

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

---

SCUOLA DI SCIENZE  
Corso di Laurea in Informatica

# Una piattaforma per la localizzazione indoor tramite crowdsensing

**Relatore:**  
Chiar.mo Prof.  
Luciano Bononi

**Presentata da:**  
Erik Minarini

**Correlatore:**  
Dott.  
Luca Bedogni

III Sessione  
Anno Accademico 2014/2015



# Abstract

Questo elaborato è atto ad analizzare e studiare il comportamento di tre algoritmi di base per quanto riguarda la localizzazione indoor e in particolare sulla tecnica del fingerprint. Maggior concentrazione andrà sull'eterogeneità dei dispositivi e su come essa possa influenzare gli algoritmi e la loro accuratezza nel risultato. Si mostreranno nozioni sull'attuale stato dell'arte e si seguirà la progettazione e lo sviluppo del sistema realizzato composto da un'applicazione Android e un web service. In seguito verranno mostrati i test effettuati e le considerazioni fatte su essi.



# Indice

<b>Abstract</b>	<b>i</b>
<b>1 Introduzione</b>	<b>1</b>
<b>2 Concetti di Base e Stato dell'Arte</b>	<b>5</b>
2.1 Wi-Fi . . . . .	5
2.2 Fingerprinting . . . . .	6
2.3 Indoor Localization . . . . .	6
2.4 Algoritmi e Stato dell'arte . . . . .	7
2.4.1 Metodo Probabilistico . . . . .	7
2.4.2 k-nearest-neighbor (kNN) . . . . .	8
2.4.3 Reti Neurali . . . . .	8
2.4.4 Smallest M-vertex Polygon (SMP) . . . . .	9
2.5 Eterogeneità . . . . .	9
<b>3 Progettazione e Implementazione</b>	<b>11</b>
3.1 Applicazione Android . . . . .	11
3.1.1 Scan . . . . .	11
3.1.2 DBList . . . . .	13
3.1.3 Recognize . . . . .	13
3.2 Web Service . . . . .	13
3.2.1 DBFunctions . . . . .	14

---

<b>4</b>	<b>Test e Sperimentazioni</b>	<b>17</b>
4.1	Fase Preliminare . . . . .	17
4.2	Sperimentazione . . . . .	18
<b>5</b>	<b>Analisi dei Dati</b>	<b>23</b>
5.1	Lettura dei grafici . . . . .	23
5.2	Majority Rule . . . . .	24
5.3	Weighted Majority Rule . . . . .	24
5.4	Least Average Error . . . . .	25
5.5	Fusion Rule . . . . .	25
5.6	Etereogenità . . . . .	26
5.7	Considerazioni finali . . . . .	26
	<b>Conclusioni</b>	<b>31</b>
<b>A</b>	<b>Codice Ausiliario</b>	<b>33</b>
A.1	Script Python . . . . .	33
A.2	Script Bash . . . . .	33
A.3	Strumenti grafici . . . . .	34
	<b>Bibliografia</b>	<b>35</b>

# Elenco delle figure

1.1	Esempio dell'utilizzo della potenza WI-Fi per l'individuazione di un piano.[1] . . . . .	2
1.2	Esempio di heatmap che mostra la forza del segnale ricevuta. [1] . . . . .	3
2.1	Normalizzazione della potenza di segnale tramite Hyperbolic Location Fingerprinting. [8] . . . . .	10
2.2	Errori di tre diversi dispositivi. [8] . . . . .	10
3.1	L'Activity Scan che permette di effettuare le rilevazioni attraverso il dispositivo . . . . .	15
3.2	L'Activity DBList che permette di cancellare i dati o procedere con l'upload sul server. . . . .	16
4.1	Mappa del dipartimento dove sono state eseguite le rilevazioni.	18
4.2	1 - 10 Heatmap, realizzate dopo la prima raccolta dati, rappresentano la potenza di segnale rilevata per ogni BSSID percepito nel dipartimento al momento della raccolta. . . . .	20
4.3	11 - 18 Heatmap, realizzate dopo la prima raccolta dati, rappresentano la potenza di segnale rilevata per ogni BSSID percepito nel dipartimento al momento della raccolta. . . . .	21
5.1	4 Grafici che rappresentano la percentuale di individuare l'esatto punto, nell'ordine OnePlus One,LG Nexus 4, Samsung S2, Samsung S6. . . . .	27

5.2	4 Grafici che rappresentano la percentuale di individuare la stanza, nell'ordine OnePlus One, LG Nexus 4, Samsung S2, Samsung S6. . . . .	28
5.3	4 Grafici che rappresentano l'errore in metri degli algoritmi, nell'ordine OnePlus One, LG Nexus 4, Samsung S2, Samsung S6. . . . .	29
A.1	Screen di parte dello script python. . . . .	34

# Capitolo 1

## Introduzione

Grazie al sempre più comune utilizzo di dispositivi portatili come Laptop, SmartPhone e Tablet, la creazione di sistemi in grado di localizzare la posizione di un utente all'interno di un edificio è diventata possibile. La localizzazione in ambienti chiusi, indoor, differisce dalla localizzazione outdoor poiché essa viene tipicamente effettuata utilizzando il Global Positioning System (GPS) che ha un'accuratezza di circa 10 metri, il GPS però risulta poco efficace all'interno di un edificio, poiché il dispositivo non ha una linea di vista con almeno tre satelliti. In letteratura gran parte della ricerca fatta sulla localizzazione interna riguarda lo standard IEEE 802.11, il Wi-Fi e la tecnica del fingerprint. Focalizzando l'attenzione nell'ambito delle applicazioni indoor, un possibile esempio potrebbe riguardare la sicurezza personale: immaginiamo un grande grattacielo di una città, in caso di emergenza medica se il dispositivo dal quale si effettua la chiamata fosse in grado di comunicare immediatamente, con una buona precisione, la sua posizione potrebbe essere molto d'aiuto per i soccorsi dirigersi subito sul luogo, questo potrebbe, in caso di edifici piuttosto grandi, accelerare molto i tempi. Un altro esempio, più vicino al nostro contesto, potrebbe essere quello di tracciare velocemente un professore di cui si ha bisogno all'interno dell'Università, o una qualsiasi altra figura abbia la necessità di essere rintracciabile in poco tempo. Il contributo di questa tesi consiste nell'analisi dello stato dell'arte dei siste-

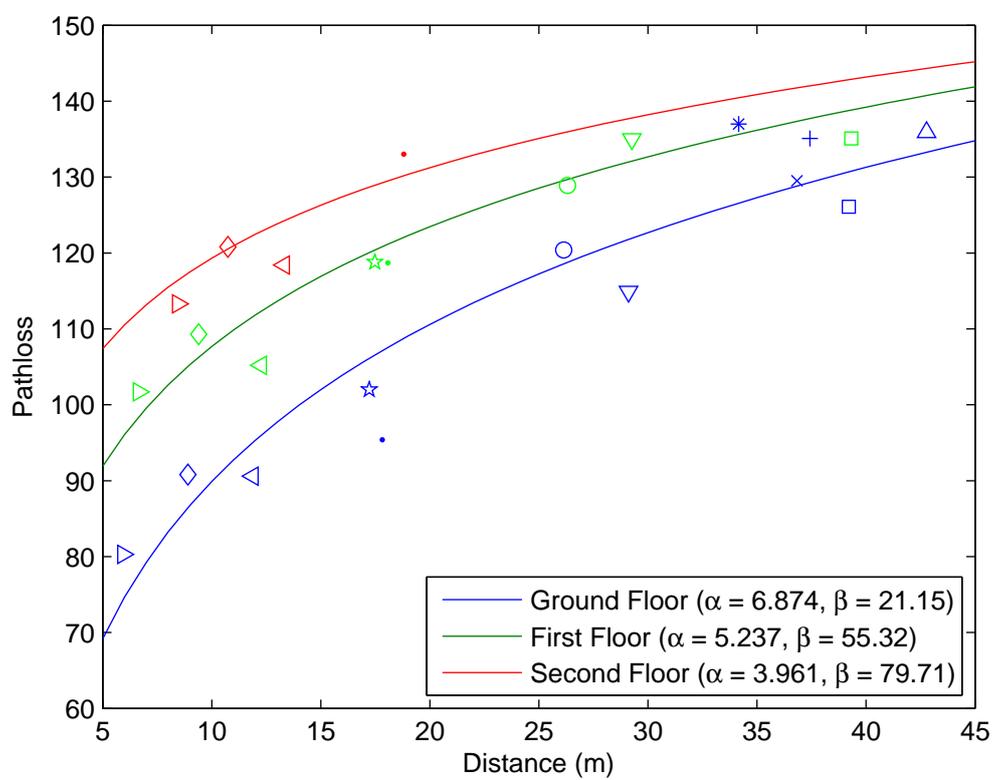


Figura 1.1: Esempio dell'utilizzo della potenza WI-Fi per l'individuazione di un piano.[1]

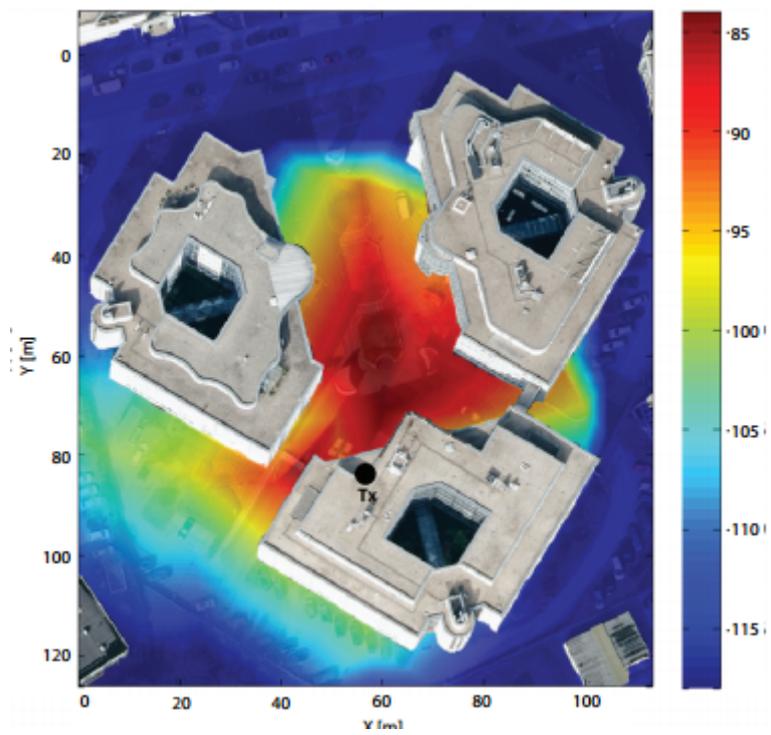


Figura 1.2: Esempio di heatmap che mostra la forza del segnale ricevuta. [1]

mi di localizzazione indoor, e nello sviluppo di un sistema di tracciamento, che prevede un'applicazione Android e un web service, basato sulla tecnica del fingerprinting ed utilizzando il crowdsensing, ovvero raccogliendo dati da dispositivi diversi e non da uno soltanto.

La trattazione sarà così articolata:

- Nel capitolo 1 forniremo una panoramica sullo stato dell'arte e introdurremo i concetti di base che verranno utilizzati nel corso dell'elaborato.
- Nel capitolo 2 tratteremo la progettazione e l'implementazione del sistema.
- Il capitolo 3 tratterà i campioni raccolti e dei test effettuati.
- A chiudere il presente lavoro, il capitolo 4 raccoglierà le conclusioni di tutto il lavoro effettuato.

# Capitolo 2

## Concetti di Base e Stato dell'Arte

### 2.1 Wi-Fi

Il Wi-Fi è una tecnologia in grado di connettere due o più terminali in modo da consentire lo scambio di dati fra di essi senza l'ausilio di fili, utilizzando invece onde radio, come specificato dallo standard 802.11. Questo sistema di comunicazione permette, oltre alla condivisione di file, anche una connessione ad Internet vera e propria, collegando un dispositivo con un access point. Per identificare una rete Wi-Fi si fa riferimento al suo SSID (Service Set Identifier), ovvero un nome solitamente scelto dall'utente che può essere d'aiuto nel riconoscere la propria WLAN. Ognuna di queste reti è composta da uno o più access point, ovvero punti di accesso alla rete in grado di propagare il segnale, e uno o più client connessi a questi punti. Inoltre, ogni access point è identificato da un BSSID univoco e corrispondente, a livello infrastruttura, all'indirizzo MAC dello stesso. Tale BSSID, solitamente, propaga il segnale in un'area compresa fra i 50 e i 100 metri a seconda della configurazione architettonica della zona in esame. Nel 1999, lo IEEE (Institute of Electrical and Electronics Engineers) rilasciò il protocollo 802.11 che definì tutti i principi strutturali di questa tecnologia. Le certificazioni dello IEEE han-

no permesso di stabilire, di volta in volta, gli standard tecnologici ai quali i produttori potevano riferirsi per realizzare i propri dispositivi Wi-Fi. L'intensità del segnale Wi-Fi è espressa in una unità di misura indicata con dBm: il dBm misura la potenza del segnale radio ed è espresso in decibel sul livello di riferimento di un milliwatt. I decibel (dB) sono numeri puri ottenuti effettuando il logaritmo del rapporto fra due grandezze omogenee; per tali ragioni risulta chiaro come a 0dBm corrisponda il segnale ottimo che va peggiorando allontanandoci, su scala negativa, dallo 0. Si osserva che circa ogni 3dBm di aumento di segnale sia necessaria una potenza doppia.

## 2.2 Fingerprinting

Il fingerprinting è una tecnica di pattern matching e si realizza in due fasi: Nella prima fase vengono raccolti i campioni, più campioni per ogni punto di interesse (POI). Un campione è composto da received signal strength (RSS) di diversi access point, ovvero una serie di tuple formate da BSSID e potenza di segnale. Questa serie è detta fingerprint (impronta digitale). Il fingerprint viene salvato in un database dove ogni entry è del tipo:

$$F_i = (x, y, RSS_1, RSS_2, \dots, RSS_n), i = 1, \dots, L \quad (2.1)$$

dove L è il numero della rilevazione effettuata e N il numero di access point "sentiti" in quel punto. Nella seconda fase il terminale raccoglie un campione in un punto e lo invia al server il quale, in base agli algoritmi di riconoscimento, risponderà con una posizione [2] [3]. Gli algoritmi utilizzati per il matching dei fingerprint possono essere di diversi tipi e verranno trattati in maniera più approfondita nel prossimo capitolo .

## 2.3 Indoor Localization

La localizzazione indoor nasce per essere applicata all'interno di edifici, campus universitari, ospedali, uffici; è basata su tecnologie radio ed infrarossi, con un raggio di comunicazione alquanto limitato. Questo sistema

può essere realizzato sia mediante infrastrutture dedicate, che facendo uso di quelle preesistenti come nel caso delle reti WLAN. Nel primo caso i terminali da localizzare saranno equipaggiati con badge che lavorano con segnali infrarossi, sonori, o con la tecnologia Radio Frequency Identification (RFID), i quali ricevono e comunicano con la stazione base più vicina. Se invece il posizionamento è realizzato usando delle reti WLAN, i terminali dovranno essere provvisti di una scheda di rete che comunicherà con gli AP. I vantaggi della localizzazione indoor sono il basso consumo dei dispositivi coinvolti, e la relativamente elevata precisione dovuta al corto raggio delle tecnologie usate.

## 2.4 Algoritmi e Stato dell'arte

Il maggiore problema dovuto alle tecniche basate sul metodo fingerprinting è che l'intensità del segnale ricevuto può essere affetta, in un contesto di propagazione indoor, da diffrazione, riflessione e scattering. Ci sono molti algoritmi di localizzazione basati sul fingerprinting che utilizzano tecniche di pattern recognition, alcuni di essi sono: metodo probabilistico, k-nearest-neighbor (kNN), reti neurali e Smallest M-vertex Polygon (SMP) [4].

### 2.4.1 Metodo Probabilistico

Questo metodo considera il posizionamento come un problema di classificazione. Supponiamo che l'elemento alfa si possa trovare in  $n$  possibili posizioni  $x_1, \dots, x_n$  e che  $F$  sia il vettore contenente le misure (RSS)[4]. Si indica con  $P(p_i|F)$  la probabilità che alfa sia nella posizione  $p_i$  dato il vettore dei segnali ricevuti  $F$  e con  $P(x_i)$  la probabilità che alfa sia nella posizione  $p_i$ . Basandoci sulla probabilità a posteriori otteniamo questa regola di decisione: Si seleziona  $x_i$  se

$$P(x_i|F) > P(x_j|F), \forall i, j = 1, \dots, n, i \neq j. \quad (2.2)$$

### 2.4.2 k-nearest-neighbor (kNN)

Vengono cercate le  $k$  corrispondenze più vicine fra il fingerprint percepito e i dati raccolti, presenti nel database. Si utilizza solitamente il metodo dei minimi quadrati o la distanza euclidea. Successivamente si effettua il calcolo della media delle  $k$  posizioni, ottenendo una stima della posizione [5] [6].

### 2.4.3 Reti Neurali

Grazie ad un training set formato da valori d'ingresso (il valore RSS) e dalle corrispettive uscite (le coordinate geografiche), la rete neurale è in grado di imparare a riconoscere la relazione che li lega. Questo tipo di apprendimento è detto supervisionato (supervised learning). Dopo il primo apprendimento la rete viene sottoposta ad un opportuno algoritmo (backpropagation), il quale usa i dati con l'obiettivo di minimizzare l'errore della previsione dovuto al training set utilizzato, modificando i pesi ed altri parametri della rete stessa. La rete impara a riconoscere la relazione che lega l'input con l'output, ed è quindi in grado di fare previsioni avendo un dato in ingresso; l'obiettivo finale dell'apprendimento supervisionato è quindi la previsione corretta del valore d'uscita per ogni elemento valido in ingresso, basandosi soltanto su alcuni esempi di corrispondenza. Per i sistemi di localizzazione basati su reti neurali viene generalmente utilizzata una rete Multilayer Perceptron (MLP) con un unico strato di neuroni nascosti. Il vettore  $X$ , contenente i valori di RSS, passato in ingresso alla rete viene moltiplicato per il vettore dei pesi  $W1$  e poi sommato con il bias dello strato di input. Assumendo  $N$  la dimensione dei vettori  $X$  e  $W1$ , si ha:

$$y_i = g_i\left(\sum_{j=1}^n w_{i,j}x_j + \vartheta_i\right) \quad (2.3)$$

dove  $g$  è la funzione di attivazione (del singolo neurone),  $w$  e  $x$  sono rispettivamente il peso della sinapsi  $n_j - n_i$  e l'ingresso  $i$ -esimo,  $\vartheta$  è lo scostamento (bias) assegnato ad ogni strato durante la fase di apprendimento. Il risultato  $y_1$  è passato alla funzione di trasferimento dello strato di neuroni nascosto.

L'output viene moltiplicato per il vettore dei pesi  $w_2$  per poi essere sommato con il bias dello strato nascosto. Il risultato finale del sistema è un vettore a due o tre elementi, contenete della posizione cercata in due o tre dimensioni [4].

#### 2.4.4 Smallest M-vertex Polygon (SMP)

Consideriamo M nodi fissati i quali vengono interrogati da un nodo incognito. Considerando il fingerprint di ogni nodo si costruisce un poligono a M-vertici. Ripetendo questa procedura si otterranno più poligoni, in base a quanti nodi sono vicini al nodo incognito, da cui si estrae il minore. Facendo la media fra le coordinate di questo poligono, si stima la posizione del nodo incognito [5].

## 2.5 Eterogeneità

L'eterogeneità delle schede WLAN all'interno dei dispositivi è un grosso problema per chi fa ricerca riguardo alla localizzazione indoor, in letteratura si notano anche differenze attorno a 25 - 30 dBm nei fingerprints di dispositivi differenti [7]. Le differenze possono essere attribuite sia all'hardware sia al software e soprattutto alla mancanza di uno standard. In letteratura una soluzione che gestisce l'eterogeneità fra dispositivi è l'uso di Hyperbolic Location Fingerprinting (HLF), ovvero i fingerprints vengono salvati come rapporto fra gli acces point invece che una assoluta potenza di segnale [8].

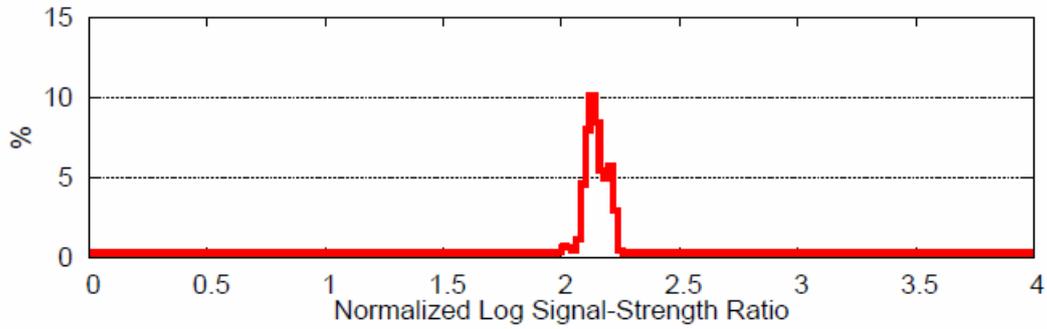


Figura 2.1: Normalizzazione della potenza di segnale tramite Hyperbolic Location Fingerprinting. [8]

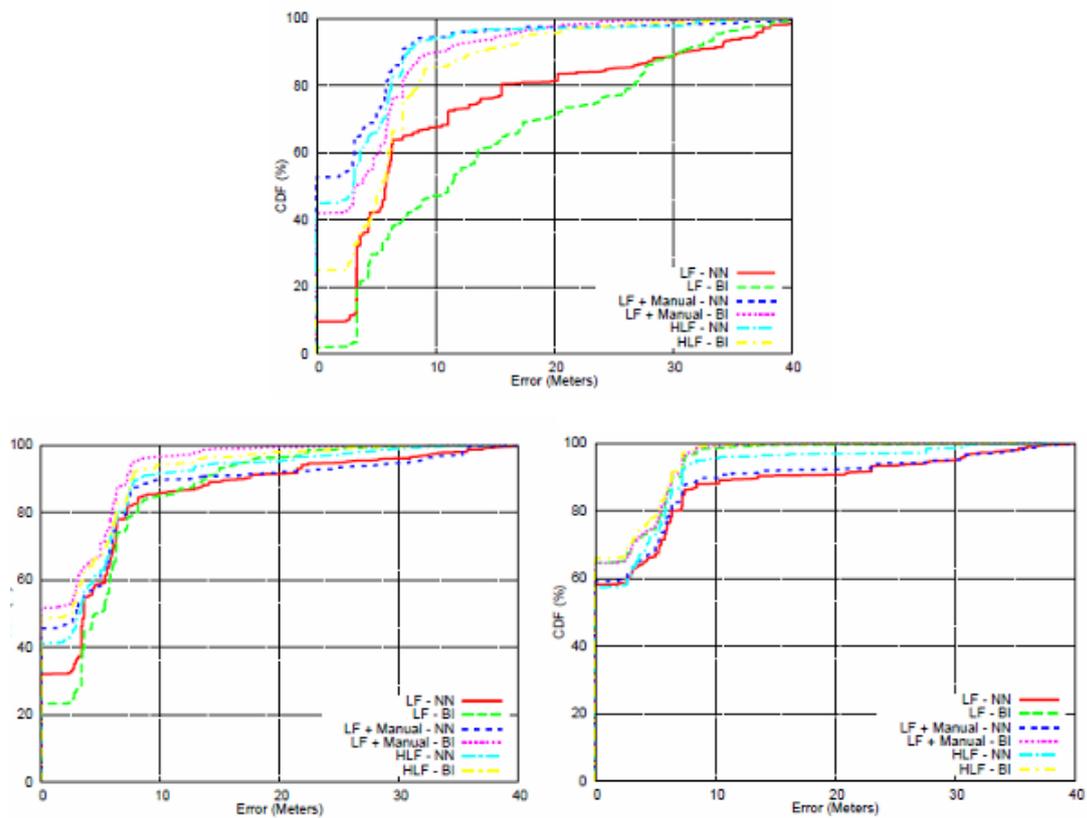


Figura 2.2: Errori di tre diversi dispositivi. [8]

# Capitolo 3

## Progettazione e Implementazione

Il progetto, come questo capitolo del resto, si divide in 2 parti principali:

- L'applicazione Android,
- Il web service.

### 3.1 Applicazione Android

L'applicazione Android svolge due principali compiti: prendere le rilevazioni in modo da tracciare la zona e farsi tracciare dal web service, comunicando all'utente dove si trova. All'avvio dell'applicazione ci troveremo davanti due opzioni, Scan e Recognize, che avvieranno le rispettive activity e tramite un tasto opzioni potremmo impostare l'indirizzo del server.

#### 3.1.1 Scan

Scan è l'activity che si occupa del tracciamento dell'area, la sua interfaccia si compone di: una TextArea (Tag) nella quale potremmo inserire un tag, che verrà associato alle coordinate di latitudine e longitudine ed avrà il duplice compito di aiutare nella lettura delle informazioni e fornirci un grado di

accuratezza intermedio; una `TextArea` (`Timer`), nella quale inseriremo ogni quanti secondi le rilevazioni dovranno partire; un `Button` (`Scan`) il quale effettuerà una singola rilevazione; un `Button` (`Play`) che farà partire/fermare un timer che effettuerà una rilevazione ogni  $n$  secondi in base a quelli impostati in `Timer`; una `View` (`Map`) che mostrerà una mappa di `OpenStreetMap`, nella quale, con un `longTap`, è possibile posizionare un marker che corrisponderà alla nostra posizione nell'edificio; quattro `Button` (`Up`, `Down`, `Right`, `Left`) con i quali sarà possibile spostarsi all'interno della mappa più agevolmente e, infine, un tasto menù, in alto a destra, che porterà all'activity `DBList` la quale mostra tutto il database interno al dispositivo e permette di effettuare la cancellazione totale dello stesso o l'upload dei dati sul server. L'activity `Scan` funziona solo con il Wi-Fi abilitato, poiché sarà proprio quest'antenna a comunicarci che BSSID percepiamo, per cui se è disabilitata verrà abilitata. All'inizio `Map` troverà la posizione del dispositivo e la utilizzerà come centro, ovviamente essendo all'interno di un edificio la posizione percepita sarà molto imprecisa, motivo per cui sarà l'utente stesso a posizionarsi correttamente sulla mappa (e anche motivo del trattato stesso). Il metodo `singleScan`, richiamato una sola volta dal tasto `Scan` e ogni  $n$  secondi dal timer, si occuperà di raccogliere tutti i dati e salvarli in un database interno al dispositivo. Per prima cosa si ottengono le informazioni sul tipo di dispositivo, quindi modello e produttore, poi viene effettuato un timestamp. Dopo vengono raccolti tutti i BSSID percepiti con la loro rispettiva potenza, si ottengono dalla mappa latitudine e longitudine, infine si prende il tag dalla `TextArea`. Otterremo così una tupla di questo tipo: (produttore, modello, lat, long, datetime, tag, RSS1, ..., RSSn) non ci resta che salvarla nel database, per farlo abbiamo un metodo `save` della nostra classe `DBManager`. I tasti `Up`, `Down`, `Right` e `Left` rimuovono il marker dalla mappa (se presente) e spostano longitudine o latitudine di più o meno ( $0.00001 * 1E6$ ) poi creano un nuovo marker con le coordinate cambiate e lo reinseriscono nella mappa.

### 3.1.2 DBList

L'activity DBList si presenta semplicemente con quattro elementi: una TextView che indica il percorso nel quale è situato il database interno, un Button (Delete) che permette di svuotare completamente il database, un Button (Upload) il quale consente di inviare tutto il database al server ed una ListView che mostra tutte le entry presenti all'interno del database. Il tasto Delete fa partire un nuovo Thread il quale utilizza il metodo delete della classe DBManager spostando il cursore sull'entry successiva fino a che il database non è vuoto ed infine manda un messaggio all'handler che aggiorna l'adapter della ListView. Il tasto Upload fa partire un nuovo Thread il quale, per ogni entry nel database manda i dati al webservice, utilizzando un AsyncTask e se il POST è andato a buon fine cancella la riga dal database e manda un messaggio all'handler per aggiornare la UI, poi continua il ciclo fino a che il database non è vuoto.

### 3.1.3 Recognize

Recognize è l'activity che si occupa di comunicare all'utente dove il sistema lo posiziona all'interno dell'edificio. Graficamente si presenta con un Button (Recognize) che si occuperà di inviare i dati e una View (Map) che, come per l'activity Scan rappresenta una mappa di OpenStreetMap, sulla quale verrà mostrata la posizione. Il tasto Recognize raccoglie i dati in modo analogo al metodo singleScan, con la differenza che non verranno salvati sul database interno al dispositivo ma verranno direttamente inviati al webservice e, dopo aver ricevuto risposta, posizionerà uno o più marker, uno per ogni algoritmo che abbia dato un risultato, all'interno della mappa.

## 3.2 Web Service

Il web service è scritto in PHP ed è il fulcro del sistema per quanto riguarda i calcoli, gli algoritmi e le analisi, è composto da varie parti, avremo:

- un file di configurazione nel quale vengono semplicemente definite le costanti utili per l'accesso al database, come nome utente, password e nome del database.
- La classe DBConnect che va a definire l'oggetto che utilizzeremo per connetterci al database, questa classe ha solo un metodo connect(), che ritorna un oggetto connesso al db.
- La classe DBFunctions è sicuramente la parte più importante del web service, essa contiene alcuni metodi utili alla gestione ed all'analisi dei dati e l'implementazione dei tre algoritmi utilizzati.

### 3.2.1 DBFunctions

Al suo interno sono presenti: due strutture dati contenenti l'elenco dei BSSID rilevati e l'elenco dei TAG utilizzati per annotare i punti (POI) nei quali sono stata effettuata le rilevazioni. Un metodo per aggiungere un'entry al database, che inserirà nella tabella generale tutte le rilevazioni e separerà ogni coppia BSSID - Power per inserirle nella tabella Fingerprints. Il metodo maxID che ritornerà l'ID massimo presente nella tabella generale. Il metodo randomEntry che si occuperà, dato un minimo, un massimo e un device, di generare un fingerprint casuale appartenente a quel particolare dispositivo. I metodi ap e deviceType che ritorneranno la lista rispettivamente degli access point e dei dispositivi differenti presenti nel database. I tre metodi per popolare le tabelle con la media, il minimo ed il massimo delle potenze per ogni punto (POI) ed ogni tipologia di device. Ed infine i tre metodi che implementano gli algoritmi utilizzati per calcolare la posizione del dispositivo in base al fingerprint inviato. I tre algoritmi implementati sono: MajorityRule: viene selezionata nel database la riga che ha più corrispondenze, BSSID e sua relativa potenza, con il fingerprint analizzato. Weighted MajorityRule: si cercano all'interno del database le righe con più corrispondenze, dando però più peso ai BSSID con potenze maggiori (più vicine allo 0) e meno a quelle più lontane, poiché rilevare una variazione di qualche dBm su una potenza

bassa è meno influente che rilevarla in una potenza alta, in altre parole, più la potenza è vicina allo zero più qualche dBm di differenza evidenzia uno spostamento concreto e non un semplice disturbo. LeastAvgError: per ogni entry del database che contiene almeno m BSSID in comune con il fingerprint, con m fissato, viene calcolato l'errore medio sugli access point condivisi e si ritorna la posizione con errore medio più piccolo.

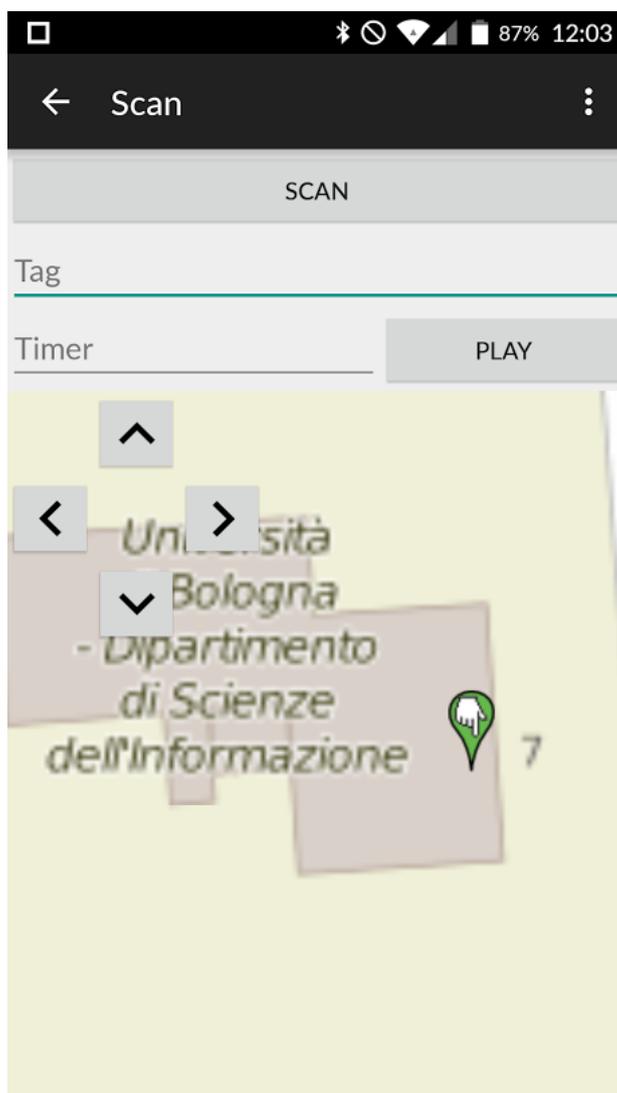


Figura 3.1: L'Activity Scan che permette di effettuare le rilevazioni attraverso il dispositivo

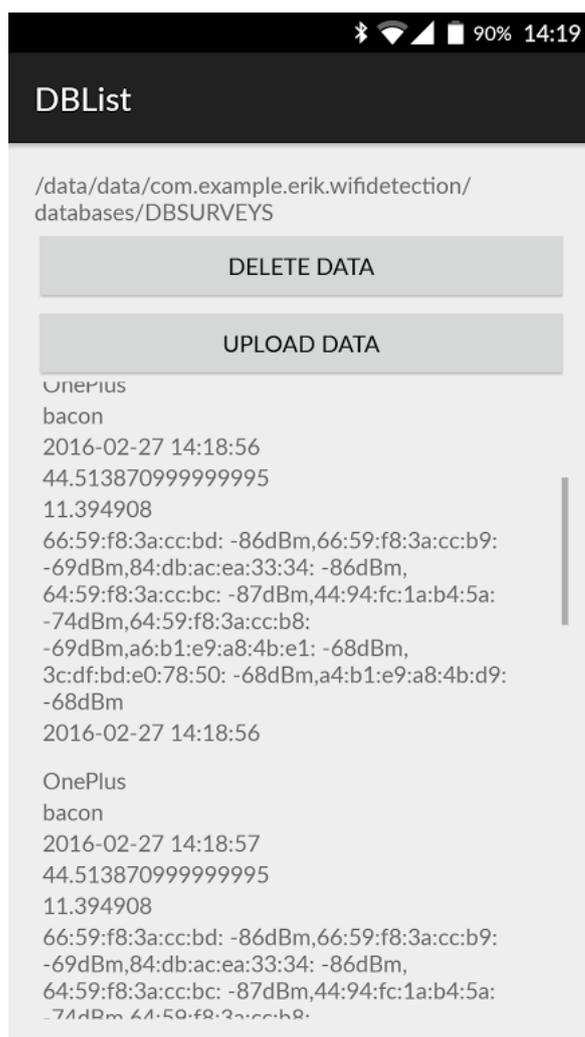


Figura 3.2: L'Activity DBList che permette di cancellare i dati o procedere con l'upload sul server.

# Capitolo 4

## Test e Sperimentazioni

### 4.1 Fase Preliminare

Per prima cosa si sono effettuate alcune rilevazioni, procedendo come segue: con una cartina del dipartimento si sono evidenziati dei punti di interesse (POI), vari a seconda dalla grandezza della stanza. Per ognuno di essi, con un dispositivo si sono "catturati" dieci fingerprints a distanza di due secondi l'uno dall'altro utilizzando l'applicazione Android da me sviluppata. Successivamente si sono effettuate altre rilevazioni con le stesse modalità delle precedenti ma variando il numero per ogni punto (cinque anziché dieci), e la quantità dei dispositivi utilizzati (quattro anziché uno). Utilizzando Google Maps si sono ottenute le misure del dipartimento, in modo da poterlo disegnare in un piano cartesiano e facendo così corrispondere ogni punto ad una coppia di coordinate  $(x,y)$ . In una primissima analisi si sono visualizzati tutti i BSSID percepiti nel dipartimento, e per ognuno di essi è stata realizzata una heatmap (Figg. 4.2, 4.3), mostrando il livello di ricezione per ogni punto tracciato durante le rilevazioni. Si sono ovviamente create le heatmap solo dei BSSID con un segnale percepito sufficiente.

Sono state realizzate tre coppie di tabelle che sono fondamentali per i test, ovvero Minimi, Medi e Massimi, che, per ogni punto (POI) e per ogni dispositivo utilizzato nella raccolta dati, calcolano la potenza minima, media

e massima percepite da ogni BSSID. Queste tabelle son state pensate prima di effettuare tutte le rilevazioni e sono state realizzate in modo da consentire di prendere più rilevazioni per ogni punto e non una singola e in seguito trattarle in questo modo.

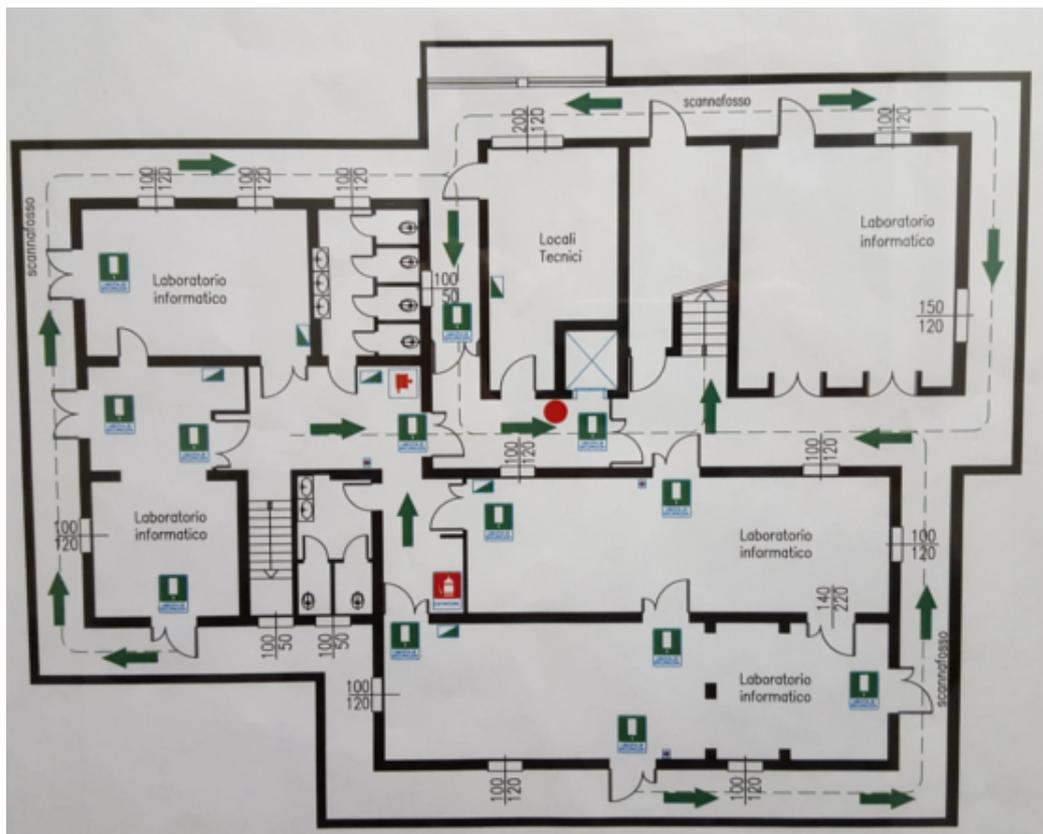


Figura 4.1: Mappa del dipartimento dove sono state eseguite le rilevazioni.

Non si é realizzata una heatmap per ogni BSSID ma solamente per quelli percepiti con una buona potenza.

## 4.2 Sperimentazione

Uno script di test genera casualmente un ID, in modo da prendere un fingerprint qualsiasi dalla tabella generale e sottoporlo ai tre algoritmi utilizzati, con le rispettive varianti, ovvero: utilizzando tutti i dispositivi, solo

---

uno, o tutti meno quello selezionato, iterando questi tre possibili casi sulle tabelle di minimo, medio e massimo. Avremo quindi, per ogni fingerprint testato, 27 possibili risultati (3 algoritmi, 3 tabelle, 3 modalità di scelta dei dispositivi). Il precedente test è stato ripetuto molte volte per ogni dispositivo, in modo da raccogliere molti campioni. In seguito si è calcolata la media e la distanza (utilizzando l'algoritmo di Euclide per la distanza fra due punti in un piano cartesiano) fra il punto reale e il risultato ottenuto, arrivando così ad avere la percentuale di successo nell'individuare il punto esatto o la stanza e il relativo errore in metri. I risultati ottenuti sono stati usati per realizzare dei grafici che mettessero in luce il comportamento dei vari algoritmi in funzione della tabella e del dispositivo/i utilizzato/i.

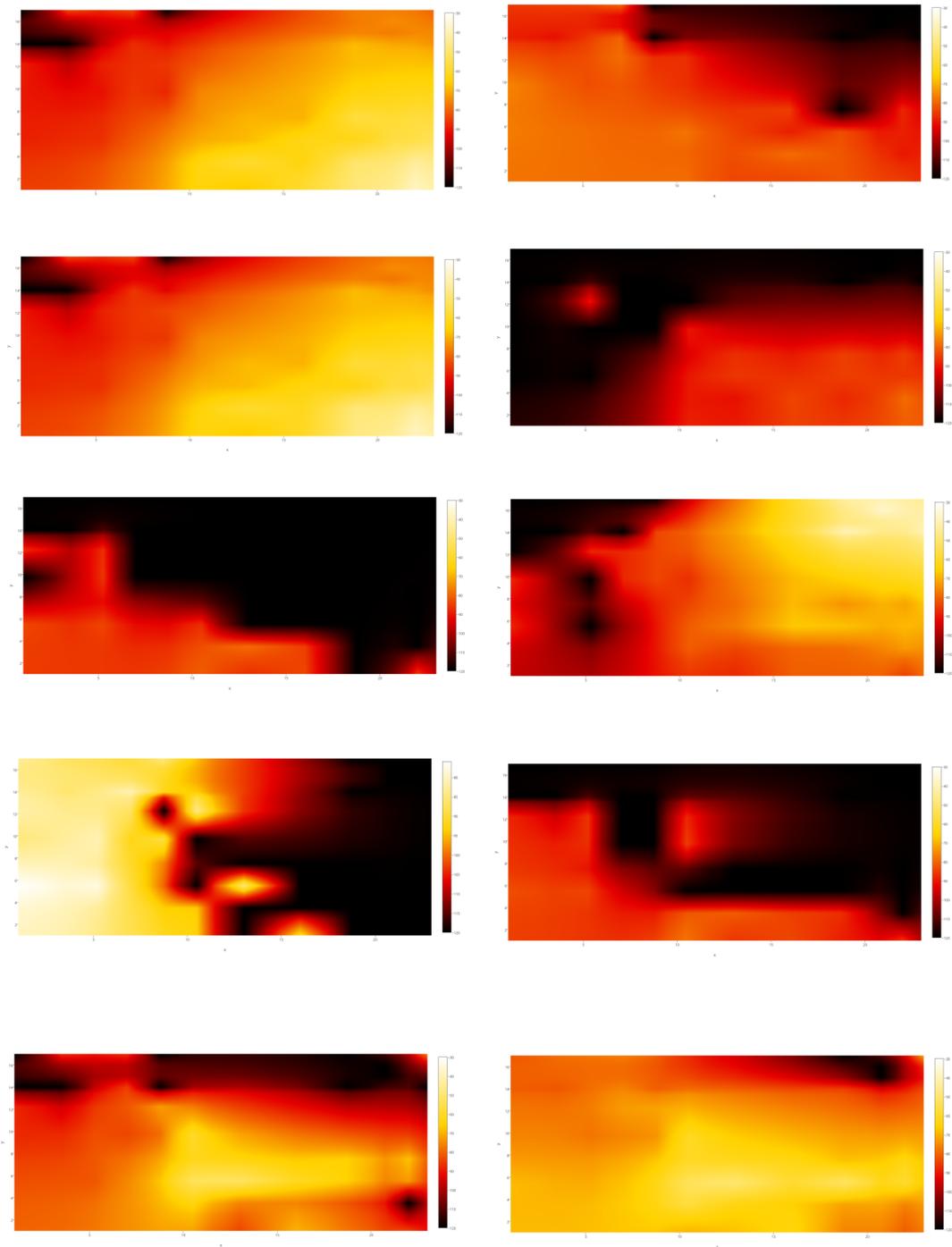


Figura 4.2: 1 - 10 Heatmap, realizzate dopo la prima raccolta dati, rappresentano la potenza di segnale rilevata per ogni BSSID percepito nel dipartimento al momento della raccolta.

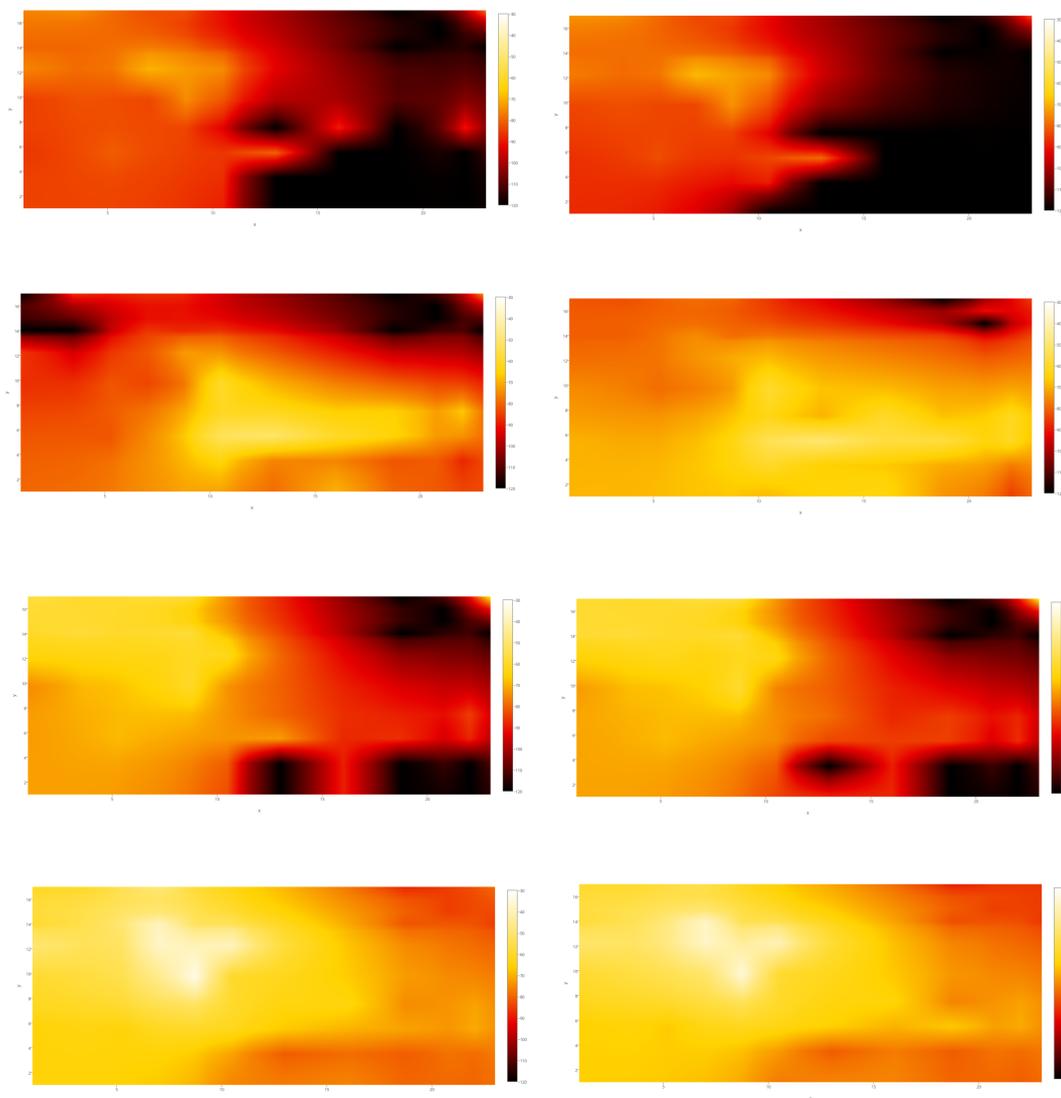


Figura 4.3: 11 - 18 Heatmap, realizzate dopo la prima raccolta dati, rappresentano la potenza di segnale rilevata per ogni BSSID percepito nel dipartimento al momento della raccolta.



# Capitolo 5

## Analisi dei Dati

In questo capitolo vengono analizzati i risultati ottenuti dai test svolti. La concentrazione è sui tre algoritmi utilizzati, sulla percentuale dei risultati ottenuti e l'andamento che essi hanno, l'eterogeneità e l'influenza che ha avuto la fusion rule sui test.

Come già detto in precedenza, si sono relizzati numerosi test sui quattro dispositivi a disposizione, per ogni dispositivo si sono creati tre grafici che rappresentano i risultati ottenuti.

### 5.1 Lettura dei grafici

Ogni grafico può essere letto nel seguente modo:

- i tre differenti colori rappresentano gli algoritmi (Blu - Majority Rule, Arancione - Weighted Majority Rule, Verde - Least Average Error), le colonne sono quindi divise in gruppi di tre, ogni gruppo numerato da uno a nove;
- i gruppi di colonne vanno considerati in sottoinsiemi a tre a tre: 1,2,3 - 4,5,6 - 7,8,9;
- ogni sottoinsieme rappresenta una tabella differente, rispettivamente Minimi, Medi e Massimi.

- all'interno di ogni sottoinsieme avremo quindi tre gruppi, ogni gruppo rappresenta il variare del tipo di dispositivo, rispettivamente, tutti i dispositivi, solo quello selezionato, tutti meno quello selezionato;

## 5.2 Majority Rule

L'algoritmo che ha dato i risultati migliori, in poco più della metà dei casi è il Majority Rule, su entrambi i livelli di accuratezza, sia per il punto esatto che per la stanza. Il Majority Rule ottiene risultati migliori se eseguito sulla tabella dei Massimi o dei Minimi diversamente dai device. Mentre sempre sulla tabella dei Medi ma considerando tutti i dispositivi o tutti meno quello selezionato ha il rendimento peggiore. Per quanto riguarda le percentuali di rilevare la stanza corretta nelle condizioni dove da il meglio arriviamo a ottenere quasi il 90% di probabilità, per il punto esatto invece si supera il 70%. L'errore più alto è si aggira intorno ai 5 metri nelle condizioni peggiori, con uno solo dei dispositivi.

## 5.3 Weighted Majority Rule

Il Weighted Majority Rule ha un comportamento molto simile al Majority Rule, con alcune differenze, in particolare, si nota nel determinare l'esatto punto, con il primo dispositivo nella tabella dei Medi prendendo in esame tutti i dispositivi un rendimento migliore di quasi un 10%. A differenza dell'algoritmo precedente nella tabella dei Minimi otteniamo risultati migliori superando l'90% nella rilevazione della stanza e riuscendo a superare il 70% per quanto riguarda il punto esatto. Il suo errore medio più alto è di 5 metri circa. In sintesi appare come il Majority Rule con una leggerissima perdita di qualità verificatasi quasi esclusivamente sul primo dispositivo.

## 5.4 Least Average Error

Questo algoritmo non mostra mai dei buoni risultati, arrivando ad errori attorno ai 16 metri. Il Least Majority Rule ha un errore così alto probabilmente proprio per il contesto in cui è stato usato, il dipartimento non è così grande e se calcoliamo la media delle potenze percepita in un punto è molto simile a quelle circostanti. In un ambiente più grande si sarebbe notata una migliore prestazione poiché avremmo avuto "impronte" differenti nei punti tracciati. Durante i test il parametro  $m$ , ovvero il minimo di BSSID da considerare, usato era 13, questo significa scartare determinate misurazioni che, in questo contesto, influiscono sull'errore. Il parametro  $m$  condiziona molto il risultato di questo algoritmo, e non è facile trovare il giusto equilibrio in base all'ambiente e agli access point percepiti.

## 5.5 Fusion Rule

La Fusion Rule utilizzata è piuttosto semplice, per permettere la raccolta di più fingerprint per ogni punto e mantenere una certa affidabilità si è scelto di creare le tre tabelle: Minimi, Medi e Massimi. Analizzando i dati si nota che l'utilizzo di una tabella piuttosto che un'altra influenza realmente il risultato, per esempio analizzando il punto esatto, la percentuale di successo del Majority Rule aumenta del 10% fra la tabella dei Minimi e quella dei Massimi. Un altro esempio è la tabella dei Medi, che influenza quasi sempre negativamente tutti gli algoritmi, ma allo stesso tempo permette al Weighted Majority Rule di "vincere" sul Majority Rule nel caso in cui si considerino tutti i dispositivi. Notare che, la tabella dei Medi ha sempre risultati peggiori delle altre, questo è un segno che il sistema predilige i dati reali e non quelli calcolati.

## 5.6 Etereogenità

I grafici generati per il punto, la stanza e l'errore denotano un andamento più o meno costante, con alcune differenze in base al dispositivo. Una differenza in base al pull di dispositivi presi in esame (tutti, solo uno, tutti meno uno) si inizia ad apprezzare solo sul primo dispositivo (OnePlus One), in particolare vedendo i dati numerici, questo è indice che per un numero di campioni circoscritto questo fenomeno rimane latente, mentre più si aumenta la raccolta dei dati più assume importanza la presenza del dispositivo all'interno del database, infatti per questo dispositivo si sono potuti raccogliere quasi tre volte i campioni presi per gli altri. A parte la considerazione appena fatta non si notano grosse differenze in base a quale/i dispositivo/i prendere in esame per quanto riguarda il Majority Rule e il Weighted Majority Rule, invece il Least Average Error nonostante, la sua bassa percentuale di successo ne viene influenzato.

## 5.7 Considerazioni finali

Attraverso questo lavoro si è verificato come la tecnica del fingerprinting possa essere applicata per la localizzazione indoor all'interno di edifici reali, in particolare al dipartimento di Informatica dell'Università di Bologna. Utilizzando algoritmi che stanno alla base delle tecniche più raffinate e con dispositivi differenti si è comunque portato a termine un sistema che possa, con un buon grado di precisione, individuare un dispositivo all'interno del piano. Le analisi finali hanno portato a individuare comportamenti simili nei quattro terminali, questo denota che gli algoritmi utilizzati subiscono poca influenza dalla tipologia del dispositivo, questo effetto è dovuto sicuramente al campione non troppo numeroso, alla relativa contemporaneità dei dispositivi e probabilmente anche dalla poca distanza temporale fra le rilevazioni di ognuno di essi. Questo andamento può spingerci a cercare un perché di questi comportamenti. La tabella dei Medi, si è notato che influisce sempre negativamente, il perché deriva dal modo in cui la si crea, facendo appunto

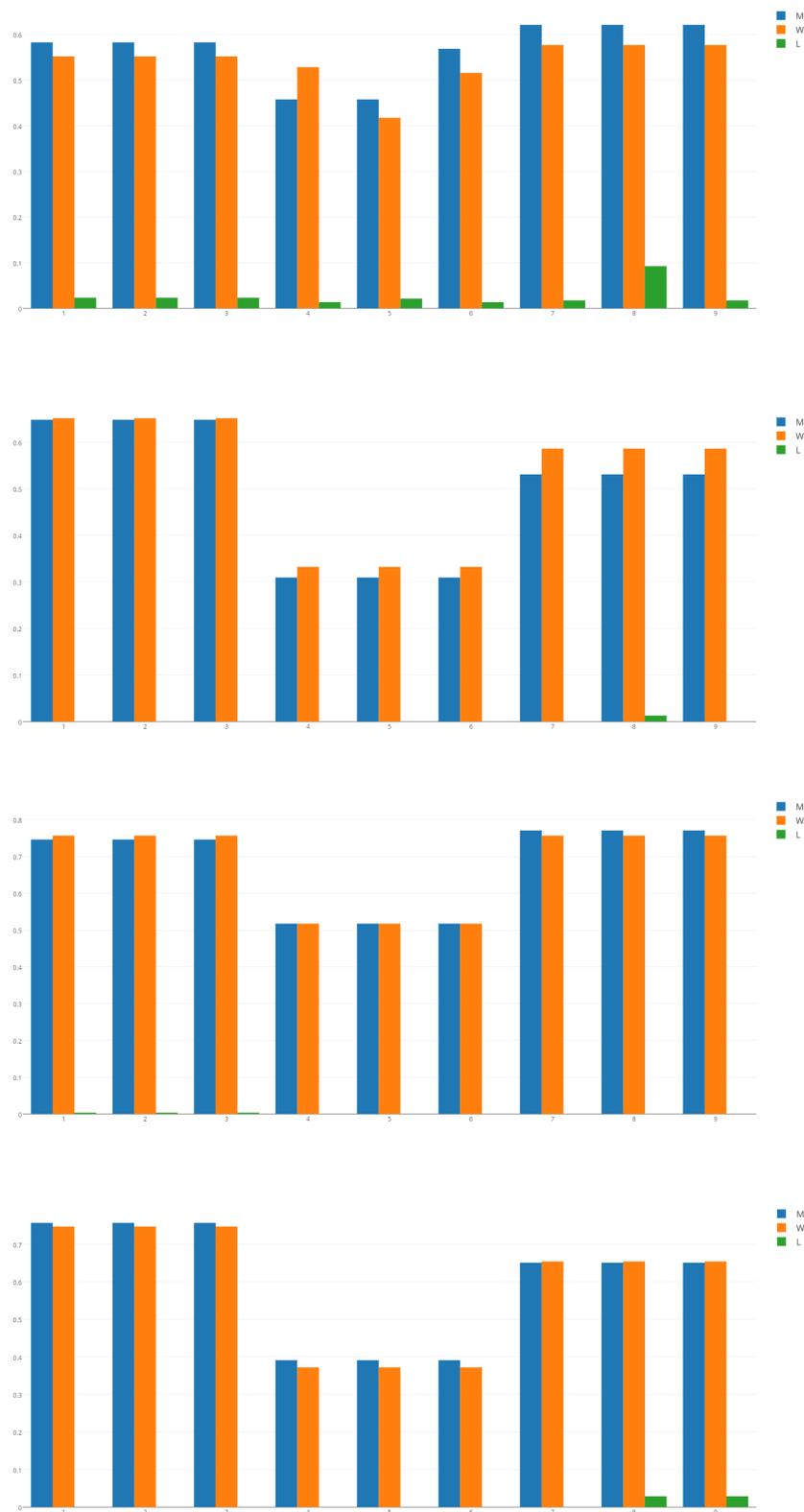


Figura 5.1: 4 Grafici che rappresentano la percentuale di individuare l'esatto punto, nell'ordine OnePlus One, LG Nexus 4, Samsung S2, Samsung S6.

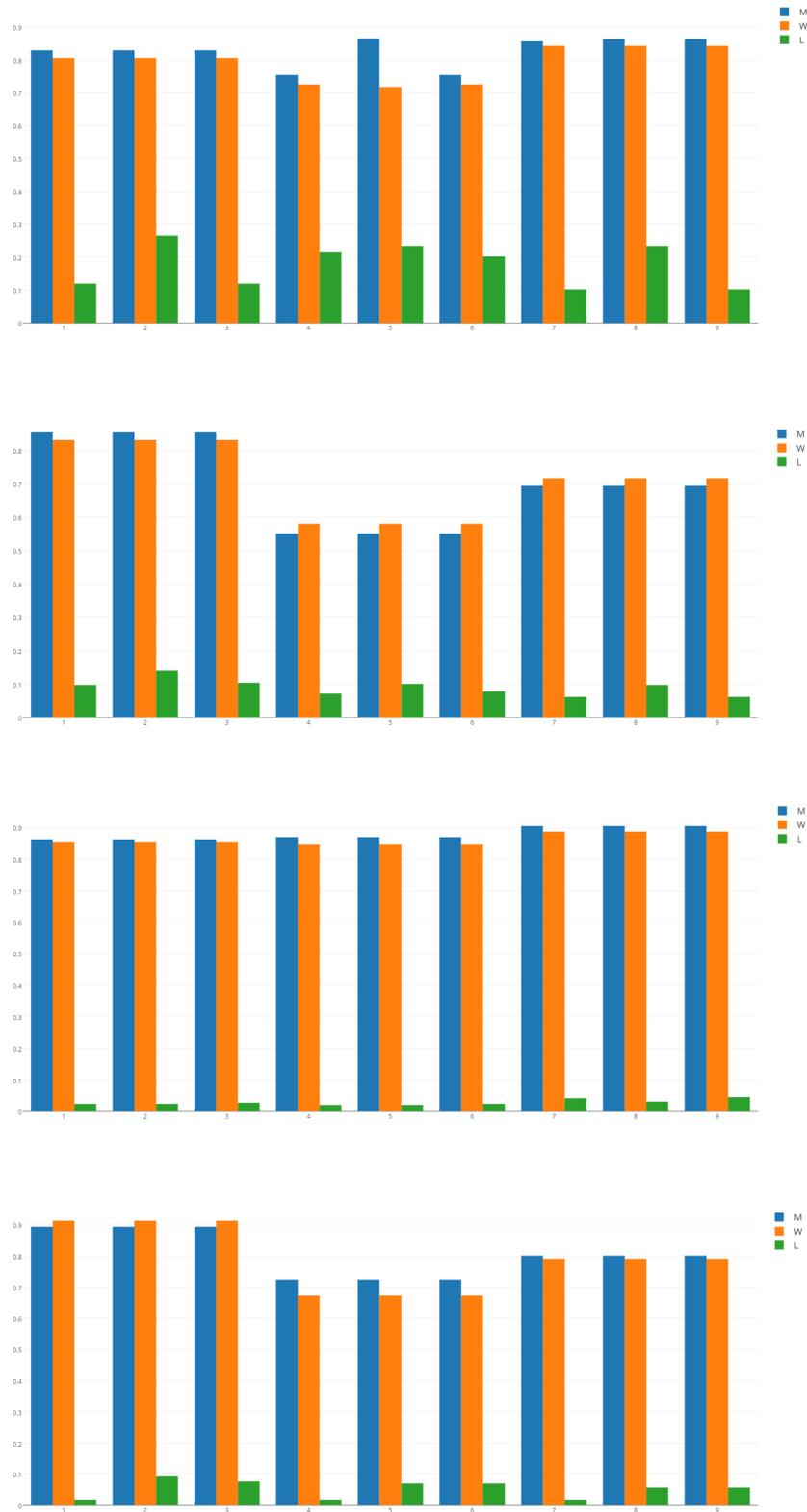


Figura 5.2: 4 Grafici che rappresentano la percentuale di individuare la stanza, nell'ordine OnePlus One, LG Nexus 4, Samsung S2, Samsung S6.

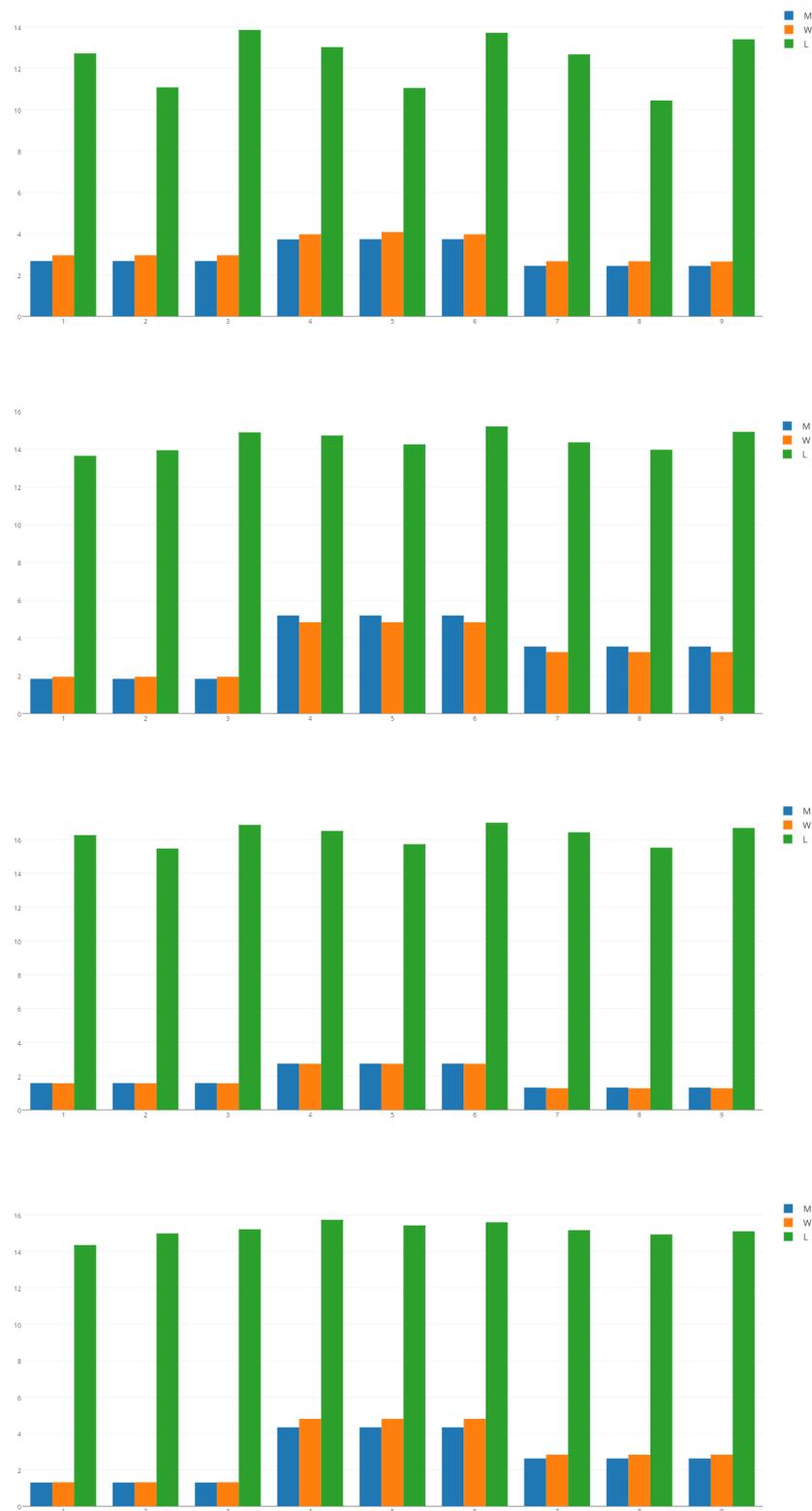


Figura 5.3: 4 Grafici che rappresentano l'errore in metri degli algoritmi, nell'ordine OnePlus One, LG Nexus 4, Samsung S2, Samsung S6.

la media delle misure e quindi introducendo un valore non apprezzato realmente dai dispositivi. Questo, per come sono fatti gli algoritmi, ovvero che tendono a prediligere una precisa corrispondenza, li altera negativamente. Il Majority Rule ha risultati migliori sul dispositivo con un maggior numero di rilevazioni, il OnePlus One, poiché una maggiore quantità di misure implica una maggiore quantità sia delle potenze più alte che di quelle più basse. Il Weighted Majority Rule, il quale predilige le potenze più vicine allo zero infatti subisce una leggera perdita della precisione in questo caso. Anche se, come già spiegato, il Least Average Error non ha dei buoni risultati si può comunque notare che, ove ci sono più misurazioni, tende a crescere sulla colonna corrispondente all'uso del solo dispositivo selezionato; l'algoritmo infatti cerca l'errore più basso che di norma corrisponde alle rilevazioni dello stesso terminale preso in esame.

Una possibile estensione dell'attuale progetto potrebbe riguardare l'introduzione di nuovi modi per il riconoscimento della posizione e lo studio del loro comportamento in rapporto con quello al momento ottenuto.

# Conclusioni

In questa tesi, inizialmente, si è descritto lo stato dell'arte parlando degli algoritmi e della problematica della eterogeneità, successivamente si è mostrato il progetto del sistema e la sua realizzazione composta da applicazione Android e Server PHP. Si sono mostrati i test eseguiti e gli algoritmi implementati, atti alla localizzazione. Infine è stata mostrata un'analisi approfondita dei risultati ottenuti. L'obiettivo fondamentale di questa tesi è stato di sviluppare un sistema per la localizzazione indoor e studiarlo per notarne i comportamenti e capire quali fattori influenzano di più il riconoscimento. L'idea non era quindi quella di creare un nuovo sistema alternativo, ma percorrere i passi che vanno dal concetto allo sviluppo del tutto e comprendere, principalmente riguardo all'eterogeneità dei dispositivi, i punti forti e deboli dello stesso. Il progetto realizzato ha messo in luce che è possibile costruire un sistema del genere e popolare i dati attraverso il crowdsensing. Seppur con algoritmi non troppo raffinati si sono comunque ottenuti dei risultati soddisfacenti e soprattutto differenti in base alla situazione ed alla Fusion Rule utilizzata. Su un campione più grande probabilmente si sarebbero apprezzate più variazioni in particolare riguardo al diverso uso delle tabelle ausiliarie e alla pool di dispositivi presa in esame, i quali, su larga scala, condizioneranno l'esito dell'algoritmo in base alla loro presenza nel database. Da sottolineare il fatto che si è individuato un andamento comune, che si può notare al meglio nel primo gruppo di grafici (Figura 5.1) riguardo al punto esatto. Esso è ovviamente influenzato dal dispositivo ma rispetta una linea guida generale che mostra un diminuire parallelo dei risultati relativo alla tabella dei Medi.



# Appendice A

## Codice Ausiliario

All'interno di questa appendice vengono mostrati degli script ausiliari utilizzati per il corretto funzionamento dei test. Questi script sono stati scritti per semplificare le analisi e per tanto non sono fondamentali al funzionamento di tutto sistema.

### A.1 Script Python

Si è realizzato uno script Python il quale svolge il compito di leggere da un file csv, nel quale sono stati salvati tutti i risultati degli algoritmi, e per ogni risultato esegue la seguente analisi: verifica se il punto è esattamente lo stesso, verifica se la stanza è la stessa, calcola l'errore in metri. Dopodiché scrive i risultati su un altro file csv che verrà utilizzato come input dello script bash.

### A.2 Script Bash

Lo script Bash conta tutte le righe del file e per ogni colonna calcola la media, le colonne sono da prendere a gruppi di tre per ogni singolo punto, poiché lo script Python salva tre risultati ad ogni analisi. Questo script ha permesso la realizzazione dei grafici mostrati nel capitolo 5.

## A.3 Strumenti grafici

Per la realizzazione dei grafici si è utilizzato GNUPlot e in seguito una libreria Open Source chiamata Plotly disponibile per Python, MATLAB, R, JavaScript.

```
50 def dist(pointA, pointB):
51     d = math.sqrt(((pointA[0] - pointB[0]) **2) + ((pointA[1] - pointB[1]) **2))
52     return d
53
54 for linea in f.readlines():
55     results =linea.split(",")
56     good = results[0]
57     print good
58     w.write(good+s)
59     print linea
60     for i in range(1, len(results)):
61         if(good.strip() == results[i].strip()):
62             w.write(str(1)+s)
63         else:
64             w.write(str(0)+s)
65
66         truncG = re.sub('-.*', '', good)
67         truncT = re.sub('-.*', '', results[i])
68         #print truncG, truncT
69
70         if(truncG.strip() == truncT.strip()):
71             w.write(str(1)+s)
72         else:
73             w.write(str(0)+s)
74
75     if results[i].strip() == '0':
76         w.write(str(-1) + s)
77     else:
78         meter = dist(coords[good.strip()], coords[results[i].strip()])
79         w.write(str(meter)+s)
80     w.write(end)
81
82 w.close()
83 f.close()
84
```

Figura A.1: Screen di parte dello script python.

# Bibliografia

- [1] Luca Bedogni, Andreas Achtzehn, Marina Petrova, Petri Mähönen. "Smart Meters with TV White Spaces Connectivity," A Feasibility Study for Two Reference Network Topologies" *In Sensing, Communication, and Networking (SECON), 2014 Eleventh Annual IEEE International Conference, 2014.*
- [2] P. Prasithsangaree ; Telecommun. Program, Pittsburgh Univ., PA, USA ; P. Krishnamurthy ; P. Chrysanthis, "On indoor position location with wireless LANs.", *In Personal, Indoor and Mobile Radio Communications, 2002. The 13th IEEE International Symposium (Volume:2)* ).
- [3] Nicolas Le Dortz ; Florian Gain ; Per Zetterberg, "WiFi fingerprint indoor positioning system using probability distribution comparison", *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on 2012.*
- [4] Houshang Darabi, Member, IEEE, Pat Banerjee, and Jing Liu, "Survey of Wireless Indoor Positioning Techniques and Systems" ,*IEEE Transactions on Systems, Man, and Cybernetics-part c: applications and reviews, vol. 37, no. 6, november 2007.*
- [5] Youngjune Gwon and Ravi Jain DoCoMo Communications Laboratories USA, Inc., "Error characteristics and calibration-free techniques for wireless LAN-based location estimation", *Proceeding MobiWac '04 Pro-*

- ceedings of the second international workshop on Mobility management & wireless access protocols, Pages 2-9.*
- [6] Thierry Denoeux, "A k-nearest neighbor classification rule based on Dempster-Shafer Theory", *IEEE Transactions on Systems, Man, and Cybernetics* (Volume:25 , Issue: 5)
- [7] Francescantonio Della Rosa ; Dept. of Comput. Syst., Tampere Univ. of Technol., Tampere, Finland ; Helena Leppäkoski ; Stefano Biancullo ; Jari Nurmi, ". Ad-hoc networks aiding indoor calibrations of heterogeneous devices for Fingerprinting applications", *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on 2010.*
- [8] Mikkel Baun Kjærgaard ; Dept. of Comput. Sci., Aarhus Univ., Aarhus ; Carsten Valdemar Munk. " Hyperbolic Location Fingerprinting: A Calibration-Free Solution for Handling Differences in Signal Strength (concise contribution)" *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference.*

# Ringraziamenti

Innanzitutto vorrei ringraziare il Professor Bononi che mi ha dato la possibilità di realizzare questo progetto e il Dottor Luca Bedogni che mi ha seguito per tutto il corso del lavoro con grande disponibilità e dandomi sempre preziosi consigli. Ringrazio la mia famiglia per ogni sforzo fatto in questi anni per permettermi di proseguire negli studi e per avermi sempre sostenuto. Ringrazio tutti i miei amici, compagni di corso e non, che mi hanno aiutato con lo studio e fatto ridere quando ce n'era bisogno. In fine ringrazio Elena, per essermi stata accanto in ogni situazione come poche persone avrebbero saputo fare ed aver sopportato con me i momenti più difficili.