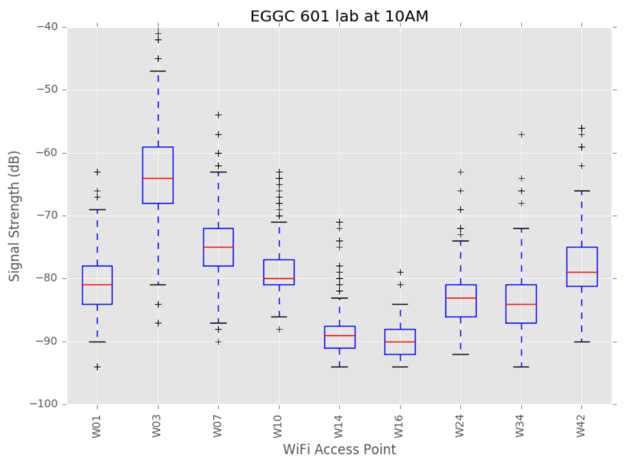


Figure 7.1

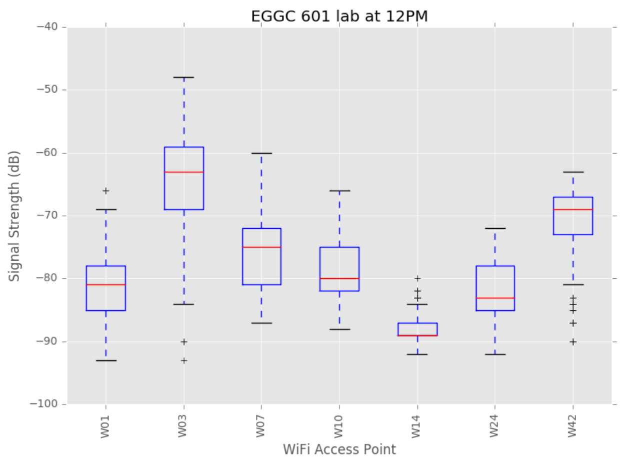


Figure 7.2

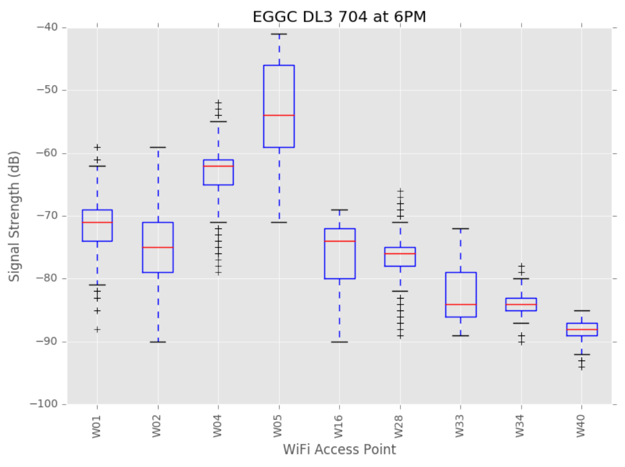


Figure 7.3

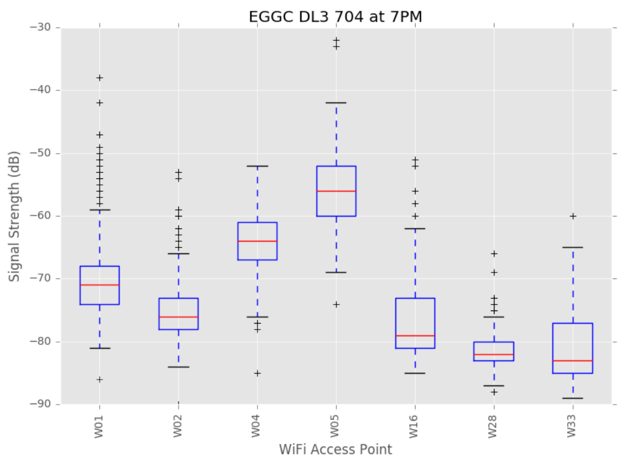


Figure 7.4

Prediction Accuracy

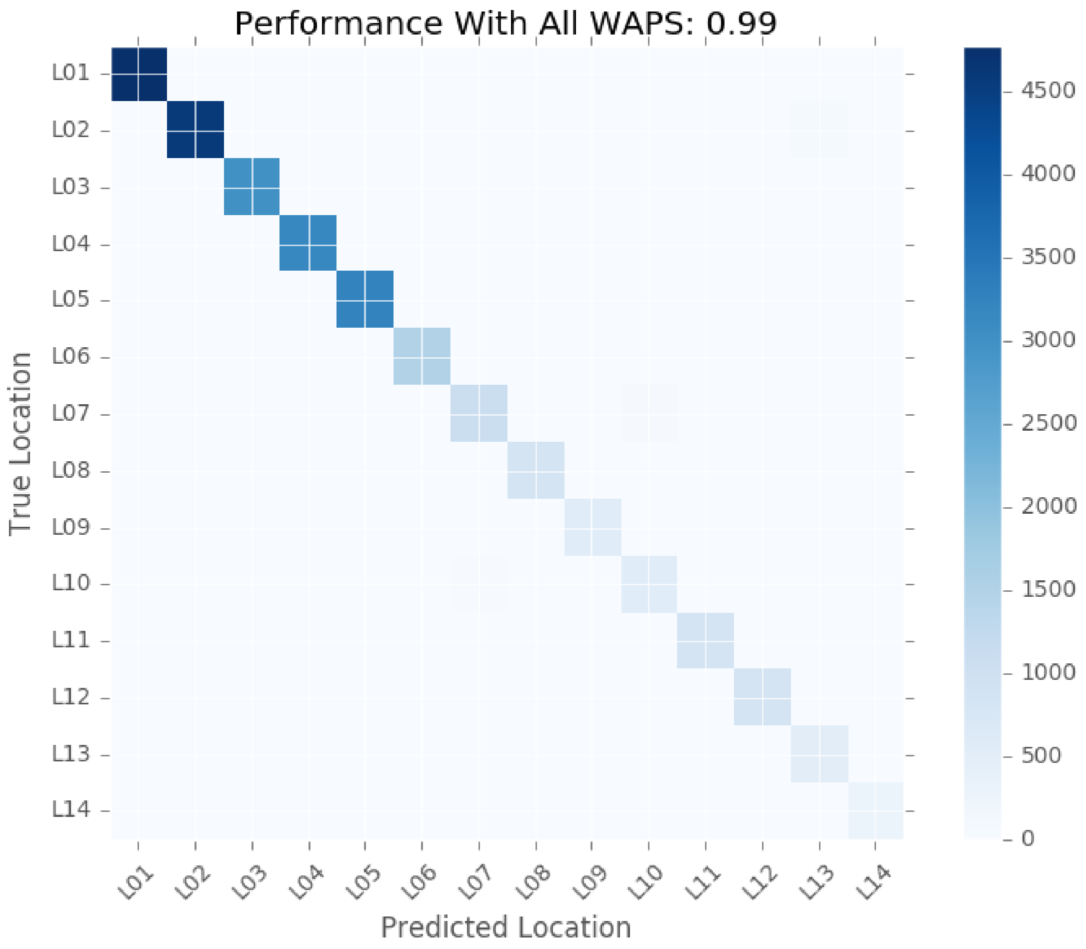


Figure 8.1

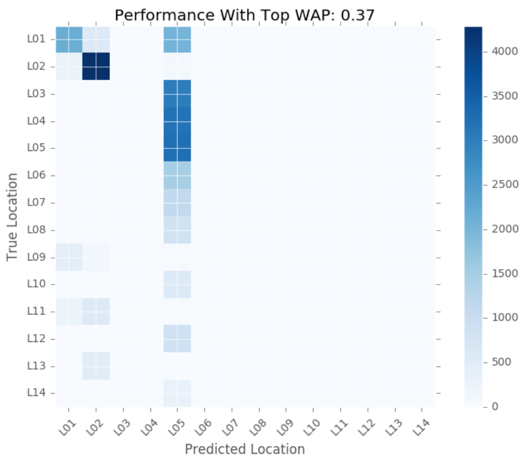


Figure 8.2

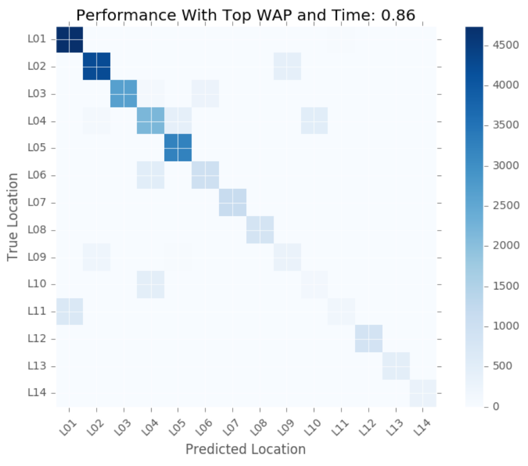


Figure 8.3

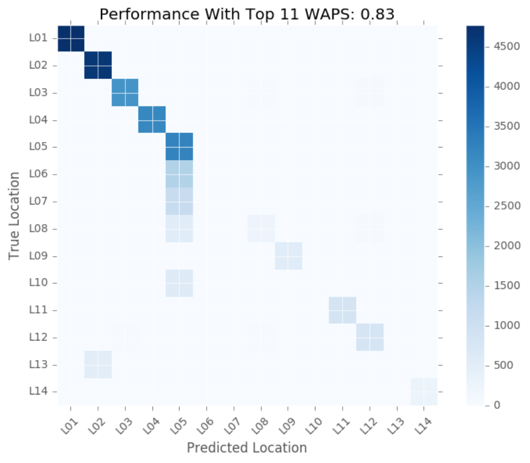


Figure 8.4

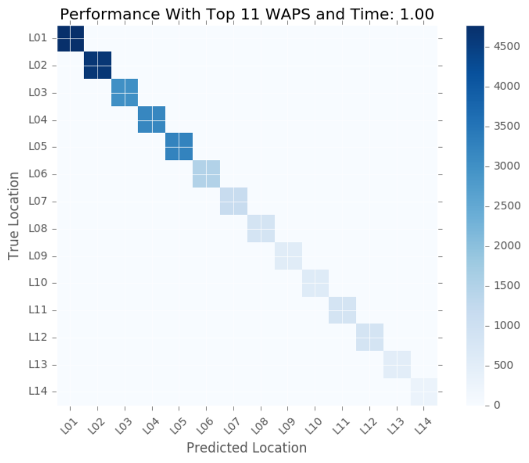


Figure 8.5

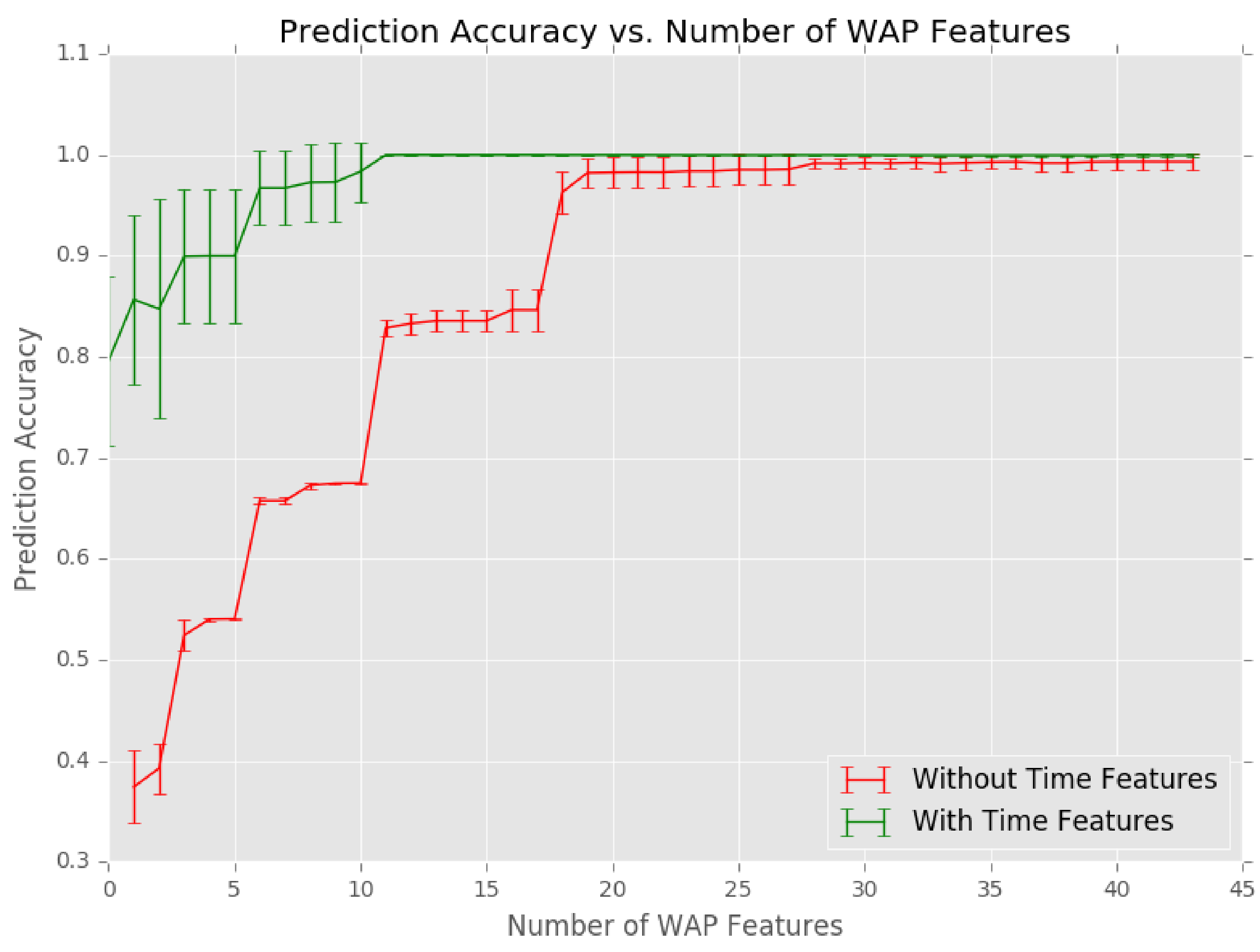


Figure 9

Prediction accuracy is significantly improved by including time features in the classification algorithm. With my dataset 100% accuracy can be achieved by using time and 11 access points. Without time the 28 most common access points are required to get close to the same accuracy. However as shown in Figure 5.1 and Figure 5.2, my dataset is highly unbalanced with regard to both time and location. That might be the reason we see such high improvement by using time as a feature.

Future Projects

The app should be modifed to record the phone's model number. This is important since the signal reading is dependant on the antenna and each modle of phone may have it's own.

In order to easily balance the dataset, raspberry pi's should be placed in each room and be programed to collect training data. Then the classifiers should be tested with samples from the app.

Learning Outcomes

Server	**Android**	**Machine Learning**
Database	SQLite	Training Classifiers
REST API in Flask	Broadcast Receivers	Cross Validation
Apache with HTTPS	Background Services	Confusion Matrices
Sending Email	Making HTTPRequests	Matplotlib
	Notifications	