# Predictability of Human Mobility from Highly Granular Location Data

Rafaela-Ioana Voiculescu


Kongens Lyngby 2014

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Matematiktorvet, building 303B,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3351
compute@compute.dtu.dk
www.compute.dtu.dk

# Summary

TODO - The goals of this thesis is to..

# Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfilment of the requirements for acquiring an M.Sc. in Informatics.

The thesis deals with ...

The thesis consists of ...

Lyngby, 01-August-2014


Rafaela-Ioana Voiculescu

# Acknowledgements

I would like to thank my....

# Contents

Chapter 1

# Introduction

Chapter 2

# Related work

There is a high interest and a huge amount of work the scientific community dedicates to understanding the patterns of human mobility. The knowledge we can gain from the results of this work has the potential to benefit a wide variety of industries from the modeling and maintenance of the transportation infrastructure, to the medical industry where we can use this knowledge in trying to prevent the spreading of epidemics. [DB08]

Various studies have been conducted in order to gain a better understanding of the human mobility patters. These studies give us results that seem to support each other in the idea that people are less spontaneous than they would like to think themselves and that, indeed, our behaviour shows that we are quite rooted into habits when it comes to the way we travel.

## 2.1 Mobility patters uncovered by the disipation on bank notes

Brockmann, Hufnagel and Geisel[DB06] have analyzed the human movement based on the way bank notes were dispersed through the United States (excluding Alaska and Hawaii). Their study shows that a relatively small percentage

of bank notes (23.6%) traveled for more than 800 km, while a fraction of 19.1% did not traveled for more than 50 km even after a year of being observed. The possible explanation the authors have given for these findings are that, in general, people would be less inclined to leave the areas of the large cities or the places they usually conduct their lives.

The problem identified with this approach for tracking individuals is that the bank notes exchange hands and the behaviour which is identified by the way they circulate can't be attributed to a single individual, but rather to different ones that at any moment have had the bank note in their possession. Despite this, the result have a high scientific value as they do identify patterns in human travel behaviours in general.

## 2.2 Mobility patterns of mobile phone users

A. L. Barabasi, M. C. Gonzalez and C. A. Hidalgo have conducted a study [MCG08] that deals with studying the trajectories of over 100000 mobile phone users with anonymized identities. The study was conducted in order to see if there are any patterns in our mobility habits. Among the things that have been subjected to testing was the return probability of individuals in the same place as in the past. The study shows there is, in general, a peak in the return probability after 24, 48 or 72 since they have left a particular location. This shows that we humans tend to visit locations periodically. This can be explaineQd by our going to places such as work, school, grocery shops near our home etc.

The authors have also ranked the locations the mobile phone users frequented based on the number of times they have been spotted nearby. The results for this have shown that the probability of finding someone near a location that is ranked for them with a level $L$ can be estimated with $1/L$. Another interesting finding that is mentioned in the paper is that, in general, people seem to be spending the majority of their time in just a few locations, while diving the remaining time just between a limited number of locations that varies for the subjects from as low as 5 to around 50.

There are some note worthy plots that the authors present in the paper. They can be seen in figure and they show that most people travel over short distances, yet there is a small number of people that regularly travel over big distances.

The results of this study are a major indicator that individuals display a high level of regularity and that we have a tendency to spend most of our times in places that are familiar to us, or that require us to visit them regularly (e.g.

home, work).

## 2.3 Mobility patterns in massive multiplayer online games

R. Sinatra and M. Szell have studied the way in which users of a massive multiplayer online game behave inside the virtual universe provided by the mentioned game [RS14]. It has been established that the massive multiplayer games provide people with a virtual reality where they can interact with others through their characters and can, in fact, form groups and, as such, display both individual as well as collective behaviour actions that can translate to the non-virtual world [Bal03].

This study gives an interesting insight into the habits and actions of the characters which are controlled by the players. Among the things the authors have analyzed are the predictability of the characters, the entropy generated by the mobility of the characters in the virtual universe and general strategies or patterns that could be observed.

The game the authors have been using for the study is called Pardus [Par]. This game is quite complex, as it allows the manifestation of normal real-life activities such as the creation of alliances or friendships, communication between the players, economic related action, or even actions which have a negative connotation such as attack of another user, removal of a friendship link etc. The universe of the game consists in hundreds of nodes which represent cities or sectors in the game. These virtual cities are tied to each other through links which mark the possibility for the users to move their characters from one place to another.

By analyzing the why in which characters have interacted through the years, the authors have observed that the mobility of the characters through the universe is highly predictable, as users in general will seem to be choosing a random location to visit next in just about 10% of the cases.

## 2.4 Eigenbehaviours

N. Eagle and A. S. Pentland analyze data of individuals and communities with the purpose of trying to predict and cluster the daily habits and behaviour of

people [NE09]. The consider that the behaviour of one person throughout a day can be close to a sum of their primary eigenbahaviours throughout that day. The results of the study have shown that when having a weighted sum calculated for the first half of a day, the behaviour of the same person throughout the remaining of the day can actually be approximated with 79% accuracy.

The results have applicability in more fields, as they allow us to consider the possibility of clustering people into various communities based on the similarity of their behaviours. It goes even further, as the findings show that this enables the possibility of calculating similarity for groups as well and thus permitting the a classification that, according to the experiment, can be 96% accurate for determining affiliations in the social network of a particular population.

As a last observation in the paper by N. Eagle and A. S. Pentland it is stated that eigenbehaviours can be used in order to identify the possible friendship ties between people. The observations in this paper have been done based on the Reality Mining data set that tracked the behavior for 100 individuals at MIT for the duration of one year.

## 2.5 Human movement recorded through real traces

Studies as the ones with the travel of bank notes or the recorded location of mobile users through telephone is not very exact and does not reflect the real traces for the people. They do provide a very useful estimation, however with the technology that we have access to nowadays, we are able to record mobile phone users' real traces either through GPS or Wifi. The data that can be acquired through these means allows us to conduct studies that can take into consideration a very good approximation of the real location of individuals.

In the paper by M. Kim, D. Kotz and S. Kim [MK06], the authors present us with a method in which the locations of users can be estimated based on the WiFi signals that their devices register. The experiment is conducted considering the data for a duration of 13 months. The user traces that have been used consist of the trace data from the Darmouth College. The mobility traces are defined as the lists of access points that are associated to a user's devices at a given timestamp.

The mobility traces allowed the authors to extract the tracks (locations) of the users. They have explored three methods in which the location can be extracted from the data. The first approach presumed the calculation of the center (intersection of medians) of the triangle defined by the past three access

point associations of the mobile device of the user. This approach has a downside since the devices do not necessarily change the associations in a periodic manner. This lead to the second approach which consisted in considering a time window after which the associations needed to be updated in case new associations have appeared during that time. The third and last approach explored the use of Kalman filters [Kal60].

The validation the path extractors the authors have compared the results with GPS data. This validation has prove that the type of the used device has at the moment a significant importance in how accurate the results can be as it seems that some devices can be more aggressive in updating the associations with access points while others try to stay associated with the same access points as long as possible before switching to new ones. This leads to problems as different distances between users and access points considered by different devices and as such it affects the estimated paths. The best estimations have been given in this experiment by the approach that used the Kalman filters, however both the other two approaches have provided fairly good estimations as well.

Another paper which explores the travel patterns from real data is the one written by T. S. Azevedo, R. L. Bezerra, C. A. V. Campos and L. F. M. de Moraes [TSA09]. The authors propose another approach for analyzing the mobility of people. They take into consideration the following movement components: velocity, acceleration, direction angle change and the pause time and they are using the GPS data in order to estimate the locations of individuals. The experiment takes place in a park in Rio de Janeiro and is done based on the data received from around 120 volunteers. The results have shown that people seem to have in general smooth trajectories without abrupt changes.

## 2.6 Entropy and predictability

One step further from understanding the way we travel from place to place is to predict our future locations based on a previous knowledge our our past patterns. There has been an extensive study done in this area of the scientific playground as well and the results which have emerged up until now are remarkable.

In the paper by C. Song, Z. Qu, N. Blumm and A. L. Barabasi [CS10], the authors take up the challenge of studying how predictable people can be. They analyze the mobility patterns of mobile phone users and calculate the entropy of these users. The locations are defined by the telephone towers the users are encountering at hourly intervals and the trajectory of the user is given by the ordered sequence of these towers. The real entropy of each user i is calculated

as $\Sigma_{T_i' \subset T_i} P(T_i')log_2(P(T_i'))$, where $P(T_i')$ represents the probability of encountering a time-ordered subsequence $T_i'$ in the sequence of hourly encountered telephone towers $T_i$.

The results for this particular study show that, for the considered users, the uncertainty of where they could be at a certain moment, based on the real entropy calculated for them would be very low as they would most probably be in one of two locations.

The authors also take a look into the maximum predictability which can be expected for a user. Their results show that, with the right algorithm, a user's future location can be predicted with between $80 - 93\%$ accuracy. This shows that we are less spontaneous than we might think and that our mobility patterns are, in most cases, rooted into a very well established routine.

There have been numerous other methods or experiments conducted in order to analyze or to forecast human mobility patterns. Some of these methods include the Markov chain models [Ros09] [GL96], the neural networks [SCL03] or the Bayesian networks [AS07] as well as some that work with finite automaton [JP04]. Most of the studies support the idea that people's actions and travel behavior is indeed far from being random and thus the science world needs to dedicate further effort and time in order to use this knowledge in order to improve our quality of life and the world we live in.

Chapter 3

# Prerequisites and tools

In order to research the way in which people travel we firstly need to have access to a database of information that can be used for this purpose. As it was mentioned in Chapter 2, scientists have been trying in numerous way to identify and work with location information. During our study, we have dedicated our time in working with information about the access points that were visible to the users' mobile phones throughout their day. This has allowed use to implement and analyze different ways in which locations can be extracted from such information.

## 3.1 SensibleDTU

The data we are using is part of a large-scale study that aims to make observations based on the lives of volunteering students - the Copenhagen Network Study. The data is collected from a variety of sources. Some of them require the volunteers to interact with the system through questionnaires and others track them automatically through their smartphones. The aim of this project is to offer an extensible framework for different studies. The deployments from 2012 and 2013 are based at the Technical University of Denmark and are named SensibleDTU [AS14b].

The students that consented to being volunteers for this ambitious project have received smartphones that are able to track different aspects of their lives and through which they can interact with the system. The big number of volunteers [1] has allowed the gathering of a considerable amount of data regarding the mobile phone users' behaviour.

The data gathered for the SensibleDTU experiment consists in data gathered through questionnaires [2], Facebook data [3], sensor data, qualitative data and Wifi data.

Since the majority of the collected information about the students is sensitive [AS14a], keeping the data secure is and has been a top priority from the beginning of the experiment. The data is annonymzed and stored securely and the students that are part of the experiment have access to tools that allow them to see what data are they sharing, what it is done with this data and that allow them to control how much they want to share.

## 3.2 Using Wifi data

## 3.3 Implementation tools

Before starting the work on the present research, we have overviewed possible tools that can be useful in our work.

The scripts that are used for analyzing, transforming and working with the data are developed in Python. The reasons behind using Python instead of any other programming language are numerous. Python is elegant and simple to use, it allows fast development and the code can be easily adapted and reused. Due to its high scalability, it is the perfect choice for both large and small projects, being easily extensible at the same time. Another very important reason for using Python is that there is a large number of libraries that can be used with it and that allow the visualization or handling of big data. [4]

[1] During the second iteration, there have been deployed approximately 1000 smartphones to students who wanted to take part in the study.

[2] A survey was presented to the participants in 2012 consisting of over 90 questions. In 2013 an addition of over 300 questions were ask per participant. The questioned targeted different aspects from working habits and various socio-econimic factors to Big Five Inventory measuring personality traits [JS99] and self-esteem.

[3] Participants have the option of allowing the gathering of Facebook data such as friendships and various interactions such as likes, statuses etc.

[4] Examples of libraries and packages used: numpy, matplotlib, pickle, datetime, sympy etc.

An additional tool that has been used for the present project is Gephi [Gep]. Gephi is a platform that allows the exploration and handling of various networks and graphs. Further information on how this tool has proven helpful can be found in Chapter 5.

Chapter 4

# Data processing

The present project uses data that has been selected from the database of the SensibleDTU experiment. The data is fully anonymized and the users that have been a part of the study have been chosen randomly from the database.

## 4.1 Statistics

We use the data collected from 131 users from the SensibleDTU database. The students that have been selected for the present study had data collected for a period of almost a year. [1]

The application that is installed on the smartphones of the students who are part of the experiment is configured to scan periodically (around every 15 seconds) for Wifi networks, however, it is also set to record the scans which are triggered by any of the other applications that are present on the mobile phone.

[1] The starting time of collection for the 2012 deployment of SernsibleDTU is October $1^{st}$ 2012 and the end is September $1^{st}$ 2013.

## 4.2 Wifi and GPS data

For the present study we are not using all the fields that are accessible from the database of collected information. The aim of the study is to analyze the predictability and patterns in the human mobility and as such we need information that can help us identify the locations of the users that are part of the study. For this we are accessing fields of the **Wifi information** associated to the selected group of users. The results regarding the users' locations over time are afterwards compared with locations extracted from **GPS data** and as such we are accessing this information from the database as well.

For working with the Wifi information that is available in order to identify user locations, we extract from the database the fields that can be seen in Tab. 4.1.

| user | timestamp | ssid | bssid | rssi | context |
|---|---|---|---|---|---|
| 1 | 1349185621 | 1 | 1 | -75 | 0 |
| 1 | 1349185685 | 4 | 4 | -86 | 0 |
| 1 | 1349185700 | 5 | 5 | -84 | 0 |

**Table 4.1:** This table shows a few examples of possible data recorded from users

A short explanation for each of the fields can be found below:

- The user (first) field gives us information about what user we are currently observing. The real identities of the users are concealed and replaced by an ID which is unique for each of them.
- The timestamp (second) field gives us information about the moment of time at which the scan occured and for which the information is gathered. The time format is Unix timestamp. [2] This timestamp can be easily manipulated and converted to any other timestamp format in Python by using the datetime module that can be found in the Python Standard Library [PSL].
- The SSID (third) field stands for Service Set Identifier and it represents the unique ID that can be used in order to identify the wireless networks. This identifier is responsible for the correct sending of data when multiple wireless networks overlap.
- The BSSID (forth) field stands for Basic Service Set identifier and it represents the MAC address of a wireless access point.

[2] The Unix time stamp represents a way in which time can be tracked as the total number of seconds starting from January $1^{st}$, 1970 at UTC and a particular date and time.

- The RSSI (fifth) field stands for Received Signal Strength Indication and it represents the strength for a signal picked up by the mobile phone from an access point. The RSSI values in our case are registered as the real signal strength recorded in dBm and are therefore negative values. As such, the signal is stronger when the value recorded for it is closer to 0.
- The context (sixth) field is based in the SSID and it translates to the possibilities presented in Tab. 4.2

| context | translation |
|---|---|
| 0 | unknown |
| 1 | AndroidAP |
| 2 | eduroam |
| 3 | dtu |
| 4 | device |
| 5 | eksamen |
| 6 | iPhone |
| 7 | Bedrebustur (wifi on bus) |
| 8 | CommuteNet (wifi on train) |

**Table 4.2:** This table shows the possible contexts for the retrieved Wifi information from the students

## 4.3 Interferences in Wifi networks

Nowadays, Wifi networks are used for a multiple of activities from web browsing to video viewing and even to voice or text communication between people all over the world. As the usage of this technology is expanding so does the need for an even more reliable provided service. The current issue with the Wifi networks is that they are using the IEEE 802.11 protocol [WLP] that uses the 2.4 GHz Industrial, Scientific and Medical Radio Frequency band [Fli03]. This band is, however, unlicensed which means that various devices (Wifi and non-Wifi alike) can use it. This leads to the apparition of interferences.

The results of the experiment conducted by Mahanti et. al. [MCWA10] show that a variety of factors can affect the Wifi networks transmission and signal strengths. For example, microwave ovens, analog wireless video cameras, analog cordless phones and wireless jammers can have a severe impact on the Wifi operations.

However, the issue that causes the most problems in our data set is the existence of signals that come from access points which can be observed for just a very

short period of time as they or the user quickly move by, or that are sufficiently far away from the device and as such their signal level is very low and they can periodically be missing from the scanned access points in the same location [FBSW08].

### 4.3.1 Assumptions about noise and initial data cleaning

Before we have started eliminating the noise in our Wifi data, we have made a few assumptions on what is to be considered noise in the date for the present study. The assumptions are as follows:

- Data received from access point that are part of bus or train Wifi networks are to be ignored (meaning entries that have the context number set to 7 or 8). This assumption was made as it would be hard to determine the characteristics of a given location considering the access points present in buses or trains. For example, a person can take different buses which have a come portion of a route, yet the access points identified by the phone would be completely different and thus the locations would be impossible to be matched based only on this information.
- Data received from hot spots created from Android or iPhone devices (entries that have the context number set to 1 or 6) can also be ignored. These access points are most probably mobile and will not be present in the same locations. This means that they are not reliable when defining locations based on the Wifi networks visible to the mobile phones.
- The signal strength of the registered access points can give information about the distance between the device and the access points and as such it can be a factor in determining what access points need to be taken into consideration when computing the locations. The paper by Zhang et. al. [ZF12] presents the POLARIS system that aims to deal with localization based on Wifi and it also deals with eliminated noise or disturbances in the data. They consider that any signal that has the signal strength indication outside the range of $-60$ to $-99$ dBm can be catalogued as signal disturbances. However, during our data analysis we have observed that the devices can register signals that have a RSSI value above $-60$ (which means that the signal is more powerful) and as such, four our data we consider just the lower bound of $-99$ dBm as a limit for noise. The data registered for access points that have an RSSI value below this one are ignored in order to helps ensure that only the access points who are acceptably close to the device are taken into consideration when trying to determine the location of the user.

### 4.3.2 Data cleaning

Data cleaning is important as inconsistent or incorrect data might lead to inaccurate conclusions and observations. Considering this, the noise elimination in Wifi data is of high importance. Keeping in mind the previously made assumptions (Section 4.3.1), we have eliminated the entries that did not respect the previously mentioned criteria.

During our work with the data, however, we have observed that there are other cases in which additional problems can appear. These situations have been encountered when dealing with the extraction of the mobile users' locations from the information we have regarding the associations made between their phones and various access points. Some of the algorithms used for computing the locations are very time consuming and as such the presence of unnecessary data can burden even further the analysis causing an exponential increase in the execution time.

The situations in which we can struggle with data that does not give any additional information for the identification of locations and that are not necessarily solved by the noise elimination done based on the assumptions presented in the previous sections are caused by the existence of what we will name *isolated observable access points* [3].

Fig. 4.1 illustrates a possible case in which these access points can cause problems rather than help. As we can see there are 7 access points that have appeared in the mobile phone scans over a period of one day. Let us consider that access point AP1 only appears during two consecutive scans, however, it will be taken into account when computing the locations that can be identified for this scenario. A algorithm will identify location L1 and location L2 as being the same location, yet it would do the same thing in case we ignore AP1 and it would require less time to do so. A bad algorithm might not even consider location L1 and L2 as being the same in case the way in which the fingerprint for the location is calculated in a manner that will attribute a high weight to the difference between the present access points.

The above scenario considers a very small number of access points and a very short period of time. The time gains in eliminating the access point which does not provide so much information in this case would be very small. However,

[3]We define as isolated observable access points the access points that are visible to the mobile devices for a very short period of time after which they stop being visible for a long period of time.The reason behind the access point not being visible for longer periods of time can be varied, for example: defective access point, the distance between the access point and the user is increasing very fast in the short period of time between scans etc.

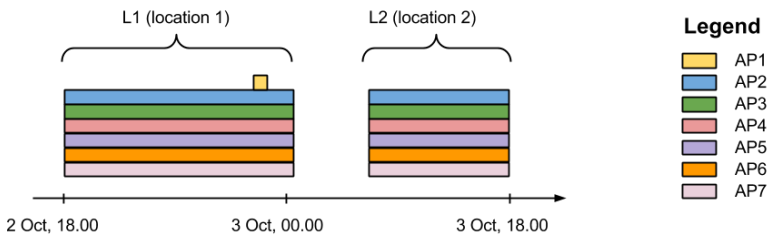


**Figure 4.1:** Example of an isolated observable access point

If we are, for example, looking at a month of collected data for a user, we will have entries for thousands of different access points that were observable at any moment during this time. Out of these entries there can be hundreds of access points which are never visible to the device during close scans and as such their importance when determining the fingerprints for the different locations is very limited, yet they do have a huge impact on the execution time needed to actually extract the locations.

In order to solve this issue, we eliminate from the access points those ones who are not respecting the following condition:

- There is no time window of at least 5 minutes throughout the time duration of the analyzed data in which the access point has appeared for at least 5 times.

We have chosen to use time windows of 5 minutes as we make the assumption that any user will choose to spend minimum 5 minutes at each stop location. In case the user spend less time, we can consider that they are just transitioning until the next stop location.

Chapter 5

# Extracting locations from Wifi data

Human mobility has been attracting a high degree of attention from numerous study fields among which we find urban and traffic planning, traffic prediction, the spreading of diseases and many others [AGB13] [DB08].

The studies that have been conducted on this subject have been using various ways to identify the travel behaviour of people. Some of them have focused on studying the information gathered from observing the way in which money is dispersed through time [DB06], or they have been focusing in studying the behaviour of mobile phone users by analyzing the way they move based on the communication towers their phones are connecting to when they are engaging in voice communication [MCG08]. There are studies that try to understand human mobility through the glass of social networks [YYZS10], as it can be observed that individuals prefer to meet with other people that are part of their community more often [MM07]. GPS data has also been considered for various studies [CLL14], [ZG10]. The list of elements that have been taken into consideration for trying to understand and predict the way in which we are conducting our daily travels is far from being short.

## 5.1 Wifi based positioning

Even from the beginning of the 21st century, research has been actively conducted for trying to use the Wifi system in order to determine real positioning and different databases for positioning systems have been created. These databases usually included the positions of the Wifi access points or RF (radio-frequency) identified fingerprints [CSC+06] [CCLK05] [YA05] [BP00]. Modern databases for Wifi positioning are created with information about the signal strength for the Wifi access points and can even have information about where they were discovered.

Koo et. al. [KC11] have explored an algorithm that can help estimate the relative positions of access points corresponding to the real geographic configuration with the help of multidimensional scaling techniques. Considering the fact that access points are not able to tell real distances between themselves and other access points, the study aims to estimate the dissimilarities between different access points using scans. They have also conducted an experiment in an office building in order to test the proposed algorithm and the results showed an estimation error of approximately 7 m.

Another study conducted in this similar direction is the one by Mok et. al. [MR07]. The authors explore the possibility of determining the location of a device which can scan Wifi access points based on the signal strength that the access points are displaying at the moment of the scan. They estimate the positioning by performing a trilateration based on the information the device gets from multiple access points. The accuracy for their algorithm for the conditions that were present in their experiment was of about $1 - 3$ m.

Athanasiou et. al. [AGGP09] give a very clear and concrete description for two classes of wireless positioning systems. Their work focuses on experimenting with parameters for these algorithms in order to find the optimal solution in terms of accuracy under realistic settings. They also adapt a global map matching algorithm in order to extract travel time maps from wireless data and they propose a demonstration for showing that for high sampling frequencies, the locations identified are comparable to the ones derived from GPS data.

The two classes of algorithms that are explored by the authors are: centroid and fingerprinting. *Centroid* is presented as the fastest method for positioning, however it depends on having the real location of the access points. This information is in general unavailable and as such a proposed solution is to estimate the locations of the access points by calculating an arithmetic mean of all the coordinates at which it was visible. The *fingerprinting* method is based on the assumption that the access points are stable over time (they do not change po-

sitions). This leads to the fact that at any time, a measurement at a particular location will return the same list of access points with the same signal strengths. As such, this list can be considered as the unique fingerprint of the location.

Zhang et.al. [ZF12] propose an algorithm based on fingerprinting for estimating locations that takes into consideration the fact that the signal strength from various access points does not necessarily stay constant throught the time. They propose a way in which a similarity between fingerprints can be calculated in order to determine if two fingerprints are in fact representing the same location.

These are just a selection of works that have been conducted on finding a solution for Wifi based positioning systems. With the growth and improvement of Wifi systems, in time all barriers can be overcome and we could have a positioning system that is as accurate yet considerably cheaper than GPS positioning systems.

## 5.2 Determining the fingerprint of a location

In order to have a better understanding about the way in which the mobile phone users have been moving throughout the experiment, we needed to have an image of the way a given period of time would look based on their Wifi records from SensibleDTU. As it has been presented in Section 4.2, the Wifi data we are using for the present project consists in the following fields: user id, timestamp, SSID, BSSID, RSSI and the context. However, considering the amount of data involved, just by looking through the log files it is almost impossible for us to understand at what moment the user might have reached a location and when did they leave from it. In order to be able to do this, we have created various visualizations considering different options, different time frames and for multiple users in order to begin to understand what the data can tell us, what can we use, what would we need and what can we discard when moving further to defining what makes a location.

### 5.2.1 Signal strength over time

The first thing that we have tried to visualize was the access points (APs) that were scanned by users' mobile phones throughout different periods of time. We have plotted the APs and their registered signal strength for varied users in order to see if we notice any patterns in their movements.

In Fig. 5.1 we can see how a day from the life of a random user (referred to as userX) looks like. The day for which we have plotted the data started on a Tuesday at 12 : 15 pm and ends the next day right before the same hour. The hourly intervals can be seen on the X axis, while the signal strength values can be seen on the Y axis. The legend contains the top 10 most popular [1]

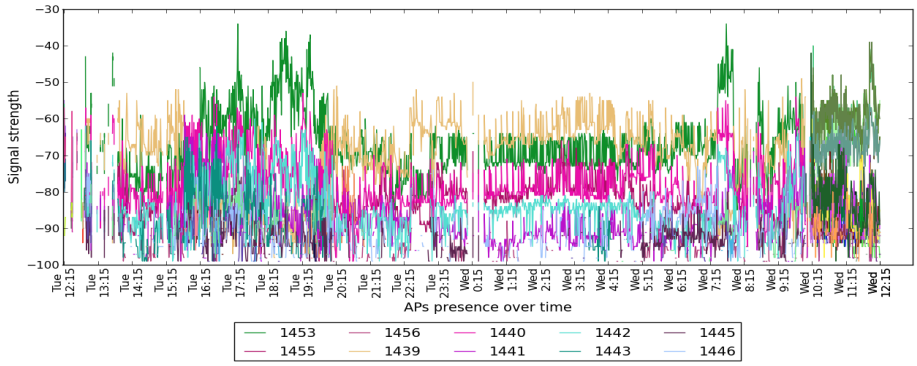


**Figure 5.1:** Example of the APs registered for an user throughout one day (using connecting lines markers)

The steps for creating this type of visualization are as follows:

- Retrieve data for the time duration for which the visualization is made
- Keep track of all the timestamps at which each AP has been seen and the AP's signal strength at that moment
- In case an AP is scanned no more than 2 minutes after a it was previously scanned, then a line can unite the two moments in order to mark their proximity. If the apparitions are more than 2 minutes apart there is a high possibility that there has been a location change or that the AP is experiencing technical problems and as such has stopped being active.

Although we have tried to visualize this type of information in various ways (using different types of markers), we found that this way is the easiest to interpret by people. If we leave out the lines, for example, as it can be seen in Fig. 5.2, it is quite hard to interpret where location might start or stop.

Other ways in which we have been experimenting with visualization for this can be found in Appendix A.1.

[1] An AP is more popular than another in case it appears more times during the period of time for which the Wifi scans are analyzed

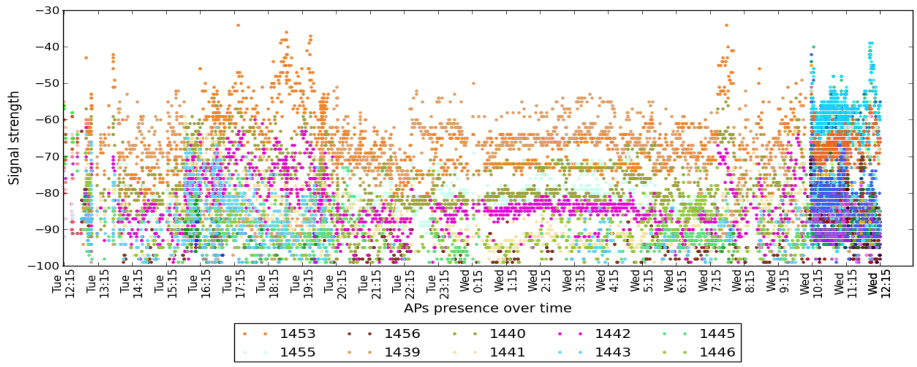


**Figure 5.2:** Example of the APs registered for userX throughout one day (using point markers)

By looking at Fig. 5.1 we are at some level able to distinguish moments of time at which the user seems to be arriving at a location [2], however it is hard to notice any patterns because we are only observing a single day in the life of userX.

Let us look at the data gathered through 7 days from another user's (referred to as userY) life. The visualization for this data can be seen in Fig. 5.3. The image gives out some very interesting information. We can, for example, notice the repeating patterns which are dominated by the orange, light green and blue colors. These patterns appear during the evening and the night and we can assume that the user is spending this time at the location which we can label "home".

We can notice some periods of time that are free. These free gaps like, for example, from Monday morning until Monday evening are gaps in which no signal was scanned and can mean that either the mobile phone was closed or that the user decided to switch off the Wifi.

We can also notice fragments in which the density of signals is quite high, for example on Wednesday morning. This means that the user was located in a place which has a large number of APs near and since we can notice a regularity in this pattern we can assume that this place can be the University. This might seem unlikely based on the fact that the patterns sometimes is identified during the night, however this particular week is set in October when there are deadlines for school projects that need to be handed in.

[2]For example, we can say that what we notice from Wednesday at 10:15 until the same day at 12:15 is different than anything we can see before that time so we can assume that it is a new location.

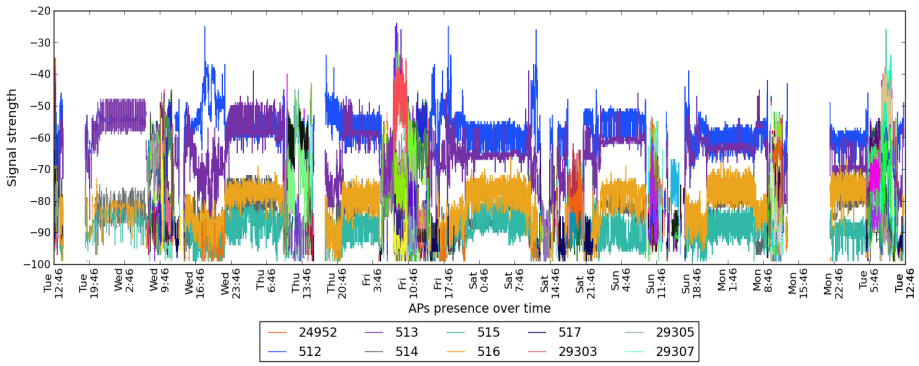


**Figure 5.3:** Example of the APs registered for userY throughout 7 days

As we can see, these visualization can offer us a good first glance at what the locations might be like, yet they also make us consider other things that we can learn about the data. For example:

- How many samples from each access point are received during a given time frame
- What is the average signal for various time frames for a given access point
- What are the running averages for signals from various access point

### 5.2.2 Sample density

When trying to identify locations based on the Wifi data, it is important to only take into consideration the access points that actively contribute to the fingerprint of the mentioned location. Before cleaning our data (as it has been described in Section 4.3.2), isolated observable access points can appear and unnecessarily burden the algorithm used for extracting the locations. The best way to identify such access points is by analyzing the sample density [3] of the samples that are identified during scanning.

In order to determine the sample density for each AP, we need to define a time bin over which the sample density needs to be calculated. We have calculated the density considering a time bin of 5 minutes as we can assume that this amount of time can be considered the minimum duration for which a user needs

[3] We define the sample density for an access point as the number of times it appears in scans over a predefined time bin.

to be situated in approximately the same place in order for us to not consider that the location is a transition instead of a stop location.

In Fig. 5.4 we have the different APs and their RSSI values at the different moments when the mobile phone has identified them in the scans for userX during the second day of observations. In Fig. 5.5 we can observe the sample density for one of the APs that are predominant during the visualized time frame. As we can see, the number of times the AP is present in the scans throughout the day is quite high and it is registered during numerous different periods during the day. We can easily assume that this AP is one of the key APs that define one of the locations the user has been associated with.

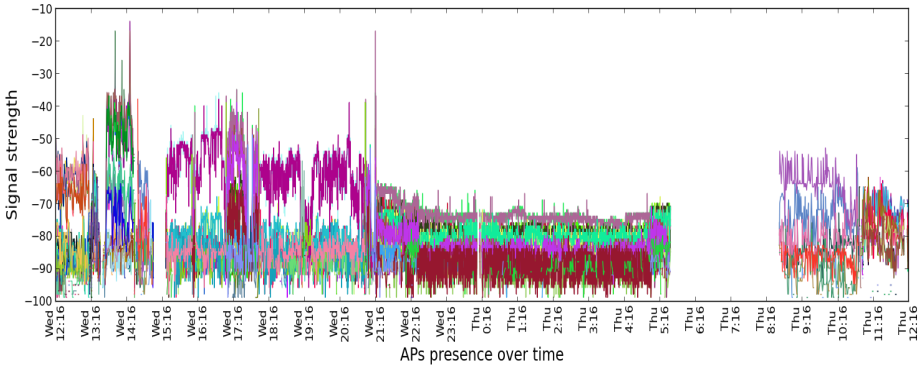


**Figure 5.4:** Example of the APs registered for userX throughout day 2

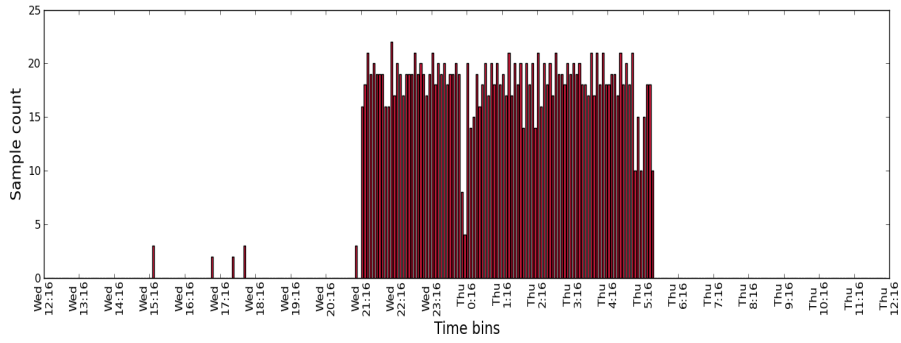


**Figure 5.5:** Example of an AP which appears often

On the opposite end as number of times it has appeared during the scans, we have the AP in Fig. 5.6. As it can be seen, this AP only appears 5 times over a one single 5 minute time bin. We can easily presume that the presence or absence

of this particular AP will not offer us relevant information over the location at which the user was situated when it appeared in the scans. This statement is also sustained by the fact that the user location seems to be consisted from Wednesday $12:16$ up until around $13:16$ according to what we can observe in Fig. 5.4, even though the AP does not appear throughout most of this time.

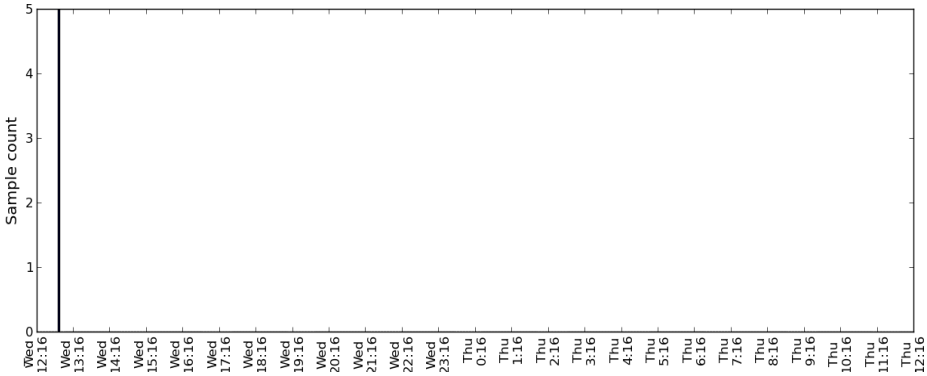


**Figure 5.6:** Example of an AP that appears just a few times

Other examples of visualizations for APs based on their sample density can be found in Appendix A.2

### 5.2.3 Exploring the implications of the signal strength

Something that is often taken into consideration during studies regarding the determination of locations based on Wifi data is the value which indicates the signal strength received from the various APs. The level of the signal strength indicator can, in general, give us a good approximation of how close we are to a particular AP. However, Wifi networks are susceptible to interferences [MCWA10], meaning that there numerous factors which can cause signals to spike even in case the device which scans the region for AP signals does not move. This can represent a factor of risk when including the signal strength value in the location extraction from Wifi data as the same location could be, at different times, be associated to an AP which has a signal strength that oscillates based on other external factors.

In order to see if we can smooth down possible fluctuations we have employed two mathematical tools. We have calculated the average signal strength, as well as the running average, considering different length time bins.

### 5.2.4 Average signal strength

In order to calculate the average signal strength of a given AP for a given time bin, we needed to identify all the moments of time inside the given time bin in which the AP has been spotted during the scans. The average signal of the AP is calculated as the sum of all the strength values that have been recorded for the AP inside the time bin and the sum is then divided to the number of recorded apparitions of the AP. For example, if we were to have an AP which appears 6 times inside a 5 minutes time bin with the following RSSI values [-60, -70, -60, -80, -90, -60], then the average signal strength for this particular time bin for our AP would be $avg = [(-60) + (-70) + (-60) + (-80) + (-90) + (-60)]/6 = -70$ dBm.

We have calculated the average signal for various users and various days. We have also calculated it for different time bin length. For example, for the same data that we can see in Fig. 5.4 and for the same AP that has the sample density represented in Fig. 5.5, if we visualize the non-null averages calculated for time bins of 5 minutes, we would have the representation in Fig. 5.7. The X axis records the time while on the Y axis records the values of the averages

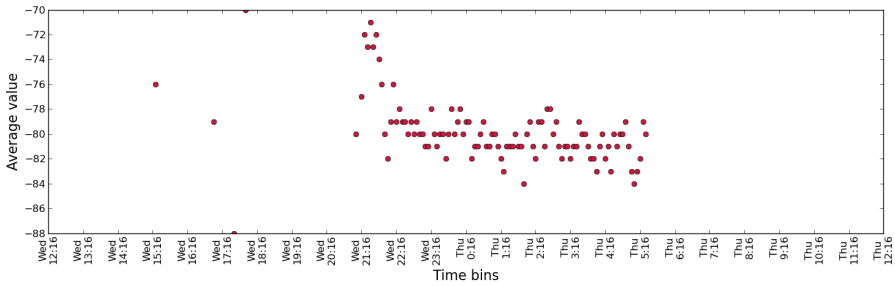


**Figure 5.7:** Example of average signal strength visualization for userX

The averages are represented by big dots symbols which appear at the beginning of the time bin for which the average is calculated. For example, if we have calculated an average for the interval $12:05 - 12:10$, the average is plotted on the visualization at $12:05$.

Additional examples of averages for different APs scanned during the same day by userZ's mobile phone can be found in Appendix A.1.

### 5.2.5 Running average signal strength

The average signal brings a small improvement as far as eliminating the signal spikes go, however, an even better way in order to smooth out any signal fluctuations is to calculate the running average[4] [Hyn09].

We have calculated the running average for different users and time frames, and we have taken into consideration different time bins when calculating it. The algorithm for calculating it is as follows:

- For the selected user and the selected time frame, we have extracted for each AP the time stamps at which it has been identified by the user's phone
- We have divided for each AP the previously mentioned time stamps into bins of 2, 5 or 10 minutes recording also the signal strength identified at each time stamp[5]
- The above identified time bins are overlapping. For example, if a sequence of signals $[-60, -80, -70, -70]$ that have each been identified at 1 minute apart is to be divided into bins of 2 minutes, the resulting 2 minute bins would be: $[-60, -80]$, $[-80, -70]$, $[-70, -70]$
- The running average is calculated as the sum of the values present in a time bin which is then divided to the number of values. For example, for the above time bins, the running averages would be $-70$, $-75$ and $-70$

In Fig. 5.8 we can see the APs associated with another user (referred to as userT) and their signal strengths over a day. Fig. 5.9 shows the signal strength for just one of the identified APs. The average signal as is presented in Section 5.2.4 for the same AP can be seen in Fig. 5.10. Fig. 5.11, Fig. 5.12 and Fig. 5.13 present the running averages calculated for the same AP for time bins of 2, 5 and 10 minutes [6]. The X axis of these figures track the succession of time moments while the Y axis keeps track of the value of the running average calculated over this time.

The way in which the fluctuations are smoothed down can be easily seen in the figures that present the running averages calculated for various time bins. The fluctuations are smoother as the time bin is increased.

[4] Also referred to as the moving average

[5] By doing this we have the signal strength for the given AP at any moments it has appeared inside the time bin

[6] In this representation, only the non-null values for running averages are displayed

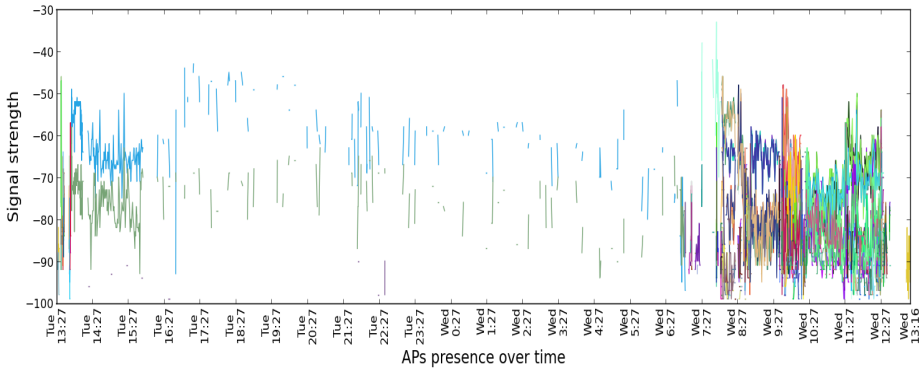


**Figure 5.8:** Example of APs presence over time for userT

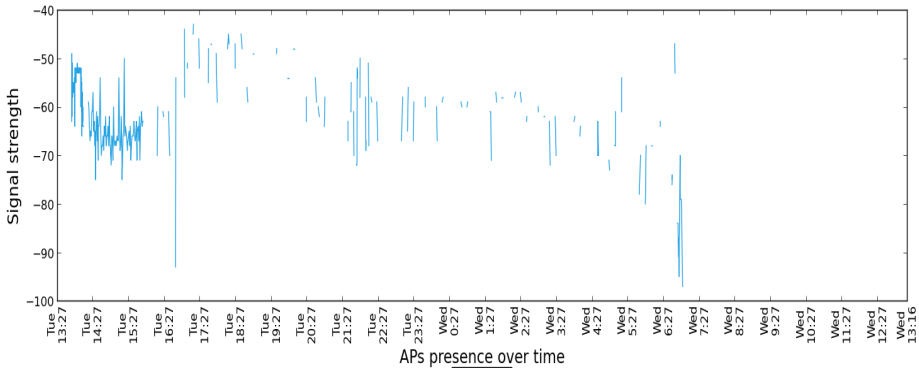


**Figure 5.9:** AP 85 for userT during 1 day

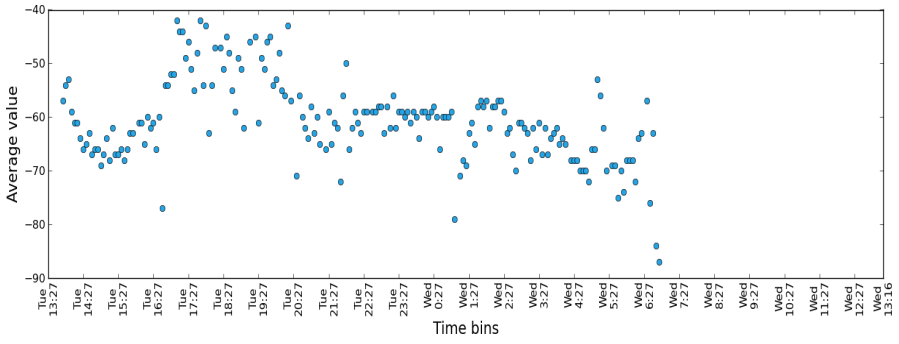


**Figure 5.10:** Average strength for AP 85 for userT during 1 day

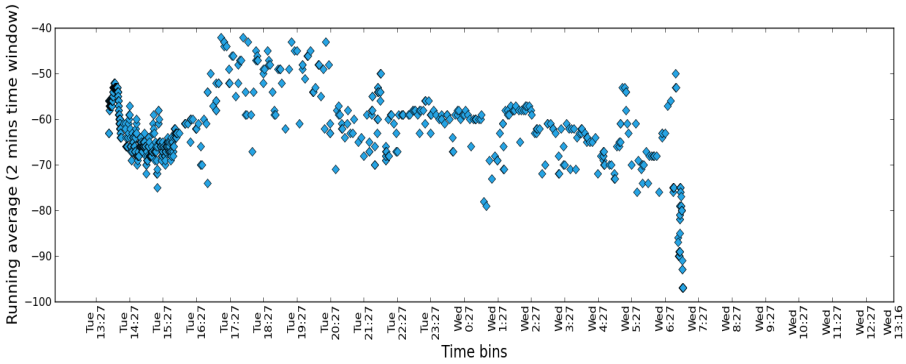


**Figure 5.11:** Running average for AP 85 for userT during 1 day (2 minute time bins)

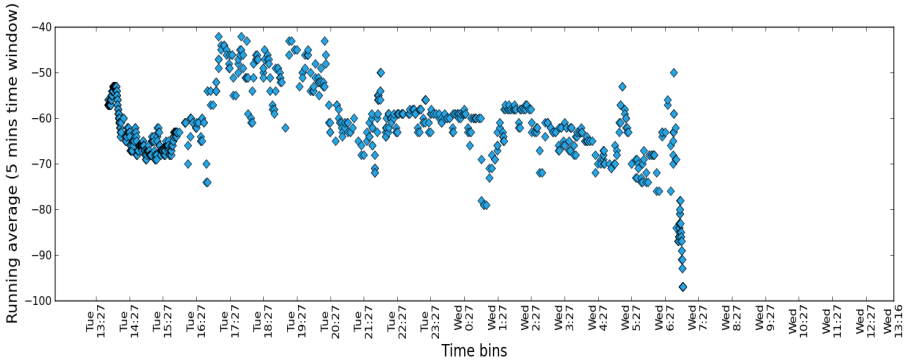


**Figure 5.12:** Running average for AP 85 for userT during 1 day (5 minute time bins)

Visualizations for running averages calculated for other APs identified during the same day for userT can be found in Appendix A.2.2.

### 5.2.6 Signal presence

Even though averaging the signal strength through time improves at a certain level the fluctuations in the signal strength, in a real environment spikes will always be present and this will bring extra difficulties in estimating locations based on fingerprints that contain the value of the signal strength for the involved APs.

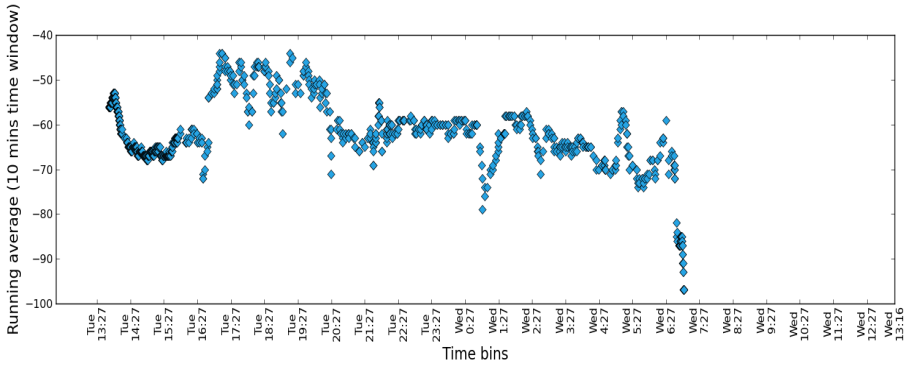


**Figure 5.13:** Running average for AP 85 for userT during 1 day (10 minute time bins)

Another way of looking at locations is by calculating their fingerprint based only on the identity of the APs that have been identified while the user was found at that particular location. Basically, instead of defining a location based on both the identity of the APs present and their signal strength, we would only associate locations to visible APs.

The idea is simple and elegant and has been used in previous studies with success [LJ09]. The concept behind is that, in general[7], at a given location the scans will always show the presence of the same APs. If, after a time, the scans change and other APs appear, it is reasonable to assume that the user has changed locations.

Since the information offered by the signal strength does not seem to be of the ultimate importance, we can, in this case, try to identify the locations only based on the presence of the APs. We consider that an AP is present at a specific moment of time if the Wifi scans at that moment register a signal strength from that AP. However, as it has been mentioned previously, due to interferences, the signal from the AP might be lost for short periods of time even when the user does not change their location.Considering this and the assumption that, in general, people tend to spend at least a few minutes in a stop location (otherwise meaning that they might be just transiting it), we have made the decision to adapt for our case the definition for the presence of an AP.

We divide our data into time bins of 5 minutes [8]. We redefine the presence of an

[7]New APs can be set up or old ones can be changed with new ones in time, which would mean a change in how the scans would look for the same location. However, this is an issue that is outside the scope of the present paper and work.

[8]We consider 5 minutes as the minimum amount of time that needs to be spent in a

AP as follows: an AP is considered to be present for the duration of a 5 minute time bin if it appeared in the scans at any point inside this time interval.

We can use visualization in order to see how this transforms the way in which we can understand the data. In Fig. 5.14 we have the different APs that have been scanned throughout the duration of 2 days for userX. In Fig. 5.15 we can see the top 50 predominant APs and their presence over 5 minutes time bin during the same 2 days [9]. The X axis keeps track of the time bins throughout the 2 days, while the Y axis represents the annonymized identifiers for the APs.

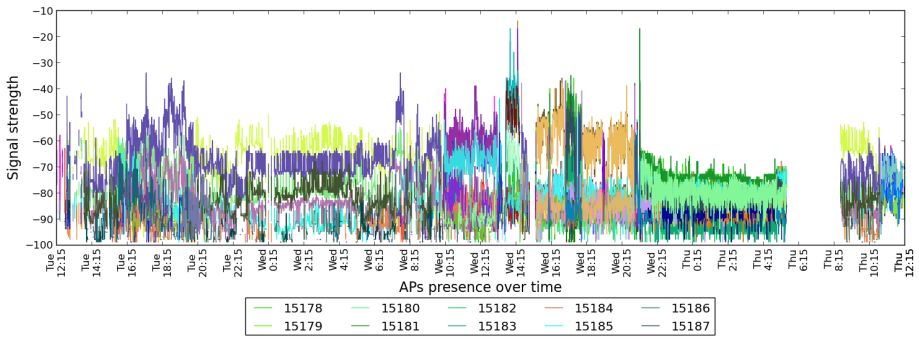


**Figure 5.14:** Scanned APs for userX throughout a duration of 2 days

By closely observing the two visualization, it is quite easy to see that indeed they are representations of the same period of time. Even if not all APs are displayed in the visualization for the presence of the APs over time, we can notice that, for example, the user has spent the time from Wednesday 21 : 15 until almost Thursday 6 : 15 in one location. This also coincides with what we can observe in the visualization for all the APs (with signal strength) scanned throughout this time.

In Appendix A.2.3 can be found a visualization for the presence of APs for a period of 2 days for another user. The presence for APs is determined for 5 minutes time bins over the 2 days.

location for it be considered a stop location. This number can be easily adjusted in case further research shows that it is not the optimal assumption.

[9] We restrict our visualization to 50 APs as it would be hard to understand an image in which we would be displaying all the hundreds of APs which were encountered throughout the 2 days.

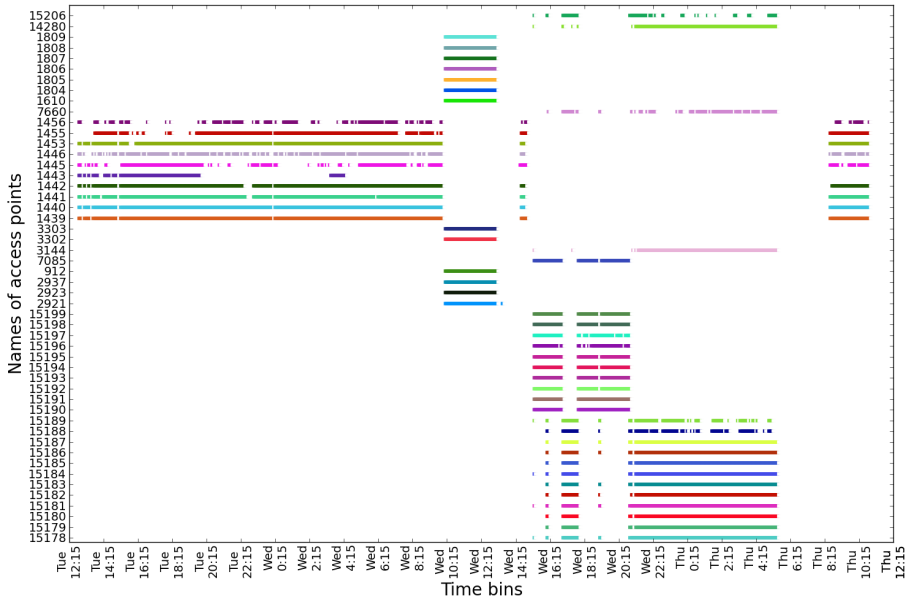


**Figure 5.15:** The most common 50 APs for userX during the given 2 days (presence visualization calculated for 5 minutes time bins)

## 5.3 Extracting locations

By visualizing the Wifi data in the way presented in Section 5.2.6, we can begin to see how locations seem to succeed each other throughout the days of a particular user. However, it is important to be able to implement a solution that will extract these locations from a large amount of data so that we would not be needed to examine the data manually. We have used different methods in order to get the best possible approximation for identifying the locations. The methods we have tried are: using *networks*, using *k-means clustering* and using *Hidden Markov Models*.

Before describing each of these methods, we have to clarify what we consider a fingerprint of a location at a given time. A fingerprint of a location is calculated based on the AP presence inside 5 minutes time bins (Section 5.2.6) as follows:

- We extract the data we want to analyze from the user (either for 1, 2 or more days).
- We identify the APs from the data

- We divide the data into 5 minute time bins
- For each time bin we identify the APs which have been spotted during the 5 minutes and we attribute them the value of 1 (meaning that they are visible to the mobile device during that time bin); the remaining APs will have attributed the value 0 (not visible) for the given time bin.
- Each fingerprint describes a time bin and shows what APs were visible during it and which are not

for consecutive 5 minutes time bins. The fingerprint contains the names for all the APs which are associated to the user throughout the time frame we are analyzing (e.g. 1 day, 5 days etc.) and each AP has associated to it a value which represents its presence throughout the 5 minutes

### 5.3.1 Network theory

Networks have a high degree of importance when trying to understand human or animal behaviour. They have been used in combination with social platforms in order to extract a new definition of friendship [CML11], they have been used for monitoring animal behaviour [GCP+06], or to understand the economical situations cause by the way in which people interact [CJK05], or just to understand underlying communities of people. During our research, we have considered the use of network theory in order to extract locations from Wifi data.

In theory, we can expect that the APs that are identified at a particular location will not appear in the scans the user's phone will have from another location. This assumption is sound as the APs will rarely be moved and as such they should always be associated to the same place. In this case, we should expect that the succession of locations can be similar to what we can see in Fig. 5.16., where AP1-AP6 are associated to location number 1, while the remaining APs are associated with location 2 and the APs never overlap.

The idea behind constructing the network that can be used to extract the locations is simple. A graph can be created for each user from their data and the locations can be identified as follows:

- We consider each found AP from the user data as a node in the created graph
- We construct a presence matrix for the identified APs. Each line in the presence matrix is associated to an AP and contains the signal pres-

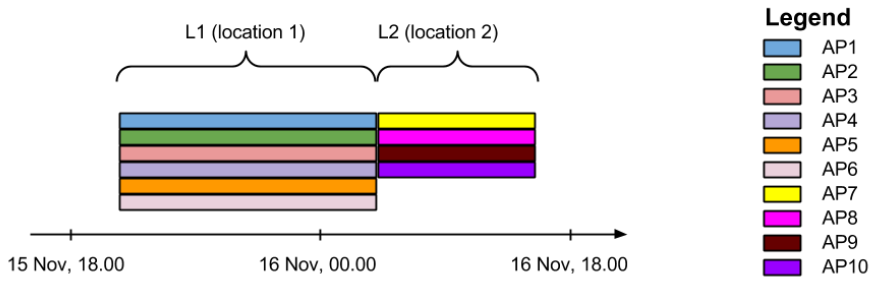


**Figure 5.16:** Example of how, in theory, locations should be displayed through the presence of APs

ence (Section 5.2.6) calculated for the AP considering 5 minute time bins throughout the time we are evaluating

- For each time bin, we identify the APs that are present trough it and we connect each two of them with an undirected edge (in case they have not been connected at a previous time)
- Since signal from various APs can be lost due to interferences, for each two APs for which we have created a connecting edge, we keep and update a variable which represents the number of times the APs have been identified in the same time bin
- After the network is completely created, we normalize the counts of how many times each two APs have been seen in the same time bin by dividing the counted value to the maximum number of apparitions of either of the two access points
- After the normalization we remove the weak links [10]
- We consider the resulting connected components to be the extracted locations

An example on how to construct such a graph if given four APs and their presence matrix can be seen in Fig. 5.17

We have applied the previously described algorithm for a selection of users, but the results have not been satisfactory.

[10] In this case, a link is considered weak if after normalization its associated value is below a given threshold

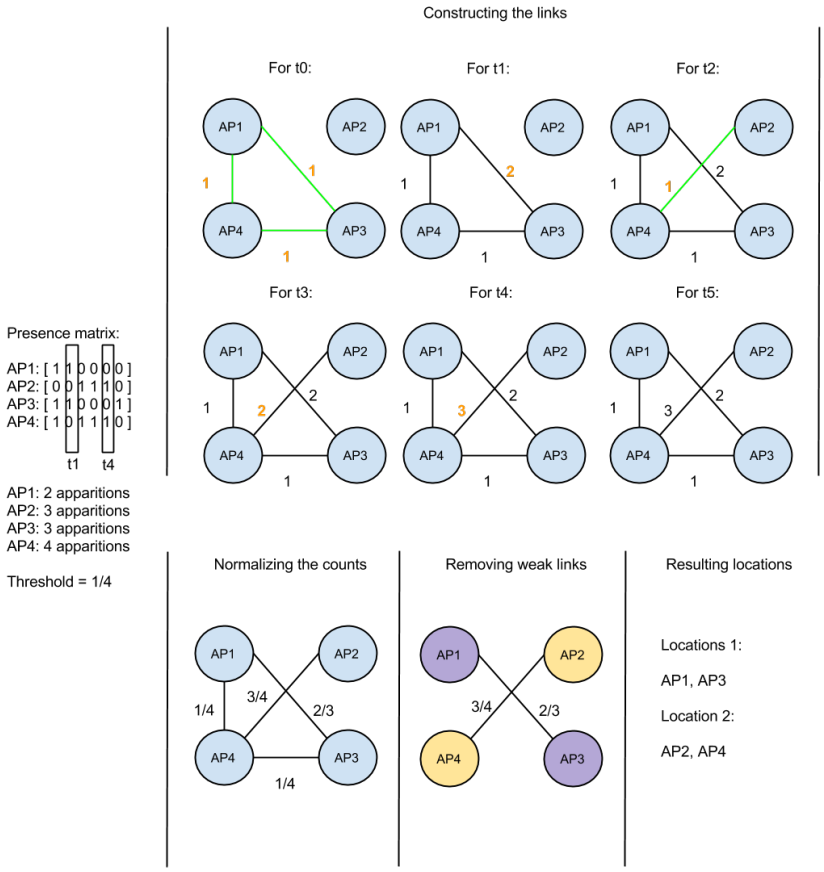


**Figure 5.17:** Example of constructing a network

For example, we can take data for one day for userX. The visualization for the identified APs and their presence throughout this time can be observed in Fig. 5.18. By looking at this image, we can observe that the user has been in f2 main locations during this day.

When running the algorithm that extract the locations based on the constructed network, we obtain 21 connected components which have the potential of being locations. The image for the connected components can be seen in Fig. 5.19

We have tried to adjust the threshold for eliminating weak links yet the results are in most cases unsatisfactory. Upon closer analysis we have observed that this

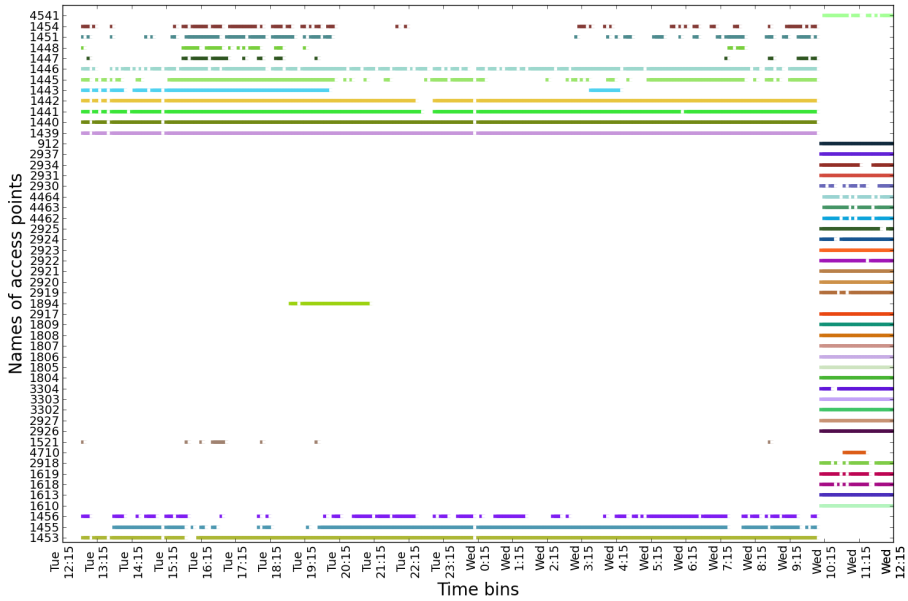


**Figure 5.18:** The most common 50 APs for userX during one day scan records

can be cause by the fact that, sometimes, there are a high number of APs that even though they are in reality tied to a given location, their signal fluctuates often and as such, the algorithm identifies them as part of a different location and as such we have locations consisting in only a very small number of APs that in reality could have been integrated in other locations. This observation is sustained by the size distribution of the generated networks (example for such a size distribution determined for the network in Fig. 5.19 can be seen in Fig. 5.20). Another thing we have observed is that there are adjacent locations which can have interfering APs signals. This means that our original supposition that, at all times, the APs that are visible from a location will stop being visible in any other location does not always hold.

For generating the networks and the size distributions, we have been using Gephi [Gep]. For visualizing the visualization in Fig. 5.19 we have used the Force Atlas 2 layer which has been configured in order to avoid overlapping of components.

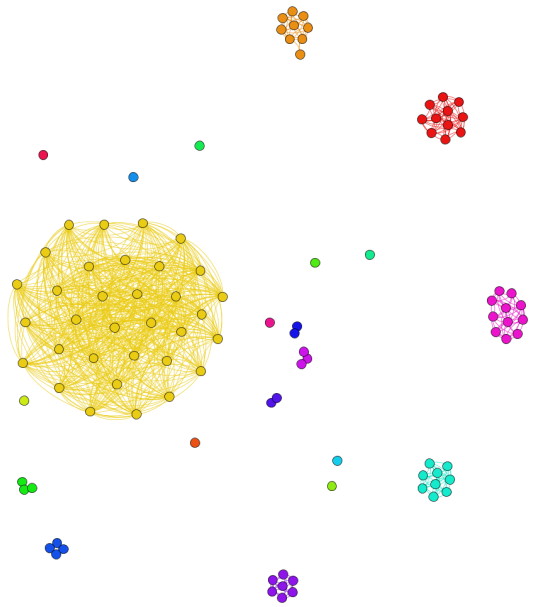

**Figure 5.19:** Locations identified with networks for userX during one day

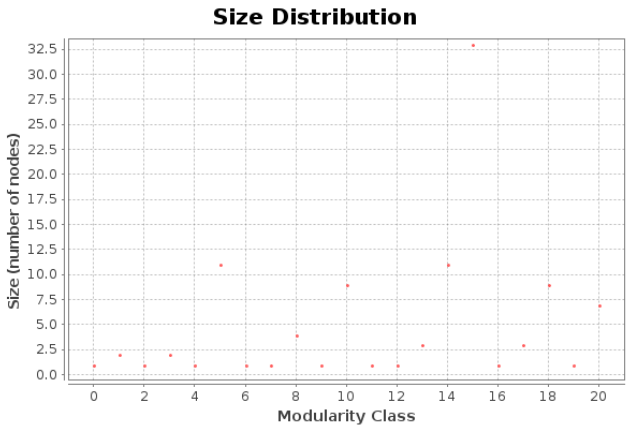


**Figure 5.20:** Size distribution of locations based on the number of APs associated to them

### 5.3.2 Cross validation

For both the k-means and the Hidden Markove Models approaches on extracting locations out of the user data we have been faced with a problem. The problem we have been faced with is that both these algorithm need to know how many locations they are trying to identify. However, we cannot know for sure, from the beginning, how many locations a user has been visited during a given time. In order for us to have a good estimation for the number of locations we could be expecting to find inside a time frame, we have used the cross validation technique, more specifically we have used the 10-fold cross validation method.

Cross validation [Koh95] is a technique for model validation that tries to asses how the results given by a statistical analysis of some given data can be generalized to an independent data set. The main use of cross validation is in problems that deal with prediction. Prediction problems usually deal with a set of training data and a set of testing data that the model needs to be able to react to as expected.

The k-fold cross validation divides the data we have at our disposal in k equal sized subsamples in a random way [11]. K-1 of the resulting subsamples are used as training data, while the remaining subsample is used as testing data. The samples are then rotated so as each of them becomes, in turn, testing data while the others for the training data. The k results are then combined in order to retrieve an unique estimation for the original data and an evaluation of the accuracy of the prediction model can be done based on how close the result is to the original data. The k value can be any number as long as the data can be divided into k subsamples. A value that is oftenly used for k is 10 [MDA05].

An example of how 2-fold cross validation works for four location fingerpints can be seen in Fig. 5.21.

In Step1 we have the four fingerprints. Let us consider that the algorithm that we are using to extract the locations based on these fingerprints have identified the location 1 and 2 as they can be seen in Step1. In order to see if our algorithm behaved as expected, we can cross validate the result using in this case a 2-fold cross validation. We are randomly selecting the 2 subsamples as is seen in Step2. The blue color is associated with the training data, while the orange color is associated with the testing data. In Step3 the first subsample is treated as training data while in Step4 the second one represents the training data. After the test data are classified based on the training data we can combined the results into one single sample which is presented in Step5. In this the cross

[11] Random in this case means that each of the subsamples contains elements from the original sample that are most likly not in their original order.

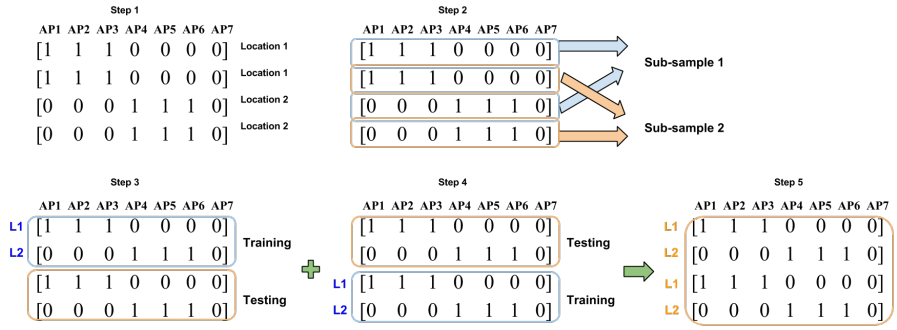


**Figure 5.21:** Example for 2-fold cross validation

validation has returned a result which matches the original estimation made by our selected algorithm, which means that the algorithm we have used has worked fine.

### 5.3.3 K-means clustering

The k-means algorithm is a popular method for analyzing clusters in data mining. The first time the "k-means" term was used was by MacQueen [Mac67], yet the standard algorithm for the k-means problem was proposed by Lloyd [Llo06]. The idea behind this algorithm is to start the clustering process by having k original groups of only one point each. After this initial setup, each new point can be added to the cluster that has the mean nearest to the new point. After a new point is added to a group, the mean of the group is updated, and at each stage the k means will represent the means of the k groups.

The Lloyd algorithm for k-means is the solution that is used for creating the k-means tool by scikit-learn [SL]. We have used the tool provided by scikit-learn in order to try to extract the places where a user has been situated at during a time frame based on the fingerprints that we can determine for that time frame. Our goals was to use the k-means algorithm to cluster the fingerprints that are similar enough as to be associated to the same location. However, since we cannot know for sure the number of locations (clusters) we are to expect for a given time frame, we are running the k-means algorithm with different values for k and we perform 10-fold cross validation in order to see what value has generate the most likely estimation.

The steps in extracting the locations with k-means and 10-fold cross validation

are as follow:

- We select the time frame (number of days) for which we want to extract the locations
- We retrieve the data and extract the fingerprints
- Since previous research shows that in general people spend most of their time in a small amount of locations (5 to 50) [MCG08], we choose the maximum number of locations we are expecting to find as the minimum value between 50 and the rezult for the number of days multiplied by $10^{12}$
- For each possible number for locations from between 2 and our previously selected maximum we run the k-means clustering algorithm on the identified fingerprints
- The estimations are cross validated in order to see which number of locations has generated the optimal approximation for the given fingerprints
- In case more locations have generated equally good results we selected as number of expected locations the highest of them
- This algorithm is ran 10 times leading to 10 estimations for the number of locations. Out of these estimations the one which appears the most times out of the 10 results is considered correct.

We ran the algorithm for 10 times in order to ensure that we the final result is as less influenced by the random fact involved in the determination of the subsamples for the cross validation as possible. At the end of the algorith we have the estimation for the locations throughout the selected time frame. An example for such an estimation can be seen in Fig. 5.23, while in Fig. 5.22 we have the presence of the APs which have been scanned throughout the same amount of time for the same user.

An additional example of locations estimated using k-means can be seen in Appendix A.2.4.

### 5.3.4 Hidden Markov Models

"Hidden Markov Models (HMMs) are a formal foundation for making probabilistic models of linear sequence 'labeling' problems. They provide a conceptual

[12] By observing the visualization for the presence of APs during different days in different users' life, we have observed that in general they seem to spend their time during a day in a most 10 locations

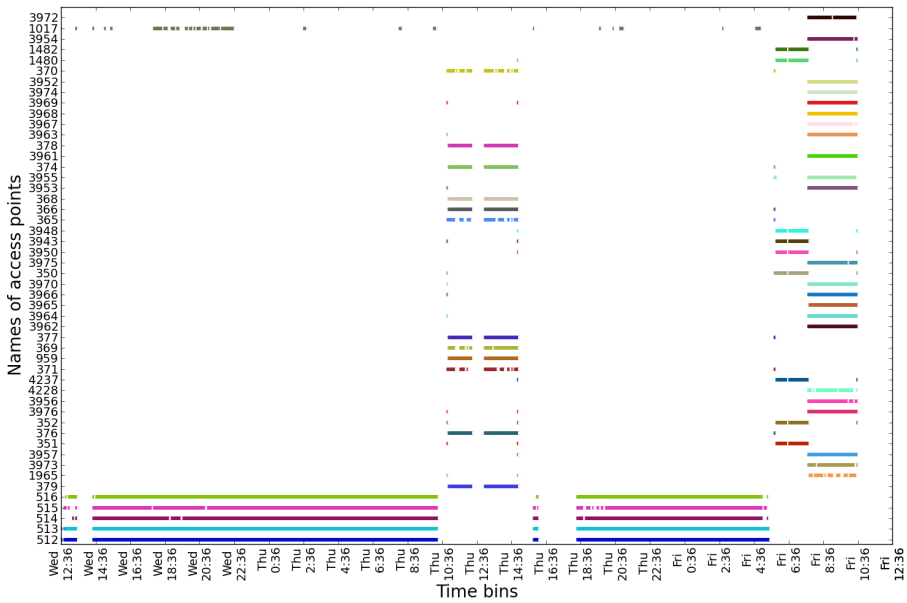


**Figure 5.22:** The most common 50 APs for userY during the given 2 days (presence visualization calculated for 5 minutes time bins)

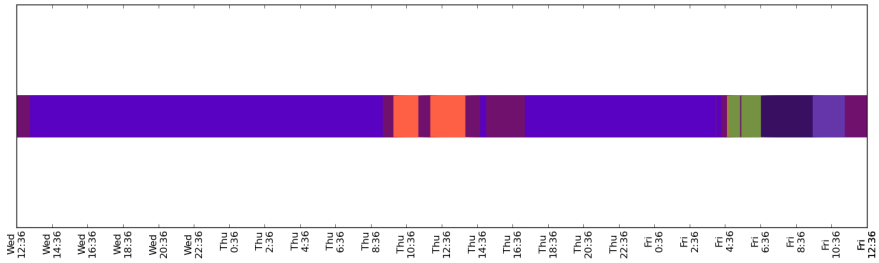

**Figure 5.23:** Locations estimated with k-means for userY for 2 days

toolkit for building complex models just by drawing an intuitive picture. They are at the heart of a diverse range of programs, including genefinding, profile searches, multiple sequence alignment and regulatory site identification. HMMs are the Legos of computational sequence analysis."[Edd04]

HMMs are, in principle, Markov Models (MMs) [Dra67] for which the modeled systems are considered to be processes with hidden states. If the the MMs the states are visible to possible observers, the difference for the HMMs is that the states are not visible, yet the results which can be observed do depend on the

hidden states [Rab89].

There area few elements that characterize the HMMs according to [Rab89]. They are as follows:

- N which represents the number of hidden states in the model. In general, these states can be interconnected in a way so that from some states others can be reached
- M which represents the number of different observation symbols which care generated by the hidden states
- A which represents the state transition probability distribution. A is a matrix for which each element $a_{i,j}$ represents the probability of moving from the state i to the state j in the system represented by the HMM
- $B = b_j(k)$ which represents the observation system probability distribution in state j. This basically means that each element in B shows the probability of seeing a particular element of M in a given state j
- $\pi$ which represents the initial state distribution, meaning what is the probability of the system to start producing output from any of the states in N

The three problems also mentioned in [Rab89] that the HMMs can be used for to solve are as follows:

- Given an observation sequence, how can the probability of the observation sequence be computed efficiently considering the given model
- Given the observation sequence how can a state sequence be chosen so that it explains in the most appropriate manner the existing observations
- How can the parameters of the model be adjusted in order to maximize the probability of a given observation sequence

The second of the three problems above addresses the uncovering of the hidden states of a given model. Which is exactly what we are trying to identify when attempting to extract what are the locations an user has been at based on observing the APs that have been scanned throughout a given time frame for the given user. In our case, N represents the number of unknown locations a user has been at, M is the set of observable fingerprints which we can calculate based on the presence of various APs in 5 minutes time bins, and A, B and $\pi$

are the various probability distributions that can be associated with the way in which the user travels from location to location.

The idea of using HMMs in order to track localization is not new. It has been explored in papers like [EKHH13], [IHO13] or [MNRS07] which sustain the potential of using an algorithm based on this method of studying the travel behaviour of people.

Scikit-learn [SL] offers an implementation for HMMs that ensures the training for the models and the inferring of the hidden states and we have been using the tools they provide for working with our data.

As with the k-means method (Section 5.3.3), the problem we have been facing was that we could not approximate from the beginning the number of hidden states (which stand for locations in our case) that we are expecting the model to find based on the input observations. However, by using the k-fold cross validation (Section 5.3.2) we can, once again (similar to the way in which we have solved the problem for k-means), test the estimation being computed based on different numbers of possible locations.

The steps in extracting the locations with the help of the HMM based algorithm that has been combined with a 10-fold cross validation are as follow:

- We select the time frame (number of days) for which we want to extract the locations
- We retrieve the data and extract the fingerprints
- We choose the maximum number of locations we are expecting to find in a similar manner we have done for the k-means algorithm, meaning as the minimum value between 50 and the result for the number of days multiplied by 10
- For each possible number of locations in the range of 2 and our previously selected maximum we run the HMM algorithm on the existing fingerprints
- The estimations are cross validated in order to see which number of locations has generated the optimal approximation for the given fingerprints
- In case more locations have generated equally good results we select as number of expected locations the highest of them
- This algorithm is also ran 10 times leading to 10 estimations out of which the one which appears the most times out of the 10 results is considered the correct one

The reason behind running the algorithm 10 times is the same as the one present for k-means algorithm. We want to ensure that the random factor which is involved in the cross validation process has a very little effect on the correctness of the estimation. At the end of the algorithm we have the hidden states (in our case, locations) that can be extracted based on the observations we have based on the presence of the various APs the user is associated to throughout the given time frame. An example of locations that have been found for userX throughout 3 days can be seen in Fig. 5.25. They can be easily mapped to the locations we can observe by looking at Fig. 5.24 where we have the presence of the APs which have been scanned throughout the same amount of time for the same user.

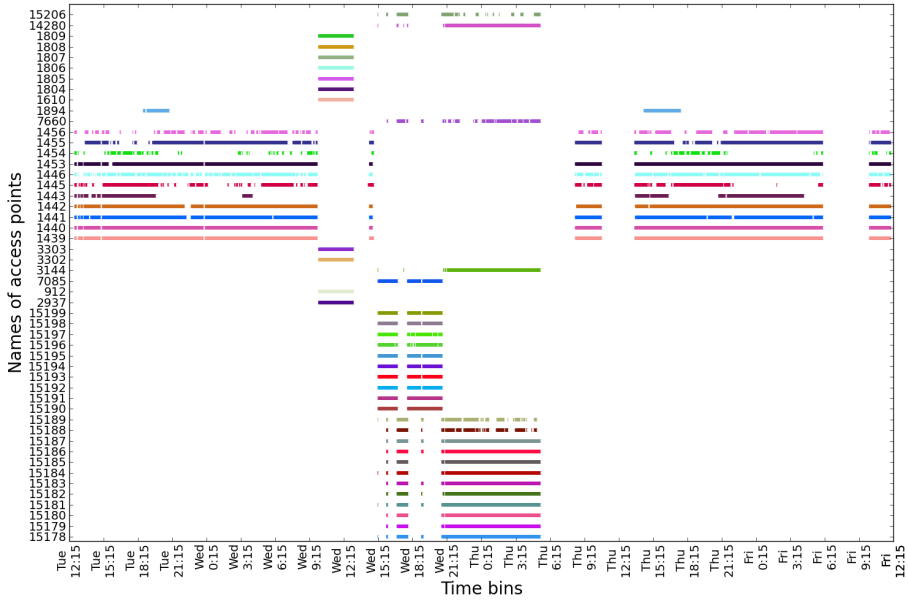


**Figure 5.24:** The most common 50 APs for userX during the given 3 days (presence visualization calculated for 5 minutes time bins)

An additional example of locations estimated using the HMM based algorithm can be seen in Appendix A.2.5.

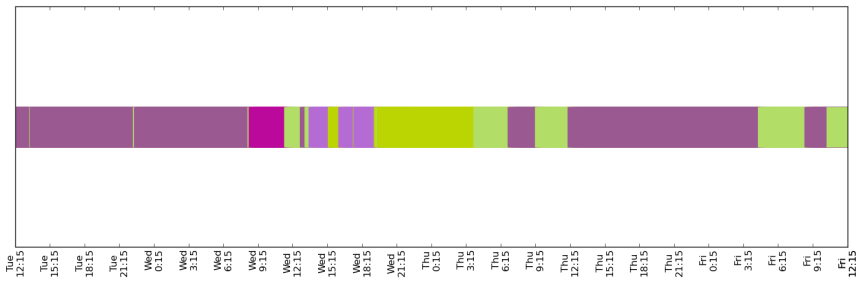


**Figure 5.25:** Locations estimated with HMM for userX for 3 days

Chapter 6

# Location matching through long periods of time

The HMM method as well as the k-means methods offer us with the possibility of extracting locations over a given number of days. However, both the algorithms perform better when the given time frame is shorter. This observation has come up while carefully observing the results for different number of days for which the algorithms have been executed.

The reason behind this behaviour seems to be the limitations of the 10-fold cross validation which has been used in order to evaluate the fitness of the results. The method implies that the data from the time frame taken into consideration is divided into 10 equal subsequences which are afterwards used in turns as training data and as testing data. However, when the data size grows, the randomly divided subsequences also grow. When we are dealing with subsequences which have a considerable size, it can happen that some of the subsequences can contain all of the fingerprints which can be attributed to a certain location and as such, that location cannot be estimated based on the other subsequences which have no knowledge of it. This leads to a decay in the efficiency of estimating the number of locations which we can expect the user has been at throughout the evaluated time.

A solution for this can be to scale the k factor of the k-fold validation in order to

use a factor larger than 10 when dealing with bigger amount of data, however this leads to a very long processing time which can be avoided by using the second possible solution. The second solution is to extract locations for each day and concatenate the results for all the days afterwards. This however leads to a new situation. We need to find a way in which to identify that a location $L_x$ from day X might be the same as a location $L_y$ from day Y. This problem is referred to in the present paper as the "matching of locations" problem.

We take into consideration three possible ways in which we can solve this new problem.

## 6.1 Methods for solving the "matching of locations" problem

We have taken into consideration three possible ways in which we can solve the problem of matching Wifi identified locations which seem to be indicating the same geographical location.

5.1. Identifying location fingerprint Ways to do this: a) Percentage similarity - for each location register all APs that ever are associated to it - keeping a dictionary with location name (e.g. 1) associated to the list of APs - before adding a new entry for a new location look at the previous ones and see how much they resembled it Problem: the APs of a location can be completely found in the fingerprint of another location and they don't need to be the same (e .g user 6 for the second day)

b) Keeping all ever registered fingerprints that were associated to a location and see if any of them can be found in another location as well (case in which we can say they are the same). In this case a fingerprint is a 1 and 0 list saying which APs are present which are not Problem : 2 locations can be the same even if there is no fingerprint that completely matches.

c) Create a overall fingerprint for a location with 1 and 0. An AP has a 1 of overall in the sub-fingerprints it was mostly present and 0 otherwise. The overall location can be compared with different thresholds for similarity. If similarity is above a threshold with another overall fingerprint for another location, than it can be assumed it is the same. * also, if a location is completely contained in another yet not above that threshold it is still the same (case in which 60the rest are missing in a location because they might have been switched off - in general locations do not share APs so much as for them to actually be kept in the overall location)

## 6.2 Location matching based on fingerprint similarity

After having the fingerprint for each location identified over a day, we needed to look at more time (e.g. 30 days) and try to match the locations up so that if the same location would appear throughout the days we would be able to pin point it and say it is the same one.

* Identifying locations with HMM over 30 days is very difficult because of the total number of APs considered (for user 6 over 2500 for example), and because the 10-fold validation over so much time means that a location has high chances of being randomly selected in the same fold and thus not being identified. Consider we have the fingerprints as dictionaries with AP1:0 or 1, AP2:0 or 1... and that we need to combine location A and B. Ways to calculate similarity between fingerprints: a) There's a A-B similarity calculated that means the number of APs in A that exist in B and have the same presence value attributed divided to the number of common APs between A and B. There is a similar B-A similarity. The overall similarity is calculated as the sum of the two previously mentioned values divided to 2.

Problem: ignores the number of APs that havn't the same value for A and B. They are not considered at all and this influences the result.

b) A reunion for bssids from both A and B is calculated. In case a bssid is not in A or B but it is in the reunion, than the presence value attributed for it is 0. In case a bssid is in A or B and the value for it in the original dictionary was 1, then it stays 1. The similarity is calculated as the number of bssids that have the same value for both the new dictionaries for A and B, divided to the number of bssids in either of these dictionaries (it's the same number since it's the reunion)

Problem: a lot of APs that are not originally in one of the locations are uselessly added to the location with a 0 value (even the ones that are 0 in the other location are added, even though they will not define any of the locations). This creates fake similarities (A, B might get extra similar locations based on an AP that wasn't even identified in B).

c) A reunion of only the bssids that are 1 in either a or b and the bssids that exist in both a and b (can also be 0) is calculated. The similarity is calculated as the number of bssids in this reunion that have the same presence associated for both a and b divided to the total number of these bssids (same in a and b).

Thresholds Ran for values between 0.98 and 0.65. In general around (80-75%) it stays to a constant estimation of locations, otherwise the jumps are quite big.

Chapter 7

# Entropy and predictability

## 7.1 Entropy of SensibleDTU users

## 7.2 Predictability of SensibleDTU users

Chapter 8

# Wifi versus GPS locations

## 8.1 Extracting stop locations from GPS data

## 8.2 Comparing results with GPS data

Chapter 9

# Results and observations

Chapter 10

# Future work

Chapter 11

# Conclusions

Appendix A

# Appendix

## A.1 Variations for signal strength visualization over time

This section contains various visualizations for different users' scanned access points over time. On the x axis we have the time frame, while on the y axis we have the signal strength for the identified access points. The legend presents only the top 10 predominant access points (which have appeared the most during scans), however the plot displays all access points. The figures are Fig. A.1, Fig.A.2, Fig. A.3, Fig.A.4.

## A.2 Sample density for APs identified for a user

This section contains the visualization for the signal strength of different APs that have been identified as being associated to a user throughout a period of 1 day (Fig. A.5) as well as the sample density visualizations for the top various APs that were scanned throughout this time (Fig. A.6,Fig. A.7,Fig. A.8).

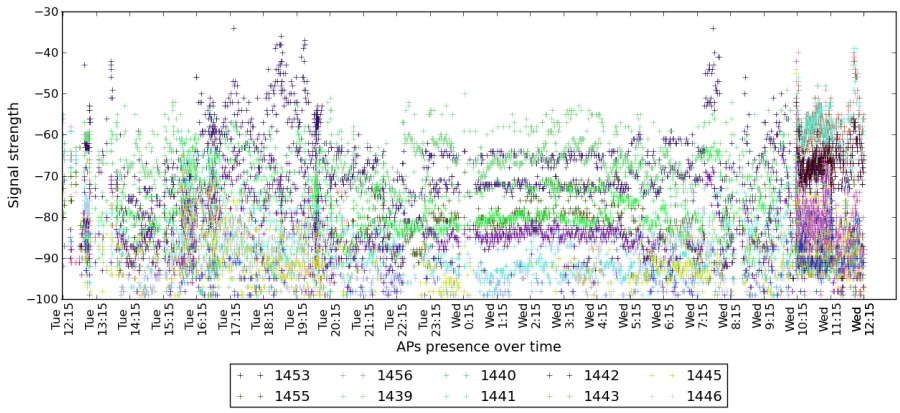


**Figure A.1:** Example of the APs registered for userX throughout one day with "+" markers

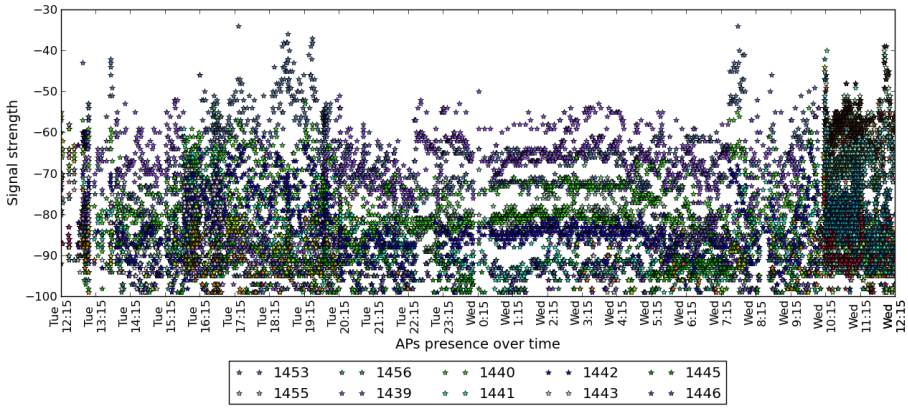


**Figure A.2:** Example of the APs registered for userX throughout one day with "*" markers

### A.2.1 Average signal strength for APs identified for a user

This section contains the visualization for the signal strength of APs 15188 (Fig. A.9), 15190 (Fig. A.10) and 3144 (Fig. A.11) calculated for 5 minutes time bins over the course of one day from the data gathered for userZ.

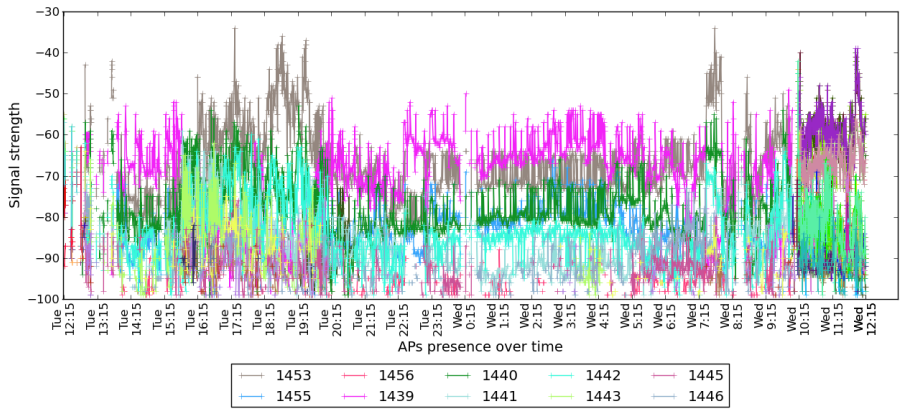


**Figure A.3:** Example of the APs registered for userX throughout one day with "+" and line markers

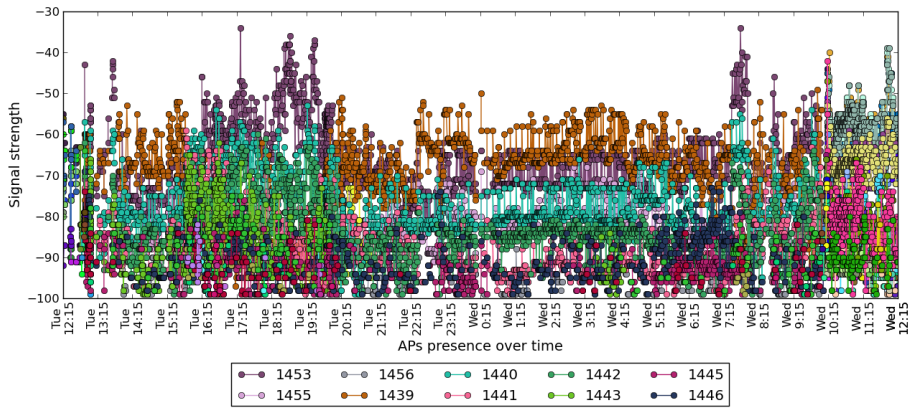


**Figure A.4:** Example of the APs registered for an user throughout one day with "o" and line markers

### A.2.2 Running average signal strength

This section contains the visualization for the running averages calculated for 2 (Fig. A.13), 5 (Fig. A.13) and 10 (Fig. A.13) minutes time bins for AP 1613 identified in a time frame of one day (Fig.??).

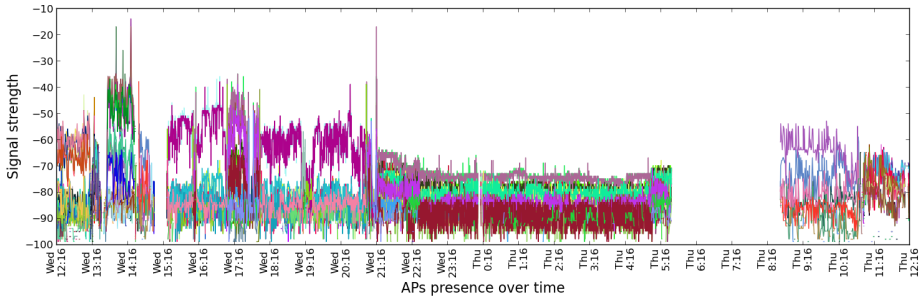


**Figure A.5:** Example of the APs registered for userX throughout day 2

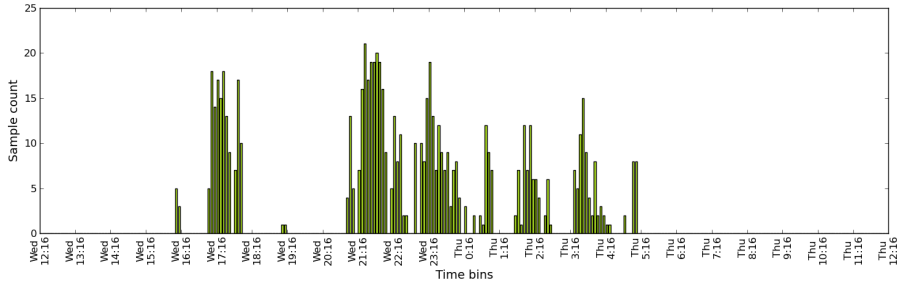


**Figure A.6:** Sample density of AP 15188 for userX

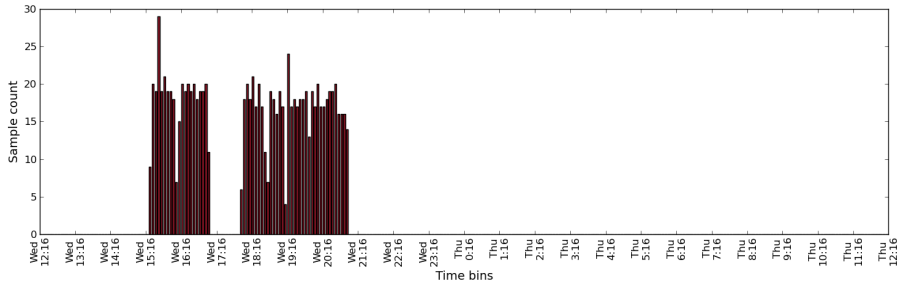


**Figure A.7:** Sample density of AP 15190 for userX

### A.2.3 Signal presence

This section contains the visualization for the presence of APs for a period of 2 days for an user from the SensibleDTU database (Fig. A.17).The presence for APs is determined for 5 minutes time bins over the 2 days. Fig. A.16 presents

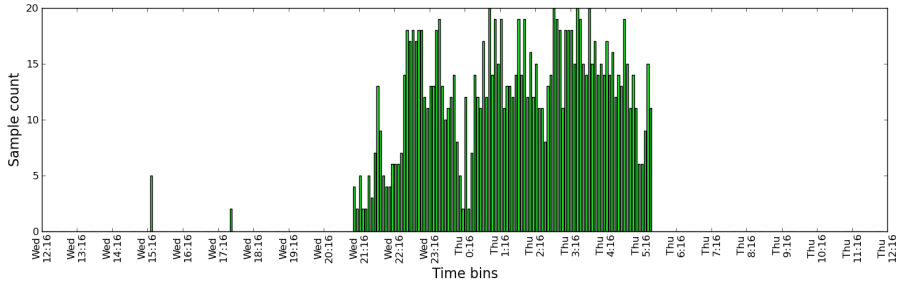


**Figure A.8:** Sample density of AP 3144 for userX

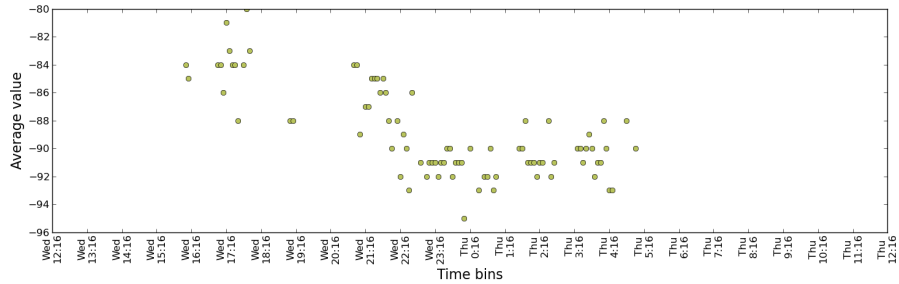


**Figure A.9:** Sample density of AP 15188 for userZ

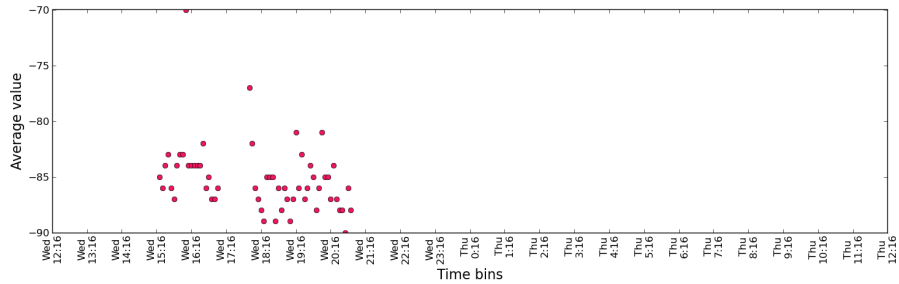


**Figure A.10:** Sample density of AP 15190 for userZ

all the APs (and their signals) visualized for the same 2 days.

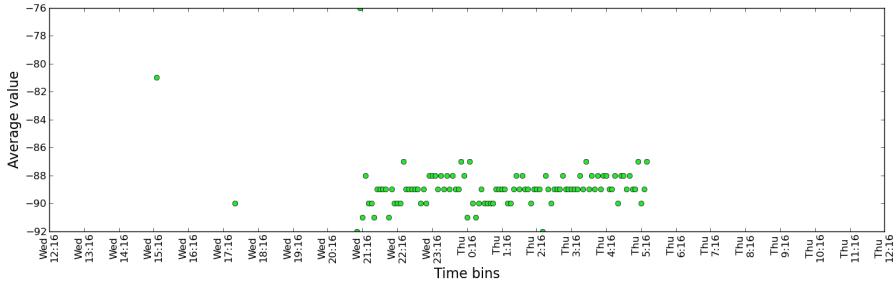


**Figure A.11:** Sample density of AP 3144

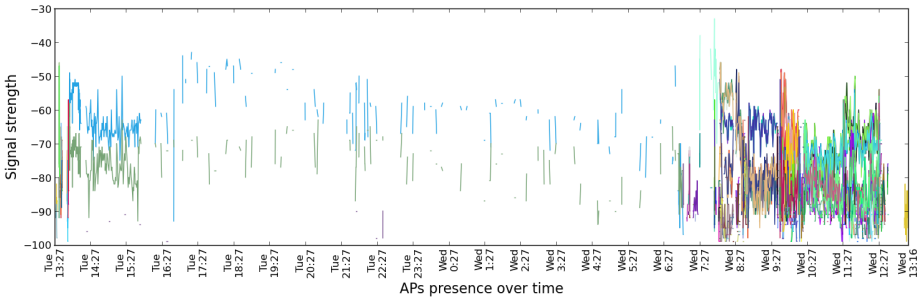


**Figure A.12:** Example of APs presence over time for userT

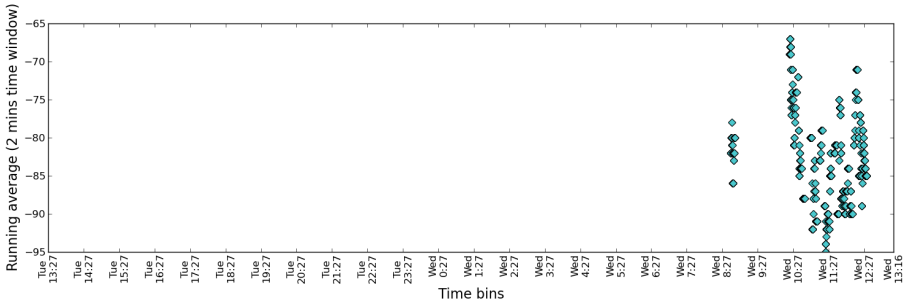


**Figure A.13:** Running average for AP 1613 for userT during 1 day (2 minute time bins)

### A.2.4 Locations extracted using k-means

### A.2.5 Locations extracted using HMM

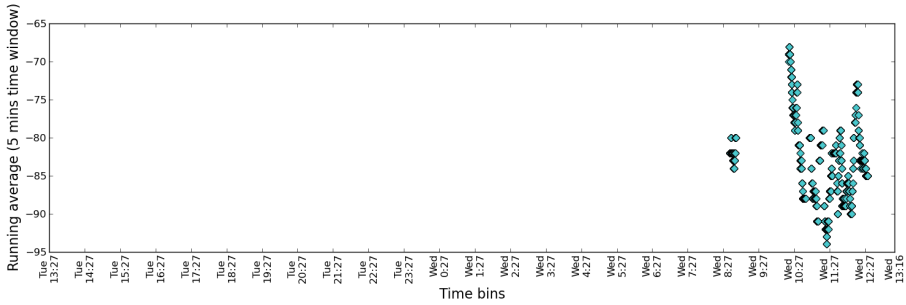


**Figure A.14:** Running average for AP 1613 for userT during 1 day (5 minute time bins)

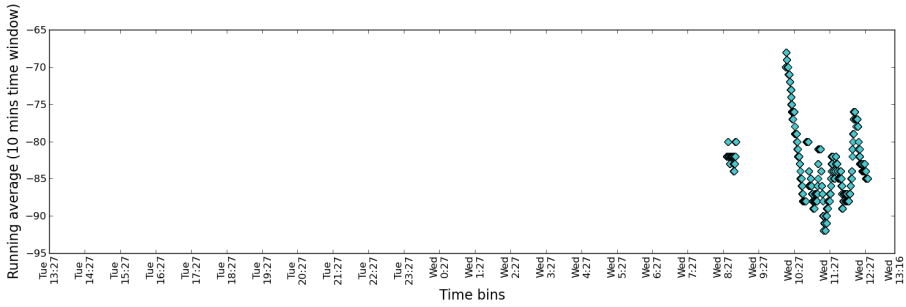


**Figure A.15:** Running average for AP 1613 for userT during 1 day (10 minute time bins)

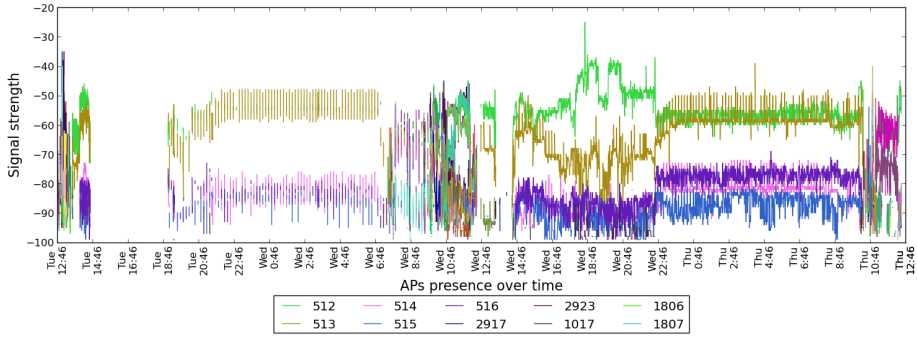


**Figure A.16:** Scanned APs for an user throughout a duration of 2 days

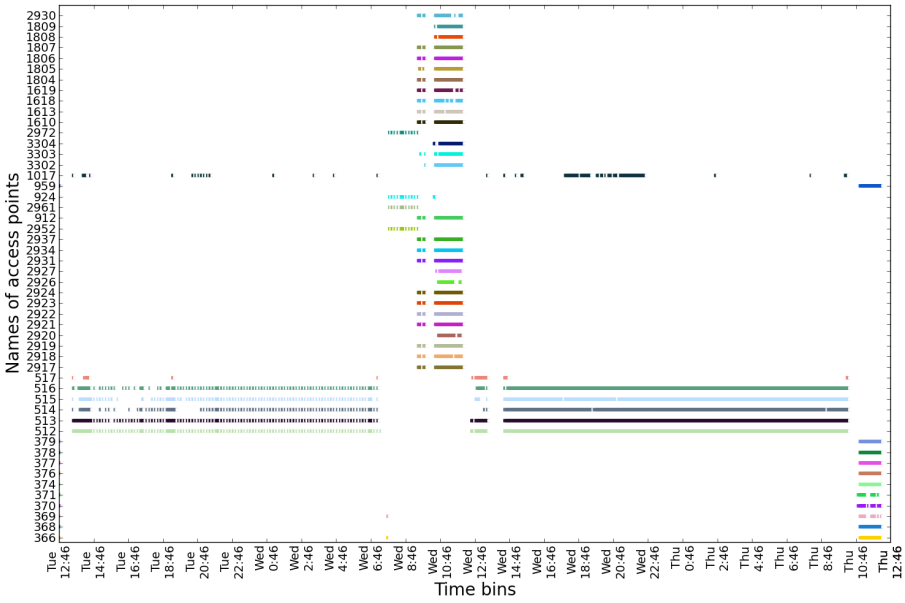


**Figure A.17:** The most common 50 APs for an user during 2 days (presence visualization calculated for 5 minutes time bins)

# Bibliography


[AGB13] Fereshteh Asgari, Vincent Gauthier, and Monique Becker. A survey on human mobility and its applications. *CoRR*, abs/1307.0814, 2013.

[AGGP09] Spiros Athanasiou, Panos Georgantas, George Gerakakis, and Dieter Pfoser. Utilizing wireless positioning as a tracking data source. In *Advances in Spatial and Temporal Databases*, pages 171–188. Springer, 2009.

[AS07] Sherif Akoush and Ahmed Sameh. Mobile user movement prediction using bayesian learning for neural networks. In *Proceedings of the 2007 international conference on Wireless communications and mobile computing*, pages 191–196. ACM, 2007.

[AS14a] Alex Pentland David Lazer Sune Lehmann Arkadiusz Stopczynski, Riccardo Pietri. Privacy in sensor-driven human data collection: A guide for practitioners. *CoRR*, abs/1403.5299, 2014.

[AS14b] Piotr Sapiezynski Andrea Cuttone Mette My Madsen Jakob Eg Larsen Sune Lehmann Arkadiusz Stopczynski, Vedran Sekara. Measuring large-scale social networks with high resolution. *CoRR*, abs/1401.7233, 2014.

[Bal03] P. Ball. The physical modelling of human social systems, 2003.

[BP00] Paramvir Bahl and Venkata N Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer*

*and Communications Societies. Proceedings. IEEE*, volume 2, pages 775–784. IEEE, 2000.

[CCLK05] Yu-Chung Cheng, Yatin Chawathe, Anthony LaMarca, and John Krumm. Accuracy characterization for metropolitan-scale wi-fi localization. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, pages 233–245. ACM, 2005.

[CJK05] Jernej Copic, Matthew O. Jackson, and Alan Kirman. Identifying community structures from network data via maximum likelihood methods, 2005.

[CLL14] Andrea Cuttone, Sune Lehmann, and Jakob Eg Larsen. Inferring human mobility from sparse low accuracy mobile sensing data. In *3rd ACM Workshop on Mobile Systems for Computational Social Science(MCSS 2014)*. ACM, 2014.

[CML11] Eunjoon Cho, Seth A Myers, and Jure Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[CS10] 1 2 3 Nicholas Blumm 1 2 Albert-László Barabási 1 2 * Chaoming Song, Zehui Qu. Limits of predictability in human mobility. *Science*, 327, 2010.

[CSC+06] Mike Y. Chen, Timothy Sohn, Dmitri Chmelev, Dirk Haehnel, Jeffrey Hightower, Jeff Hughes, Anthony LaMarca, Fred Potter, Ian Smith, and Alex Varshavsky. Practical metropolitan-scale positioning for gsm phones. In *Proceedings of the 8th International Conference on Ubiquitous Computing*, UbiComp'06, pages 225–242. Springer-Verlag, 2006.

[DB06] T. Geisel D. Brockmann, L. Hufnagel. The scaling laws of human travel. *Nature*, 439, 2006.

[DB08] F. Theis D. Brockmann. Money circulation, trackable items, and the emergence of universal human mobility patterns. *Pervasive Computing, IEEE*, 7, 2008.

[Dra67] Alvin W. Drake. Fundamentals of applied probability theory, 1967.

[Edd04] Sean R Eddy. What is a hidden markov model? *Nat Biotech*, 22(10), 2004.

[EKHH13] Reda A El-Khoribi, Haitham S Hamza, and MA Hammad. Indoor localization and tracking using posterior state distribution of hidden markov model. In *Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on*, pages 557–562. IEEE, 2013.

[FBSW08] Bo Fu, Gábor Bernath, Ben Steichen, and Stefan Weber. Wireless background noise in the wi-fi spectrum. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*, pages 1–7. IEEE, 2008.

[Fli03] Rob Flickenger. *Building Wireless Community Community Networks*. O'Reilly & Associates, Inc., Sebastopol, CA, USA, 2 edition, 2003.

[GCP+06] Y. Guo, P. Corke, G. Poulton, T. Wark, G. Bishop-Hurley, and D. Swain. Animal behaviour understanding using wireless sensor networks. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*, pages 607–614, Nov 2006.

[Gep] Gephi - the open graph viz platform.

[GL96] G. Maguire Jr G. Liu. A class of mobile motion prediction algorithms for wireless mobile computing and communication. *Mobile Networks and Applications*, 1, 1996.

[Hyn09] Rob J Hyndman. Moving averages, 2009.

[IHO13] Yusuke Inatomi, Jihoon Hong, and Tomoaki Ohtsuki. Hidden markov model based localization using array antenna. *International journal of wireless information networks*, 20(4):246–255, 2013.

[JP04] W. Trumler T. Ungerer L. Vintan J. Petzold, F. Bagci. Global state context prediction techniques applied to a smart office building, 2004.

[JS99] Oliver P John and Sanjay Srivastava. The big five trait taxonomy: History, measurement, and theoretical perspectives. *Handbook of personality: Theory and research*, 2(1999):102–138, 1999.

[Kal60] R. E. Kalman. A new approach to linear filtering and prediction problems, 1960.

[KC11] Jahyoung Koo and Hojung Cha. Autonomous construction of a wifi access point map using multidimensional scaling. In *Pervasive Computing*, pages 115–132. Springer, 2011.

[Koh95] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. pages 1137–1143. Morgan Kaufmann, 1995.

[LBG80] Yoseph Linde, Andres Buzo, and Robert M Gray. An algorithm for vector quantizer design. *Communications, IEEE Transactions on*, 28(1):84–95, 1980.

[LJ09] Jakob Eg Larsen and Kristian Jensen. Mobile context toolbox: An extensible context framework for s60 mobile phones. In *Proceedings of the 4th European Conference on Smart Sensing and Context*, EuroSSC'09, pages 193–206. Springer-Verlag, 2009.

[Llo06] S. Lloyd. Least squares quantization in pcm. *IEEE Trans. Inf. Theor.*, 28(2):129–137, September 2006.

[Mac67] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[MCG08] A. L. Barabasi M. C. Gonzalez, C. A. Hidalgo. Understanding individual human mobility patterns. *Nature*, 453, 2008.

[MCWA10] Aniket Mahanti, Niklas Carlsson, Carey L. Williamson, and Martin F. Arlitt. Ambient interference effects in wi-fi networks. In *Networking*, volume 6091 of *Lecture Notes in Computer Science*, pages 160–173. Springer, 2010.

[MDA05] Geoffrey McLachlan, Kim-Anh Do, and Christophe Ambroise. *Analyzing microarray gene expression data*, volume 422. John Wiley & Sons, 2005.

[MK06] S. Kim M. Kim, D. Kotz. Extracting a mobility model from real user traces. *The IEEE INFOCOM Proceedings*, 2006.

[MM07] Mirco Musolesi and Cecilia Mascolo. Designing mobility models based on social network theory. *SIGMOBILE Mob. Comput. Commun. Rev.*, 11(3):59–70, 2007.

[MNRS07] Carlo Morelli, Monica Nicoli, Vittorio Rampa, and Umberto Spagnolini. Hidden markov models for radio localization in mixed los/nlos conditions. *Signal Processing, IEEE Transactions on*, 55(4):1525–1542, 2007.

[MR07] E Mok and Günther Retscher. Location determination using wifi fingerprinting versus wifi trilateration. *Journal of Location Based Services*, 1(2):145–159, 2007.

[NE09] A. S. Pentland N. Eagle. Eigenbehaviors: identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63, 2009.

[Par] Pardus game.

[PSL] The python standard library.

[Rab89] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, Feb 1989.

[RCC+04] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 101(9):2658–2663, 2004.

[Ros09] S. M. Ross. *Introduction to probability models.* Academic Press, 2009.

[RS14] M. Szell R. Sinatra. Entropy and the predictability of online life, 2014.

[SCL03] H. C. Lu S. C. Liou. Applied neural network for location prediction and resources reservation scheme in wireless networks. *International Conference on Communication Technology Proceedings*, 2, 2003.

[SL] Scikit-learn.

[TSA09] C. A. V. Campos L. F. M. de Moraes T. S. Azevedo, R. L. Bezerra. An analysis of human mobility using real traces, 2009.

[WLP] Ieee 802.11 - wireless lans.

[XL13] N. Bharti A. J. Tatem L. Bengtsson X. Lu, E. Wetter. Approaching the limit of predictability in human mobility, 2013.

[YA05] Moustafa Youssef and Ashok Agrawala. The horus wlan location determination system. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services*, MobiSys '05, pages 205–218. ACM, 2005.

[YYZS10] Shunsen Yang, Xinyu Yang, Chao Zhang, and Evangelos Spyrou. Using social network theory for modeling human mobility. *Network, IEEE*, 24(5):6–13, 2010.

[ZF12] Nan Zhang and Jianhua Feng. Polaris: A fingerprint-based localization system over wireless networks. In *Web-Age Information Management*, pages 58–70. Springer, 2012.

[ZG10] M. Zignani and S. Gaito. Extracting human mobility patterns from gps-based traces. In *Wireless Days (WD), 2010 IFIP*, pages 1–5, Oct 2010.