



university of  
groningen

faculty of mathematics  
and natural sciences

SOCIAL DENSITY ESTIMATION BASED ON CONSUMER  
SMARTPHONE SENSORS

GUNTUR DHARMA PUTRA

Master's in Computing Science  
Faculty of Mathematics and Natural Sciences  
University of Groningen

January 2017

Guntur Dharma Putra: *Social Density Estimation Based on Consumer Smartphone Sensors*, Master's in Computing Science, © January 2017

**SUPERVISORS:**

Prof. Dr. Marco Aiello  
Prof. Dr. Martien Kas  
Niels Jongen MSc

**LOCATION:**

Groningen

To my beloved family.  
Bapak, Ibu, Dik Putri, and Dik Dhara.



## ABSTRACT

---

Recent developments in smartphone technologies raise the concept of mobile healthcare systems as an essential part of medical care or research processes. As opposed to the conventional techniques which are prone to human errors and biased results, smartphone based monitoring systems can provide objective results especially when dealing with longitudinal assessment of individual movement patterns in the context of social density.

In this thesis, we present a consumer smartphone based social density estimation method that estimates the number of people in a certain area by utilizing smartphone sensors. We use WiFi to count nearby Access Points (AP) and microphones to record ambient noise of the surroundings. We performed data collection in several locations, ranging from low to high level social densities, using WiFi MAC address counting and time-lapse images as the ground truth approximation.

The results indicate that smartphones have good potential for estimating social density levels. The results reveal that the AP and ambient noise have a positive correlation with the social density level, which in our experiments is 0.8 and 0.6, respectively. Furthermore, we also constructed prediction models for the social density level using new data with residual error is equal to 7.05.



## ACKNOWLEDGMENTS

---

First and foremost, I would like to express my deepest gratitude to God, the Almighty, for having made everything possible and giving me the strength and courage to finish this work.

I would also like to express my sincere gratitude to both of my supervisors Prof. Marco Aiello and Prof. Martien Kas for the hard work, agile support, and boundless knowledge for my thesis. Moreover, I would also like to thank my daily supervisor, Niels Jongs, who has been guiding me throughly until I can finish this thesis.

Last but not the least, I would also like to say thank you to my beloved family: my parents and my sisters for supporting me spiritually throughout this thesis and my study in general. Also, my friends and colleagues in Groningen who have helped me to work on this thesis.



## CONTENTS

---

<b>I</b>	<b>MASTER'S THESIS</b>	<b>1</b>
<b>1</b>	<b>INTRODUCTION</b>	<b>3</b>
<b>1.1</b>	Research Questions . . . . .	4
<b>1.2</b>	Method and Approach . . . . .	5
<b>1.3</b>	Thesis Structure Overview . . . . .	5
<b>2</b>	<b>STATE OF THE ART</b>	<b>7</b>
<b>2.1</b>	Crowd Counting Techniques . . . . .	7
<b>2.1.1</b>	WiFi-based Techniques . . . . .	7
<b>2.1.2</b>	Bluetooth-based Techniques . . . . .	11
<b>2.1.3</b>	Other RF-based Techniques . . . . .	12
<b>2.1.4</b>	Other Crowd Counting Techniques . . . . .	13
<b>2.2</b>	Ambulatory Assessment . . . . .	14
<b>2.3</b>	Discussion . . . . .	15
<b>3</b>	<b>EXPERIMENTAL SETUP</b>	<b>19</b>
<b>3.1</b>	Experimental Design . . . . .	19
<b>3.2</b>	Implementation . . . . .	20
<b>3.2.1</b>	MAC Address Randomization Investigation . . . . .	20
<b>3.2.2</b>	Ground Truth Estimation . . . . .	21
<b>3.2.3</b>	Smartphone Sensor Utilization . . . . .	23
<b>3.2.4</b>	Location and Timing . . . . .	24
<b>3.2.5</b>	Scanning Mechanism . . . . .	25
<b>3.3</b>	Data Extraction and Evaluation . . . . .	26
<b>3.3.1</b>	WiFi Raw Data Extraction . . . . .	27
<b>3.3.2</b>	Recorded Ambient Noise Extraction . . . . .	27
<b>3.3.3</b>	Manual Head Counting . . . . .	28
<b>3.3.4</b>	Metrics and Evaluation . . . . .	28
<b>4</b>	<b>RESULTS</b>	<b>29</b>
<b>4.1</b>	MAC Address Randomization . . . . .	29
<b>4.1.1</b>	iPad mini . . . . .	30
<b>4.1.2</b>	Nexus 5X . . . . .	31
<b>4.1.3</b>	Review . . . . .	32
<b>4.2</b>	Correlation between Crowd Count and Sensor Readings	32
<b>4.2.1</b>	Ambient Noise Recordings . . . . .	33
<b>4.2.2</b>	AP and Social Density Correlation . . . . .	35
<b>4.2.3</b>	Effect of Scanning Time . . . . .	37
<b>4.2.4</b>	All Result . . . . .	39
<b>4.3</b>	Social Density Prediction . . . . .	40
<b>4.3.1</b>	Linear Regression . . . . .	41
<b>4.3.2</b>	k-Nearest Neighbor . . . . .	43
<b>4.3.3</b>	Support Vector Machine . . . . .	43
<b>5</b>	<b>DISCUSSION</b>	<b>47</b>

5.1	Ambient Noise Recording . . . . .	47
5.2	Ground Truth Approximation . . . . .	48
5.3	Scanning Time Effect . . . . .	48
5.4	Correlation Between Sensor Readings and Social Density	49
5.5	Social Density Prediction . . . . .	50
5.6	Limitations of the Present Study . . . . .	50
6	CONCLUSION AND FUTURE WORK	53
6.1	Future Work . . . . .	54
 II APPENDIX		55
A	ACCESS POINT COUNT CORRELATION	57
A.1	Scatter Plots . . . . .	57
A.2	Line charts . . . . .	60
B	TIME-LAPSE IMAGES	65
B.1	Remote Area . . . . .	65
B.2	Home . . . . .	66
B.3	Paddepoel . . . . .	67
B.4	Grote Markt . . . . .	69
C	AMBIENT NOISE RECORDINGS	71
C.1	Day 1 . . . . .	71
C.2	Day 2 . . . . .	72
C.3	Day 3 . . . . .	73
D	R CODE LISTINGS	75
 BIBLIOGRAPHY		77

## LIST OF FIGURES

---

Figure 1	GoPro arrangement . . . . .	22
Figure 2	Sensor measurement in each cycle. . . . .	25
Figure 3	Example of Access Point ( <a href="#">AP</a> ) list. . . . .	27
Figure 4	A fragment of captured probe request packet.	27
Figure 5	A box plot of the population . . . . .	33
Figure 6	Line chart of ambient noise at day 4 . . . . .	34
Figure 7	Scatter plot of device and <a href="#">AP</a> count. . . . .	36
Figure 8	Scatter plot of head and <a href="#">AP</a> count. . . . .	36
Figure 9	Scatter plot of head and device count. . . . .	37
Figure 10	Line charts of ambient noise on October 30, 2016.	38
Figure 11	Ambient noise in different scanning time. . . . .	39
Figure 12	Scatter plot matrix of all parameters. . . . .	40
Figure 13	Tuning linear regression. . . . .	42
Figure 14	Tuning k-Nearest Neighbors ( <a href="#">k-NN</a> ). . . . .	43
Figure 15	Tuning Support Vector Machine ( <a href="#">SVM</a> ). . . . .	44
Figure 16	The scatter plots of the correlation between head-count and <a href="#">AP</a> count. . . . .	57
Figure 17	The scatter plots of the correlation between device count and <a href="#">AP</a> . . . . .	58
Figure 18	The scatter plots of the correlation between device count and <a href="#">AP</a> count. . . . .	59
Figure 19	The line chart of sensor readings at remote area.	60
Figure 20	The line chart of sensor readings at home. . . . .	61
Figure 21	The line chart of sensor readings at Paddepoel Shopping Center. . . . .	62
Figure 22	The line chart of sensor readings at Grote Markt.	63
Figure 23	Example of time-lapse images taken at remote area in day 1. . . . .	65
Figure 24	Example of time-lapse images taken at remote area in day 2. . . . .	65
Figure 25	Example of time-lapse images taken at remote area in day 3. . . . .	65
Figure 26	Example of time-lapse images taken at remote area in day 4. . . . .	66
Figure 27	Example of time-lapse images taken at home in day 1. . . . .	66
Figure 28	Example of time-lapse images taken at home in day 2. . . . .	66
Figure 29	Example of time-lapse images taken at home in day 3. . . . .	67
Figure 30	Example of time-lapse images taken at home in day 4. . . . .	67

Figure 31	Example of time-lapse images taken at <i>Paddepoel shopping center</i> in day 1. . . . .	67
Figure 32	Example of time-lapse images taken at <i>Paddepoel shopping center</i> in day 2. . . . .	68
Figure 33	Example of time-lapse images taken at <i>Paddepoel shopping center</i> in day 3. . . . .	68
Figure 34	Example of time-lapse images taken at <i>Paddepoel shopping center</i> in day 4. . . . .	68
Figure 35	Example of time-lapse images taken at <i>Grote markt</i> in day 1. . . . .	69
Figure 36	Example of time-lapse images taken at <i>Grote markt</i> in day 2. . . . .	69
Figure 37	Example of time-lapse images taken at <i>Grote markt</i> in day 3. . . . .	69
Figure 38	Example of time-lapse images taken at <i>Grote markt</i> in day 4. . . . .	70
Figure 39	Line chart showing the peak level (39a) and root-mean-square (39b) in each cycle at day 1.	71
Figure 40	Line chart showing the peak level (40a) and root-mean-square (40b) in each cycle at day 2.	72
Figure 41	Line chart showing the peak level (41a) and root-mean-square (41b) in each cycle at day 3.	73

## LIST OF TABLES

---

Table 1	Literature review summary. . . . .	16
Table 2	Summary of typical smartphone sensors . . . . .	17
Table 3	Summary of location selection . . . . .	24
Table 4	Smartphone sensor readings and the extracted parameters. . . . .	26
Table 5	Summarized MAC address randomization behavior. . . . .	29
Table 6	Some examples of captured probe requests from iPad mini. . . . .	30
Table 7	An example of captured probe requests from Nexus 5X. . . . .	31
Table 9	The average of RMS and PKLV of ambient noise in four scanning time. . . . .	39
Table 10	Linear model fit to the dataset . . . . .	42
Table 11	Summary of regression analysis . . . . .	45
Table 12	The average of root-mean-square and peak level of ambient noise recording in day 1. . . . .	71
Table 13	The average of root-mean-square and peak level of ambient noise recording in day 2. . . . .	72
Table 14	The average of root-mean-square and peak level of ambient noise recording in day 3. . . . .	73
Table 15	Summary of R libraries in regression analysis	75

## LISTINGS

---

Listing 1	Tuning linear regression with stepwise selection. Number of predictor <code>nvmax</code> is the tuning parameter. . . . .	75
Listing 2	Tuning k-Nearest Neighbors regression. Number of neighbors ( <code>k</code> ) is the tuning parameter. . . . .	75
Listing 3	Tuning Support Vector Machine regression. Epsilon ( $\epsilon$ ) and <code>cost</code> are the tuning parameters. . . . .	76

## ACRONYMS

---

AP	Access Point
BSSID	Basic Service Set Identifier
CSI	Channel State Information
DC	Device Count
ESM	Experience Sampling Method
FFT	Fast Fourier Transfer
FOV	Field of View
GPS	Global Positioning System
HC	Head Count
k-NN	k-Nearest Neighbors
LQI	Link Quality Indicator
MAC	Media Access Control
MFCC	Mel-Frequency Cepstral Coefficient
MSE	Mean Squared Error
NFC	Near Field Communication
OS	Operating System
PDR	Pedestrian Dead Reckoning
PKLV	Peak Level
PNL	Preferred Network List
PPMCC	Pearson Product Moment Correlation Coefficient
RF	Radio Frequency
RFID	Radio-Frequency Identification
RMS	Root Mean Square
RMSE	Root Mean Square Error
ROM	Read Only Memory
RSSI	Received Signal Strength Indicator

SA	Source Address
SC	Speaker Count
SIM	Subscriber Identity Module
SN	Sequence Number
SNR	Signal to Noise Ratio
SSID	Service Set Identifier
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle
WSN	Wireless Sensor Network

**Part I**  
**MASTER'S THESIS**



## INTRODUCTION

---

People who suffer from neuropsychiatric disorders usually withdraw from society [9, 22]. They often visit places that tend to be quite and tend to have less crowd. The current method to monitor neuropsychiatric patients involve oral questionnaires that record a patient's social interaction on a daily basis. However, the questionnaire method often leads to biased and subjective results, making an objective monitoring is needed to obtain reliable results. The objective monitoring of social withdrawal requires a longitudinal assessment of individual movement patterns in the context of social density. This is where *social density estimation* comes into play.

Social density estimation, or sometimes referred as crowd counting, is the mechanism of estimating the number of people (social density) in a certain area by using automated methods, which could replace the manual counting technique, e.g., counting heads by using tally counter. Besides its implementation in monitoring neuropsychiatric patients, social density estimation has many potential implementations, for instance, crowd surveillance [27], evacuation and rescue [3], retail store customer analysis, infrastructure development and evaluation, and queue management [40].

Several approaches of social density estimation exist. Surveillance camera utilization is one of the methods used, although it is limited by high deployment and computational cost. As opposed to video-based techniques, some researches have proposed Radio Frequency (RF) signals, e.g., WiFi or Bluetooth, to get lower deployment and computational cost. The WiFi and Bluetooth based methods make use of several characteristics present in WiFi or Bluetooth, such as probe-request, Media Access Control (MAC) address monitoring [42], Received Signal Strength Indicator (RSSI) [12], and Channel State Information (CSI) [13]. Furthermore, some of the WiFi and Bluetooth based methods are based on an assumption that says a unique MAC address refers to a different individual.

However, to the best of our knowledge, the available social density estimation techniques require pre-installed dedicated infrastructures that make the techniques not implementable for patient monitoring and for longitudinal and objective behavioral monitoring of humans in general. The preferred monitoring technique requires device that can run almost everywhere to monitor patients condition.

Speaking of patient monitoring, some researchers have proposed smartphone-based technique to gain flexibility of implementation and monitoring, as proposed by Bachmann [6] and Eskes, et al. [14]. The

method aims to achieve passive behavioral monitoring that works seamlessly under the hood, so that patients do not know that they are monitored. The proposed method monitors almost every aspects that are required to monitor depressive patients or patients with neuropsychiatric disorders.

Although the proposed methods collect patients daily activity, such as social communication in messaging applications and daily phone calls, the methods do not address social density estimation explicitly. Further study is required to investigate whether the collected data can give information about the level of social density that a patient encounters.

### 1.1 RESEARCH QUESTIONS

The present study investigates the possible combination of sensory data collected by smartphone sensors or network interfaces to estimate the level of social density in the surroundings. This leads to the main research question:

*How can we estimate the level of social density of the surroundings using smartphone as a passive behavioral monitoring scheme?*

In order to answer and validate the main research question the following sub-questions are formulated:

1. *What sensors are available in consumer smartphones and which of those are useful for the social density estimation method?*

Modern smartphones are equipped with several sensors to help users in their works, e.g., cameras, proximity sensor, finger print sensor, etc. In spite of the sensors actual functionality, some sensors must be capable of giving such information for social density estimation. We are interested in investigating the sensors which are useful to achieve social density estimation.

2. *How can we validate the sensor readings, meaning, to get the ground truth or an approximation of the ground truth?*

To validate whether the smartphone sensor data does something related to the social density, we have to compare the smartphone sensor data with the actual social density level, or the ground truth. We look a method to obtain such ground truth or at least a strong estimation of the ground truth.

3. *Can this method work everywhere? What is the scope of this method?*

We aim to devise a method that would work almost anywhere. Although this may seem too ambitious, there must be a certain limit of the present study. We formally portray the scope of the final result.

## 1.2 METHOD AND APPROACH

To answer the research questions, we first start with a literature study to discover the state of the art of the objective social density estimation techniques. Moreover, we discover potential sensors and network interfaces, which are useful for social density estimation. We also look for possible ground truth estimation as an approximation of real condition of the surrounding.

Then, we collect data using the preferred sensors and network interfaces in a certain timing and location to infer the trends of the correlation between sensor readings and the actual social density level. We plan to perform the data collection for several days in several locations, ranging from low to high level of social density.

Lastly, we investigate the results to see whether a correlation between smartphone sensor readings and social density levels exists. If there is a considerably strong correlation, then we develop data models to predict the level of social density based on smartphone sensor readings.

## 1.3 THESIS STRUCTURE OVERVIEW

The rest of this thesis contains the following. [Chapter 2](#) summarizes the state of the art that correlates with the present study. The experimental setup and its implementation regarding the social density estimation in consumer smartphone is described in [Chapter 3](#), while the result of the experiments and the data models are depicted in [Chapter 4](#). [Chapter 5](#) presents the discussion about the results. The conclusion of this thesis and the corresponding future work are described in [Chapter 6](#).



# 2

## STATE OF THE ART

---

We overview several studies that have been done with regard to social density estimation. The methods vary from video monitoring, audio tone tracking, and RF signal sensing. Furthermore, we present other applications of WiFi signal beyond crowd counting, such as pedestrian and queue monitoring. We also present studies related to ambulatory assessment that correlates with smartphone-based social density estimation. Finally, we present a brief discussion of the literature review that define our direction for the rest of the study.

### 2.1 CROWD COUNTING TECHNIQUES

Several studies have been proposed to investigate possible technique to estimate the level social density. In this section, we summarize social density estimation techniques which use several approaches, such as WiFi, Bluetooth, other Radio Frequency (RF) technology, video, and audio recording.

#### 2.1.1 WiFi-based Techniques

Within WiFi-based method, several approaches utilize one of the following, RSSI, fingerprinting based, probe request, and CSI.

##### 2.1.1.1 Received Signal Strength Indicator

Received Signal Strength Indicator (RSSI) is one of the power measurement in a received radio signal. In WiFi, RSSI indicates the power level received by the device after signal attenuation in the wireless transmission media. RSSI is represented in negative value, where a closer value to zero indicates stronger received signal strength.

Yoshida et al. presented a technique to infer the number of people present in a controlled environment [46]. The authors use linear regression and support vector regression to model the number of people from collected dataset. The method is based on a fact that says the propagation of radio signal might be affected by people in the room, which affects the RSSI. The method uses conventional WiFi AP and computers to count people tested in laboratory experiments. Benchmarking with manual counting of people via camera as the ground truth, the method achieves 77.2% of accuracy.

Another proposed method to count people using RSSI is proposed by Depatla et al. [12]. Like the method proposed by Yoshida et al.,

it works based on the fact that says people leave traces in **RF** signal, as they may block the line of sight propagation, resulting in a scattered radio signal, also known as multi-path phenomenon. The authors construct mathematical models of signal loss, which are based on the probability distribution of the received signal amplitude as a function of the total number of occupants, using Kullback-Leibler divergence. The method is tested both indoor and outdoor using a pair of pre-installed WiFi cards, involving nine people. The method achieves 96% of accuracy for indoor and 63% for outdoor, when using omni-directional WiFi antenna. 100% of accuracy is also achievable using a directional antenna. Manual counting of people is used as ground truth information and the method is not tested for large crowds.

Other than counting people, WiFi **RSSI** can also be used to monitor human queues [40]. The approach makes use of a single WiFi monitor located at the queue head (service desk), as opposed to conventional methods that rely on cameras or floor mats. The WiFi monitors collect signal traces indicated by **RSSI**. As the **RSSI** increases, the person is getting closer to the service desk. The queue phases are divided to three groups, namely wait, service period, and leave. A laboratory experiment with 90 traces of persons was conducted to test the method. Two-fold cross validation with manually logged ground truth are used to evaluate this method. Although the method seems promising, each queuing user is equipped with custom Android application that sends WiFi packets at 10 packets/second, which makes this method quite restrictive.

Similar to queue monitoring, Schauer et al. proposed a method to estimate pedestrian flows and crowd densities [32]. The authors propose several approaches, namely a naïve approach, which works by only counting MAC addresses, a time-based approach, which also includes time information, a **RSSI**-based approach, that includes WiFi signal strength, and a hybrid, which combines **RSSI** and time. The authors use WiFi and Bluetooth in two identical and synchronized laptops to sense the signals. To test the method, the authors created single realistic scenario at a German airport in 16 days. Ground truth is provided by security check in German airports. The result shows that Pearson correlation is 0.75 on average in the hybrid approach and 0.93 for the best case. Although Bluetooth is mentioned, it is not significant in terms of results, as the Bluetooth-based method has a high number of false positive.

Human movement pattern analysis is also a subject of study in a work by Abedi et al. [1]. The method utilizes **MAC** address data to determine spatio-temporal movement of humans in terms of space utilization. Specifically, this method leverages **MAC** address in Bluetooth and WiFi. This method alleges that it could track group gathering and behavioral pattern. CrossCompass by Acyclica Inc. is used

to capture **MAC** address from both Bluetooth and WiFi. To prevent biased result, passing visitors, who visit the location below 4 minutes, are filtered. Groups of people are determined when several **MAC** addresses enter and exit the lounge area in almost similar time. A **MAC** address scanner is required to perform the measurement. The author conducted three weeks of data collection, resulting in data that consists of timestamp, **MAC** address, and **RSSI**, which is stored in 35,000 log lines.

#### 2.1.1.2 WiFi Fingerprinting

WiFi fingerprinting is a process of recording the **RSSI** from several Access Points (**AP**) in range and storing this data in a database together with the coordinates of the recording device. Later, any **RSSI** vector at an unknown location is compared to those stored in the fingerprint database and the closest match is defined as the exact location. WiFi fingerprinting system may give an accuracy between 0.6 and 1.3 meters [47].

Compared to the conventional **RSSI** metering, WiFi fingerprinting requires prior knowledge of the monitored area. In the other way, a WiFi fingerprint map must be present to infer any **RSSI** vector relative to a particular location.

Kjærgaard et al. proposed a method to count and track pedestrian flocks (groups) using WiFi fingerprinting [23]. The technique uses a clustering method that operates in three different feature space: spatial (coordinates in controlled area), signal strength (within signal strength space), and pseudo-spatial (computed in a space with a defined distance unit). In the controlled environment, the author tested the method involving 16 subjects walking in a building at several floors, using manually annotated video recordings as the ground truth. The method provides 85% F-measure accuracy. The authors allege that the accuracy may increase if the number of **AP** increases.

WiFi fingerprinting can also be combined with other techniques to monitor indoor crowds. A combination of WiFi fingerprinting and Pedestrian Dead Reckoning (**PDR**) is proposed by Radu et al. to monitor indoor environment by means of crowd-sourcing [30]. This method requires pre-deployed WiFi **APs** in the monitored area. This method, which works in indoor location, utilizes Ekahau mobile survey as the ground truth source.

Another combination of WiFi fingerprinting is presented by Ahmet et al. [3]. The study presents a method that combines geo-fencing with coarse WiFi fingerprinting for building evacuation. The method works on an assumption that says almost everyone carries a mobile phone and majority of the population have smartphones. Two major components of the method are magnetofence, a novel geo-fencing, which uses permanent magnets and phone magnetometer along with inertial sensors. Motion estimator is used to determine when to infer

location and when to send data to the server. When a phone detects a fence, it locates the location and sends it to the server. The authors tested the method in an evacuation experiment that involves approximately 350 people. Using manual counting as the ground truth, the method achieves 98% of accuracy in counting people getting in and out.

#### 2.1.1.3 WiFi Probe Request

Probe request is a data packet broadcast by WiFi-enabled devices, e.g., smartphones and tablets, to scan for available AP. This mechanism is part of active scanning in WiFi standards [19], which is more energy efficient than passive scanning. Probe request packets contain detailed information about the corresponding device that broadcasts the packet, which makes it exploitable for people tracking.

Schmidt showed that it is possible to infer social density estimation using captured WiFi probe request packets [33]. The method works by capturing WiFi-packets from smartphones using dedicated scanning device: a Raspberry Pi with battery and D-Link WiFi module. The method notes device leaving or entering the monitored area. Database of MAC address is created to filter out probe requests coming from any device other than a smartphone that may otherwise falsify the result. As a ground truth, four directions photos are taken in every 10 minutes. The method gives satisfying results but the author says that the method is not suitable for high-accuracy applications.

Yaik et al. studied the correlation of crowd density with WiFi probe request [45]. The study was performed in the university open day event that lasted for 8 hours, from 9:30AM to 4:30PM. As a ground truth checking, the authors used manual people counting using tally counter at the entrance of the event. A WiFi monitor, which scan WiFi probe request, is placed close to the entrance of the venue, nearby the tally counter volunteers. The study reveals 0.893 of correlation coefficient, which is considerably strong.

Furthermore, Barbera et al. demonstrated that it is possible to infer the existence of social relationship other than merely crowd count [8]. The method assumes that two or more users sharing one or more Service Set Identifier (SSID) in their Preferred Network List (PNL) provide some information about the existence of social relationship between them. Affiliation network is used to construct the graph. The authors tested the method on a dataset collected in large national events, international events, city-wide probes, train station, and university, within three months of data collection. However, the author did not mention the accuracy of the proposed method.

#### 2.1.1.4 WiFi Channel State Information

Channel State Information ([CSI](#)) points to known channel properties of a communication link that describe how a signal propagates and how the effects of, for instance, scattering, fading, and power decay, affect the communication. [CSI](#) helps WiFi AP to adapt the transmissions according to current channel conditions.

Xi et al. presented that [CSI](#), which is sensitive to environment variation, has a monotonic relation with the number of moving people [43]. The authors conducted experiments using multiple off-the-shelf IEEE 802.11n devices, organized in grid array, with up to 30 people in the area. Using manual counting of people as the ground truth, the authors implement Grey Verhulst model to model and train the data. Furthermore, the authors say that it is hard to get the ground truth when the number of people is large.

In contrast with the study by Xi et al., Domenico et al. presented a method using [CSI](#) without prior data training [13]. The method works by implementing single WiFi transmitter and receiver from off-the-shelf devices. For less or equal than 2 person, the method achieves up to 91% of accuracy. However, it is only possible to count up to 7 people and it is harder to perform the method especially if the WiFi transceiver (smartphone) is placed in the user's pocket.

#### 2.1.2 Bluetooth-based Techniques

Several works present studies of crowd counting techniques using Bluetooth. The techniques vary from individual to collaborative monitoring by using signal strength or Bluetooth [MAC](#) addresses.

Chen et al. proposed a method to track social contextual behavior using consumer smartphones [10]. By using only Bluetooth traces captured in smartphones, the authors ran a data collection involving three volunteers in one to two weeks of experiments. A volunteer run the experiment during the day and fill out a questionnaire for ground truth checking at the end of the day. The proposed method uses six representative contextual behaviors for classification, namely working indoors, walking outdoors, taking subway, shopping in the mall, dining in the restaurant, and watching movie in cinema. C4.5 decision tree and 10-folds cross validation are utilized as evaluation method. The author alleges that the proposed method can achieve 85.8% of accuracy. However, this method has very limited contextual behavior, which we think is not representative. Although a smartphone is utilized in this method, no explicit explanation about the crowd counting technique is present.

As opposed to individual crowd counting, Weppner et al. proposed an approach of crowd counting using Bluetooth-based collaborative monitoring in smartphone [42]. The crowd densities are quantized into 7 groups, ranging from nearly empty to extremely high (crowded),

which will be the features in the training phase. The experiments were set up for 3 times, with 4 hours of duration each. The authors recruited 10 students to carry out the experiments, equipped with scanning smartphones that scans nearby Bluetooth signals. Using images taken by cameras as ground truth, the proposed method achieves 75% accuracy.

Another method of Bluetooth-based collaborative sensing achieved 82% accuracy in the best case [41]. The authors equipped several volunteers with Bluetooth enabled smartphone to walk through the crowd in three days analysis at Octoberfest, Munich, Germany. The estimation was carried out twice, individually and collaboratively. Features for individual estimation are the number of Bluetooth devices, the mean signal strength, and the variance of signal strength, while features for collaborative are average number of devices, variance of the number of device, and variance of all signal strength. The authors quantize the pictures depicting the ground truth into four levels, namely  $0.1 \text{ people/m}^2$ ,  $0.2 \text{ people/m}^2$ ,  $0.3 \text{ people/m}^2$ , and  $0.4 \text{ people/m}^2$ . The method gives 63% for individual features and 81% for collaborative features.

In contrast with smartphone based social density estimation method, Versichele et al. proposed a crowd counting method by employing 22 Bluetooth scanners placed around the monitored area [39]. The proposed method analyzes spatio-temporal movements of visitors in Ghent Festivities, a 10 days event with approximately 1.5 millions of visitors. The scanners include a combination of class 1 (larger area) and class 2 (smaller area) Bluetooth scanner. The experiments yielded large datasets consisting of 260 millions of lines. To prevent counting the same person multiple times, the authors label a record with in, out, or pass. Moreover, to reduce bias from other Bluetooth devices, Bluetooth address are used to distinguish different types of Bluetooth device, e.g., phone or car handsfree, etc. Compared to the ground truth, which is provided by official visitor count, the method counts 11% of the actual populations.

### 2.1.3 Other RF-based Techniques

RF based techniques are not only limited to WiFi and Bluetooth, but also, for instance, Wireless Sensor Network ([WSN](#)) and Radio-Frequency Identification ([RFID](#)) as described below.

Nakatsuka et al. published a technique to estimate the crowd density using ZigBee<sup>1</sup> [28]. The proposed method exploits the ZigBee power measurement profile, namely [RSSI](#) and Link Quality Indicator ([LQI](#)), using a fact that says the signal strength is attenuated by human body as human body consists of approximately 60% to 70%

---

<sup>1</sup> An IEEE 802.15.4-based specification for high-level communication by creating personal area networks for small and low-power digital wireless transmission.

of water, which is a medium that absorbs radio waves. The technique uses two ZigBee nodes, eight meters apart, for transmitting and receiving ZigBee signals. A model is constructed to count the number of people located between the transmitters. The method was tested using two scenarios, namely sitting and walking between nodes, which resulted in 300 samples.

Another technique to estimate social density using [WSN](#) was proposed by Yuan et al. [48]. Contrasted with a technique from Nakatsuka et al. that employs at least two ZigBee devices, Yuan et al. employ tens of ZigBee devices. The authors conducted experiments in controlled situation of 16 grids (small scale) and 400 grids (large scale), in which sensor nodes were deployed in the point of intersection of the grids. The experiment yielded 168 samples (small scale) and 600 samples (large scale) of data. The result shows that [RSSI](#) reading decreases as more people enter the grid. Then the variation signal strength is divided by the crowd density based to obtain the model. Using k-means clustering technique with manual counting as ground truth, the method can achieve 96% accuracy.

Another method works by utilizing [RFID](#) tags. Mitchell et al. installed [RFID](#) tags and readers across the city of Mecca, Saudi Arabia, to monitor pilgrims in Hajj [27]. Basically, each pilgrim is given an [RFID](#) tag, and optionally a smartphone application to locate and give them useful information or possibly sending an emergency signal. [RFID](#) readers are placed in strategic locations around the Hajj area dividing it into several zones. For the future works, the authors plan to involve mobile or wireless provider cell tower to provide location and extend the service to buses or other service.

#### 2.1.4 Other Crowd Counting Techniques

Vattapparamban proposed an approach of crowd counting and occupancy monitoring using an Unmanned Aerial Vehicle ([UAV](#)) for outdoor monitoring [38]. The proposed technique employs WiFi pineapple device as indoor monitoring mode. The approach works similarly by exploiting WiFi probe request transmitted by nearby devices. Having the WiFi probe request packets captured, the technique then applies [k-NN](#) clustering and linear least square estimation to infer occupancy in a certain location. The author focuses on search and rescue applications. However, there are no validation checking and accuracy mentioned, as the author are more focused on occupancy tracking techniques. The study also describes [MAC](#) address randomization observed in iOS 8, which is a potential threat for [MAC](#) address based crowd counting technique.

Audio tones are also tested to be a potential method to infer the level of social density [21]. The approach uses the speakers and microphones available on consumer smartphones. The approach works by

implementing tone exchange, in which each mobile device is installed with a tone counting application. A smartphone generates one or more simple tones and the other smartphones hear these unique patterns of sounds. The tone generating is done by using 2 main method: uniform hashing (involving a single tone) and geometric hashing (involving multiple tones). Fast Fourier Transfer ([FFT](#)) is used to analyze the received sound (peak analyzer), thus making background noise as one of the issues. Tested in three indoor and outdoor places with 25 Android phones, the proposed technique achieved up to 90% accuracy. The author also alleges that the proposed approach is 81% more energy efficient than WiFi-based approaches.

The smartphone's microphone can also be utilized as a speaker counting instrument. The speaker counting technique estimates the number of people who participate in a conversation. Several works have been dedicated to this topic [2, 4, 44]. Among those approaches, Mel-Frequency Cepstral Coefficient ([MFCC](#)) is one of the effective coefficients for speech processing. The methods only work with four speakers or less. Furthermore, these techniques require *a priori* information about the speakers, making it impossible to apply this method in our case. Another alternative of voice related technique, VAD might be a good candidate to detect whether there is a person speaking or not [34], but this technique only yields boolean value.

Ryan et al. proposed a method to count crowds using surveillance cameras [31]. The method works by using tracking and local features to count the number of people in each group, so that the total crowd estimate is the sum of the group sizes. The authors use tracking to improve the robustness of the crowd count estimation, by analyzing the history of each group, including splitting and merging events. The study used manually annotated video recording as a ground truth. The method shows 1.6 mean average error. Video-based technique is also possible to be directly performed in the surveillance camera, equipped with low-cost single-board computer [11]. The method achieves similar error as what has been proposed by Ryan et al., with mean average error of 1.68.

## 2.2 AMBULATORY ASSESSMENT

Ambulatory Assessment is concerned with data capturing for decision making of patients treatment. Ambulatory assessment in everyday life is an emerging field, for instance, when dealing with patients suffering from neuropsychiatric disorders. Several works have addressed the ambulatory assessment topic, especially with mobile devices as the measuring instruments [6, 7, 14, 24].

A study presents an identification of relevant sensor sources in ambulatory assessment [7]. The study implements an ambulatory assessment for context-aware Experience Sampling Method ([ESM](#)) in

a mobile phone application. The authors conducted an online survey with ambulatory assessment experts to identify relevant sensor sources and contexts. The result alleges that time, date, and user activity, followed by location and notifications are prioritized smartphone sensors for context-aware experience sampling method. The authors also conducted a feasibility test to confirm that all mentioned relevant sources are available and accessible on Android devices.

Furthermore, Bachmann proposed smartphone-based social interaction sensing [6]. The proposed technique monitors social interaction using WiFi, Bluetooth, audio, and user interactions in the smartphone. The technique uses nearby Bluetooth devices and devices within the same WiFi network as indicators of social interaction. Moreover, the notifications and subjects' activity in messaging applications are monitored. The technique also implements unsupervised clustering method to infer the number of speaker in the patient's daily conversation, although it is not yet completed. The technique is also monitoring the subjects location using Global Positioning System (GPS) and WiFi MAC addresses. The logged data are aggregated in certain time intervals, e.g., a week, for later analysis. Although the authors did not mention a reliable ground truth, they alleged that this proposal could achieve up to 76% of accuracy for phone interaction.

Another ambulatory assessment approach is proposed by Luo, et al. [24]. The proposal puts attention on the daily multi-modal social activities done in smartphone, e.g., phone calls, text messages, and social network applications. Moreover, the technique also monitors phone books, GPS, WiFi, and Bluetooth proximity. The phone books are labeled on groups (friend, colleague, family, etc). In the other way, contact classification, place detection, proximity analysis, and application usage. Bluetooth is used only for detecting the type of nearby device, for instance, a salesman actively meets clients, while a software developer tends to stay in the office with the team members nearby.

### 2.3 DISCUSSION

We present the summary of the literatures in [Table 1](#), based on the applications and their corresponding technologies. We categorize the instruments to gather the data into two categories, namely static and dynamic. Static instruments refer to instruments which are bound to a host that makes it relatively unmovable, while dynamic instruments refer to handheld devices which have certain physical dimensions and weight so that it is small enough to operate in the hand.

As we can see in [Table 1](#), approaches to crowd counting mostly involve static instruments, which are undesirable for passive behavioral monitoring as it requires static dedicated devices to monitor the patients. The ambulatory assessment techniques that works in a

Table 1: Literature review summary.

Applications	Technologies	Instruments	
		Static	Dynamic
Crowd counting	WiFi ( <a href="#">RSSI</a> )	[ <a href="#">12</a> , <a href="#">38</a> , <a href="#">40</a> , <a href="#">46</a> ]	
	WiFi (Fingerprint)	[ <a href="#">3</a> , <a href="#">30</a> ]	
	WiFi (Probe request)	[ <a href="#">45</a> ]	[ <a href="#">33</a> ]
	WiFi ( <a href="#">CSI</a> )	[ <a href="#">13</a> , <a href="#">43</a> ]	
	Bluetooth	[ <a href="#">39</a> ]	[ <a href="#">41</a> , <a href="#">42</a> ]
	ZigBee	[ <a href="#">28</a> , <a href="#">48</a> ]	
	Audio tone		[ <a href="#">21</a> ]
	Surveillance camera	[ <a href="#">11</a> , <a href="#">31</a> ]	
Speaker counting	Microphone		[ <a href="#">2</a> , <a href="#">4</a> , <a href="#">44</a> ]
Crowd surveillance	RFID	[ <a href="#">27</a> ]	
Voice activity detection	Microphone		[ <a href="#">34</a> ]
Pedestrian monitoring	WiFi ( <a href="#">RSSI</a> )	[ <a href="#">1</a> , <a href="#">32</a> ]	
	WiFi (Fingerprint)	[ <a href="#">23</a> ]	
Social relationship deduction	WiFi (Probe request)	[ <a href="#">8</a> ]	
Contextual behavior deduction	Bluetooth		[ <a href="#">10</a> ]
Ambulatory assessment	Smartphone sensors		[ <a href="#">6</a> , <a href="#">7</a> , <a href="#">14</a> , <a href="#">24</a> ]

single smartphone do not address social density estimation topic explicitly. Several approaches that make use of dynamic instruments exist, which implement either WiFi (Probe request), Bluetooth, or audio tone.

We plan to use a single smartphone as a monitoring tool of social density in the surroundings as it is able to work almost everywhere with no prior infrastructures. Based on [Table 1](#), we present a summary of available sensors in common modern smartphones in [Table 2](#), ranging from RF-based sensor, such as [GPS](#), Near Field Communication ([NFC](#)), and WiFi to motion sensor, such as gyroscope and accelerometer.

In the present study, we would like to find a method to estimate the level of social density in the surrounding as a passive behavioral scheme. We make use of any sensors available on the smartphone, which can be used no matter how the smartphone is operated and positioned. We plan to use WiFi and microphone.

However, we do not use WiFi probe request based monitoring directly in the smartphone, as it is not possible due to restrictions in the

Table 2: Summary of typically available sensors in a modern smartphone.  
Sensors written in bold are preferred for social density estimation.

Sensor Type		
<b>Bluetooth</b>	NFC	gyroscope
<b>WiFi</b>	cellular call	proximity
<b>Microphone</b>	touch screen	compass
<b>Camera</b>	fingerprint	barometer
	accelerometer	GPS

mobile operating system. Probe request based estimation only works in WiFi monitor mode [37, 46] and monitor mode is only accessible if the smartphone is rooted, jailbroken, or installed custom Read Only Memory (**ROM**), which is considered as illegal in most countries [17]. We plan to note the number of available **AP** in the surroundings using WiFi and record ambient noise using microphone.

To validate the result, we need to obtain the ground truth of the surroundings. Several methods have been proposed, ranging from manual counting using tally counter to annotating video recording, but it is known that getting ground truth is difficult especially for large number of crowds [43]. In the present study, we use two approximations of ground truth, namely image based method and WiFi probe request based method, as probe request is tested to have a strong correlation with the number of people nearby [45]. However, recent mobile Operating System (**OS**) has developed a technique named **MAC** address randomization [15, 25]. By using the randomization technique, the devices randomize their **MAC** address in each particular circumstance to increase the privacy of the user and thus making the current people tracking technology no longer works [37].



# 3

## EXPERIMENTAL SETUP

---

In this chapter, we present our proposed experiment, along with the goals and how to set it up. We plan to devise a social density estimation, which could replace the manual counting method. We aim to implement this estimation as a passive behavioral monitoring system in which smartphones are used as measuring instruments. The estimation may allow us to differentiate between healthy and non-healthy individuals, especially related to neuropsychiatric disorders.

We start with general experimental design of smartphone based social density estimation. We then explain the technical implementation of our initial experiment design. Lastly, we present the method of data extraction from collection of raw log files and how we evaluate the data.

### 3.1 EXPERIMENTAL DESIGN

In the present study, we would like to know whether we can use smartphone sensor readings as a way to estimate the level of social density in the surroundings. To do so, we compare the level of social density with the smartphone sensor readings, collected from subjects under real world condition. We try to get wide range of social density levels, from low to high social density level, and we try to employ as many sensors as possible.

It is known that getting the ground truth of the actual social density level in public spaces is difficult [41]. As an option, we use estimation approach. There are several means to estimate social density level, for instance, manual counting of people using tally counter as what retail stores do, manual counting by taking notes on paper, manual head counting on captured pictures, and electronic based counting. In the present study, we employ manual and electronic-based method, by using head count of captured pictures and unique device counting. We select two methods so that we can get more insight of the actual social density condition.

Modern smartphones are equipped with broad range of sensors, such as **a**) Bluetooth, **b**) WiFi, **c**) microphone, **d**) NFC, **e**) camera, **f**) cellular network card, **g**) touch screen, **h**) fingerprint, **i**) accelerometer, **j**) gyroscope, **k**) proximity, **l**) compass, and **m**) barometer. However, not all of them are exploitable for estimating the level of social density. We have selected several sensors which are potentially usable especially for passive monitoring. We select WiFi and microphone to estimate the social density level because those sensors are able to per-

ceive the surroundings no matter how the smartphone is operated and positioned.

To collect samples of real world condition with high variation of social density level, we selected locations that represent low and high social densities as well as indoor and outdoor locations. We plan to do our experiment where WiFi AP use is not restricted, i.e., people are free to set up their own WiFi AP in anyway they prefer. Some locations where the use of WiFi is restricted exist, for instance, University of Groningen complex. In the university complex, eduroam is the only the official available AP and the occupants are not allowed to install their own AP. Thus, making this not suitable for our study.

We collect the data of a certain location once per day, when the location gets its maximum number of visitor. To get the insight of the variation between days, we collect the data in several days of experiment. Furthermore, as the social density level in a certain location is very dynamic, we also conduct several data collections per day, e.g., taken at 09:00, 12:00, and so on, at a single location.

In the present study, we evaluate the correlation of social density level and the smartphone sensor readings. If the social density level and smartphone sensor readings have a correlation, we can say that it is possible to estimate the level of social density using smartphone sensor readings but we have to further investigate the accuracy and the limitation of this approach. We also examine the variation of the correlation in different scanning days and scanning time.

### 3.2 IMPLEMENTATION

In this section, we describe how we implement our proposed experimental design. We firstly study the behavior of MAC address randomization, observed in both Android and iOS devices [15], to minimize the bias in device count based social density level estimation. Then, we describe the method we use to estimate the social density level. We also specify the smartphone sensors that we use in the present study and their corresponding outcomes, followed by a description of the different location and timing setup. Lastly, we finalize with an explanation about technical scanning mechanism.

#### 3.2.1 MAC Address Randomization Investigation

MAC address randomization allows WiFi enabled devices to rearrange or change their MAC address in particular circumstance to increase the privacy of the user. The randomized MAC address uses locally administered address, which is indicated by the second-least-significant bit of the first octet of the address [19]. The original MAC address uses universally administered address, which is uniquely assigned by its manufacturer.

We capture the probe request packets from nearby WiFi-enabled devices and store the captured information to a log file. Probe request is a data packet broadcast by WiFi-enabled devices to scan for available AP. This mechanism is part of active scanning in WiFi standards [19], which is more energy efficient than passive scanning.

In the current investigation, we note **a)** the timing when the randomized MAC address appears, **b)** the Sequence Number (**SN**) field<sup>1</sup> of the packet, and **c)** the variation of the **MAC** address.

We investigate the randomization behavior in both iOS, using iPad mini with iOS 10.0, and Android, with LG Nexus 5X with Android 6.0 (Marshmallow). When experimenting on iPad mini, we turn the LG Nexus 5X off, and vice versa, to make sure that each device is not disturbing each other. We do this experiment within 30 minutes for each device.

We choose a remote area where we cannot detect any other probe request except from our device to perform the experiment. We use Wireshark<sup>2</sup>, a popular network protocol analyzer software, to capture and store the captured probe request packets to a pcap file. We run this experiment on Apple's MacBook Air with built-in WiFi card that supports monitor mode.

### 3.2.2 *Ground Truth Estimation*

We use two approaches to estimate the crowd count in the surroundings, by using manual counting on photographic images and unique **MAC** address counting from captured probe request. We estimate the crowd count as a first approximation of the ground truth because it is known that getting the ground truth of crowd density in public spaces is difficult [41].

#### 3.2.2.1 *Device Count Based Estimation*

We count unique device in the surroundings by capturing probe request packets broadcast by nearby devices and noting the unique **MAC** addresses. Counting the unique **MAC** address will result in a list of unique device in the surroundings. We decide to implement unique **MAC** address counting in probe request as it is proven to be a promising method to estimate crowd count [45].

In device count based estimation, we capture the probe request packets in WiFi channel 1 (2.4 GHz). We are not capturing packets in other WiFi channels as WiFi-enabled device will broadcast the probe requests to all available WiFi channels [19]. Furthermore, capturing packets in every WiFi channels is not doable on a single device, due to the limitation of a device of being only in a certain WiFi channel

---

<sup>1</sup> The **SN** field is a 12-bit field indicating the sequence number of a probe request packet. This marks the order of how the packets are broadcasted.

<sup>2</sup> <https://www.wireshark.org>

at a time. Capturing packets in multiple channels requires channel hopping, i.e., hopping from one channel to another within a short time interval, which is considered a lossy method and prone to losing some probe request packets.

### 3.2.2.2 Manual Head Count in Photographic Images

In addition to device count based estimation, we incorporate another crowd counting estimation method using photographic images. We consider this method as an estimation of the ground truth, as photographic images are subject to light and sight, i.e., physical obstacle would interfere the final result.

We prefer photographic images that support wide Field of View (**FOV**), i.e., able to cover 360 degrees of horizontal **FOV**. Some of the options that we consider are panoramic photograph and wide-angle photograph. During our early testing, panoramic photograph using smartphone suffers from misaligned images, as panoramic photograph is a computer-based concatenated images taken in different timestamp. This causes some objects to appear more than once, or even not at all. Thus, we decide to use wide-angle photograph to achieve 360 degrees of horizontal **FOV**.

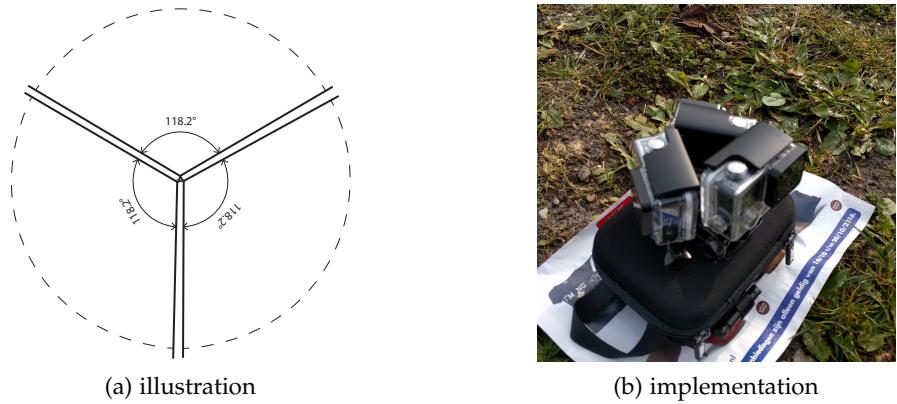


Figure 1: GoPro arrangement to achieve 360 degrees of horizontal **FOV**.

We use GoPro<sup>3</sup> Hero4 silver wide-angle action camera to get the photographic image. According to GoPro technical specifications [36], its maximum horizontal **FOV** is 118.2 degrees in 16x9 wide mode with 17.2mm focal length. To achieve 360 degrees of **FOV**, we use three GoPro cameras positioned in circular arrangement, as depicted in Figure 1. Although some blind spots are still present, this setup is still able to capture objects within intended range.

We employ time-lapse photography technique to capture the surroundings in preference to normal video capture. We capture a still image in each five seconds, resulting 12 still images per minute. We

---

<sup>3</sup> <https://gopro.com>

do not employ video recording to conserve GoPro battery. As stated in GoPro technical specification of battery life [35], estimated battery life of taking video is roughly two hours, while our experiment is planned to be more than two hours (plus back-up time). We consider battery replacement during data collection is impractical. Moreover, taking time-lapse images would increasingly save more data storage.

In the end, we manually do head counting based on captured images grouped in each time interval. We make sure that one person is counted once to avoid counting the same person multiple times.

### 3.2.3 *Smartphone Sensor Utilization*

We use WiFi and microphone as sensors because those sensors are able to perceive the surroundings no matter how the smartphone is operated and positioned. This means that the passive monitoring process can still running in background regardless how user use the smartphone. In the other hand, camera based approach requires the smartphone to be positioned accordingly to be able to capture appropriate images for crowd counting.

#### 3.2.3.1 *WiFi scanning*

Ideally, we use probe request based crowd estimation, as discussed in [Section 3.2.2.1](#), directly in the smartphone. However, this is not possible due to restrictions of the mobile operating system. Probe request based estimation only works in WiFi monitor mode [37, 46] and monitor mode is only accessible if the smartphone is rooted, jailbroken, or installed in custom ROM, which is considered as illegal in most countries [17].

We count the available AP nearby and observe its signal strength in place of unique MAC address counting in probe request packets. We take note of AP's Basic Service Set Identifier (BSSID), which is unique and based on MAC address. We are not particularly interested in AP's SSID because there might be multiple APs with the same SSID. We measure the signal strength of the AP in RSSI.

#### 3.2.3.2 *Ambient Noise Recording*

In addition to WiFi scanning, we also record the ambient noise to perceive nearby condition, assuming that more crowded locations will have higher ambient noise, while less crowded locations will have lower ambient noise. We measure the ambient noise in decibels (dB). Moreover, we implement speaker count algorithm based on unsupervised learning method proposed by Xu et al [44]. This approach will estimate how many persons are speaking in the audio recording.

Table 3: Summary of location selection for the experiment. We plan to run the experiment from Wednesday to Saturday, October 26<sup>th</sup> to 29<sup>th</sup>, 2016.

Location	Type	Crowd Level	Scanning Time
Remote area	outdoor	low	11:00-11:30
Home	indoor	low-medium	08:15-08:45
Paddepoel shopping center	indoor	medium-high	14:45-15:15
Grote markt	outdoor	high	12:00-12:30

### 3.2.4 Location and Timing

We select four different locations that represent low and high crowd densities as well as indoor and outdoor locations. [Table 3](#) summarizes our location selection.

To save GoPro battery, we select 30 minutes as the scanning duration in each location. As we are picking four different locations, this would sum up to two hours of image capture. We also select daytime as the preferred scanning time because that is the time when most crowd are observable.

Speaking of the location selection, remote area is a far-off outdoor location located in Paddepoelsterweg, Groningen. The surroundings are mostly trees and grasses. No visible buildings or dwellings exist but there is only a narrow asphalt pathway where sometimes people or vehicles pass by. Approximately no more than 5 individuals are present at the same time. We carry out the experiment in this area during daytime, from 11:00 to 11:30.

Aside from the remote area where the crowd is less likely to be observed, we conduct a data collection at home during breakfast time, from 08:15 to 08:45. This indoor location is located at Planetenlaan 264, Groningen. Currently, there are only 6 inhabitants living in the house. However, people sometimes pass by in front of the house and they might influence the measurement result.

Paddepoel shopping center is an indoor shopping center located in Paddepoel district, Groningen. The shopping center, which is 14.920 m<sup>2</sup> in size, holds around 90 shops [\[5\]](#). According to statistics [\[5\]](#), it has approximately 90.000 visitors per week. We perform the experiment during the peak hours, which is from 14:45 to 15:15.

Grote markt is a open city square which is located in the center of Groningen city. This location is able to hold roughly 8000 visitors at the same time, according to the official government statement [\[16\]](#). We are planning to do data collection in Grote Markt around mid day, from 12:00 to 12:30.

To see the variance between days, we plan to do the experiment in four days, ranging from weekdays to weekend. We plan to run the

experiment from Wednesday to Saturday, October 26<sup>th</sup> to 29<sup>th</sup>, 2016. We stick to the same schedule for each location.

#### 3.2.4.1 Effect of Scanning Time

We also plan to conduct another experiment aiming at seeing the variance between scanning times. We select Grote Markt in Groningen as the scanning location because Grote Markt is the location in which the level of social density is the highest among the other three locations that we choose ([Section 3.2.4](#)). The four different scanning time are **a) 09:00, b) 12:00, c) 15:00, d)** and 18:00, taken in the same day. The rest of the experimental setup of this process remains the same as previous experiment. However, we do not employ GoPro time-lapse image capture.

#### 3.2.5 Scanning Mechanism

In summary, we use three inputs to collect and to perceive the surroundings, namely WiFi, microphone, and wide-angle camera. The data collection process is split into several small cycles of  $x$  minutes to avoid biased result caused by randomized [MAC](#) address. The  $x$  will be determined by the [MAC](#) address randomization experiment described in [Section 4.1](#). We keep repeating the cycle until we reach 30 minutes. [Figure 2](#) depicts the description of each cycle.

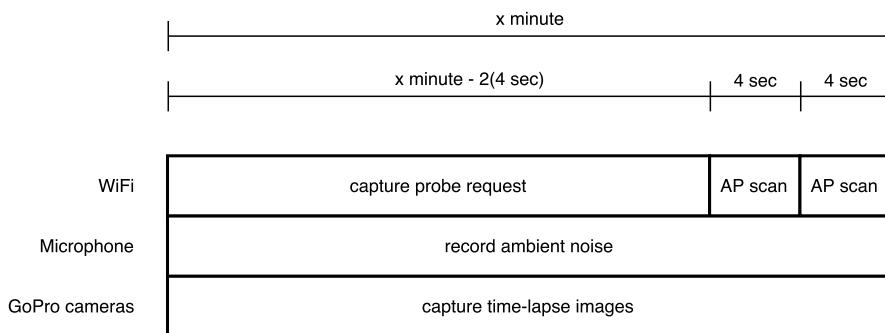


Figure 2: Sensor measurement in each cycle.

As depicted in [Figure 2](#), we work with WiFi to capture the probe request and count the number of available [AP](#). We cannot perform these tasks simultaneously, because capturing probe request must be carried out in the WiFi monitor mode. Monitor mode is one of the seven modes, namely master (acting as an [AP](#)), managed (acting as a WiFi client), ad-hoc, mesh, repeater, and promiscuous, that 802.11 WiFi cards can operate in [[19](#)]. Being in monitor mode, the WiFi card is disconnected from any wireless networks. Thus, we decide to firstly capture probe request before counting the available [AP](#). We count the available [AP](#) twice to achieve better stability. Both microphone and

Table 4: Smartphone sensor readings and the extracted parameters.

Sensor	File format	Extracted parameter	Extraction method
WiFi	.pcap	unique MAC addresses	text processing
	.txt	AP signal strength	text processing
		AP count	text processing
Microphone	.wav	speaker count	machine learning
		peak level	audio processing
		root mean square	audio processing
GoPro cameras	.jpeg	head count	manual counting

GoPro cameras continuously recording ambient noise and capturing time-lapse images in each cycle.

We use Apple MacBook Air, with built-in WiFi card and microphone, as a WiFi scanning and audio recording device. To capture probe request, we use `tcpdump`<sup>4</sup>, a free packet analyzer that runs under the command line, and we use console based Apple’s AirPort utility, a software designed by Apple to manage WiFi network, to count available access point. The output of probe request capture is .pcap file, a standard file for capturing network traffic. AP counting results in plain text files. We work with console based audio editing software, Sound Exchange<sup>5</sup>, to record the ambient noise to .wav file format.

To automate the WiFi scanning and audio recording process, we write a bash script<sup>6</sup>. We label the resulting log files in the corresponding timestamp and location. Our code is publicly available online<sup>7</sup>.

### 3.3 DATA EXTRACTION AND EVALUATION

In each cycle of data collection, we obtain approximately 1,170 raw files in several formats. The files are, namely, .txt for AP list, .pcap for captured probe request packets, .wav for recorded ambient noise, and .jpeg for time-lapse images. Each file format needs different treatment to extract the data. Table 4 summarizes all sensor readings and the extracted parameters.

<sup>4</sup> <http://www.tcpdump.org>

<sup>5</sup> <http://sox.sourceforge.net>

<sup>6</sup> Bash script is a list of Bash commands that runs on a console based application to execute some particular tasks.

<sup>7</sup> <https://github.com/gtrdp/masters-thesis-guntur>

### 3.3.1 WiFi Raw Data Extraction

The WiFi based data collection result consists of two types of files, namely, .txt file for AP list from AirPort sensing, and .pcap file for captured probe request packets from tcpdump sensing. As shown in Table 4, we extract the number of available AP and the mean of AP's signal strength from this file, while the number of unique device is derived from the .pcap file. Each cycle produces two individual .txt and .pcap labeled in its corresponding timestamp.

Figure 3 and Figure 4 display the example of sensed AP list and captured probe request respectively. From the AP list depicted in Figure 3 we are interested in BSSID column and RSSI column to count the AP and measure its corresponding signal strength. From the captured probe request packets, as depicted in Figure 4, we note the Source Address (SA). We develop a text processing program written in Python script to extract preferred elements. We plot the result as a scatter plot using the matplotlib package in Python [18].

SSID	BSSID	RSSI	CHANNEL
Ziggo	6c:aa:b3:27:72:ac	-84	140
Ziggo	6c:aa:b3:26:af:bc	-87	140
Ziggo22211-5G	0c:54:a5:b8:25:e8	-84	136,-1
eduroam	c4:10:8a:60:d4:7c	-84	132,+1
Draadloos Groningen	c4:10:8a:20:d4:7c	-84	132,+1
Ziggo21616-5G	60:02:92:11:4d:e6	-87	128,-1
Ziggo	6c:aa:b3:26:af:0c	-86	116,+1
OneTeam	e2:55:6d:20:30:96	-85	108,+1
WEEKDAY Free WiFi	e2:55:6d:20:30:92	-85	108,+1

Figure 3: Example of AP list.

```
BSSID:ff:ff:ff:ff:ff:ff DA:ff:ff:ff:ff:ff SA:ec:1f:72:10:44:01
BSSID:ff:ff:ff:ff:ff:ff DA:ff:ff:ff:ff:ff SA:0a:0f:c6:e5:e0:cb
BSSID:ff:ff:ff:ff:ff:ff DA:ff:ff:ff:ff:ff SA:2e:60:57:70:d3:eb
```

Figure 4: A fragment of captured probe request packet.

### 3.3.2 Recorded Ambient Noise Extraction

We extract single ambient noise recording .wav file to three independent measurements, namely the Peak Level (PKLV), Root Mean Square (RMS), and speaker count. We use the same audio processing program as ambient noise recording process, namely Sound Exchange, to extract the PKLV and RMS from the .wav file, while we use

unsupervised learning method proposed by Xu et al. [44], namely Crowd++, to infer the number of speaker count.

The **PKLV** is the highest value of a total waveform, while **RMS** is the effective value or the mean. The **PKLV** and **RMS** are both measured in decibels (dB).

### 3.3.3 *Manual Head Counting*

We firstly collect all time-lapse images taken in different location from the three GoPro cameras. Then, we group each time-lapse image separately to the corresponding cycle and camera. We then manually do head counting based on those images and we avoid counting the same object that appears more than once multiple times. The result is a total head count of a cycle per location.

### 3.3.4 *Metrics and Evaluation*

We evaluate the correlation of social density level and smartphone sensor readings by using Pearson Product Moment Correlation Coefficient (**PPMCC**) or Pearson's r coefficient (sometimes written as  $\rho$ ), which measures the linear dependence between two variables. This measure has a value ranging between  $+1$  and  $-1$ , where  $1$  indicates total positive linear correlation,  $0$  indicates no linear correlation, while  $-1$  indicates total negative linear correlation. If the Pearson's r coefficient of social density level and smartphone sensor readings is positive, we can conclude that it is possible to estimate the level of social density using smartphone sensor readings. However, we should note on the accuracy and the limitation of this estimation approach. Furthermore, we also employ p-value to affirm the level of significance of the correlation.

Based on the collected dataset, we perform data analyses to model the data. The analyses result in models which are useful to predict the social density levels from new data. We focus on the accuracy of the prediction. We measure the level of accuracy by using Root Mean Square Error (**RMSE**) and residual error.

# 4

## RESULTS

---

In this chapter, we report and summarize the result of the experiments, which try to investigate the correlation of social density level and smartphone sensor readings. We also investigate the behavior of **MAC** address randomization that can potentially disrupt the data collection results. This chapter contains only important findings and we present the detailed findings in the Appendices.

We performed the experiments in October 2016. The results contain a total of 17,280 time-lapse images, 660 audio recording files, and 1,320 text-based log files. The sum of the total file size is approximately 41.7 Gigabytes.

Furthermore, we present analyses about social level density estimation using all data from four days experiment so that we can establish a data model that enables us to predict the level of social density using new sensor readings.

### 4.1 MAC ADDRESS RANDOMIZATION

We performed investigations of **MAC** address randomization in iPad mini and Nexus 5X that run different mobile **OS**, namely iOS and Android, respectively. We captured the probe request using Wireshark on MacBook Air laptop in remote area, where no other probe requests are observable. [Table 5](#) summarizes the result of **MAC** address randomization investigation using metrics that we mention in [Section 3.2.1](#).

Table 5: Summarized results of **MAC** address randomization behavior for both Android and iOS.

	iPad mini (iOS)	Nexus 5X (Android)
Timing	broadcast in every 2 minutes	broadcast in every 1 minute
Variation	address changed in every 4 minutes	address changed in each probe request packet burst
<b>SN</b> field	reset in each 2 probe request packet burst	no pattern observed

### 4.1.1 iPad mini

The iPad mini always transmitted randomized MAC addresses when the screen was on and off (standby). However, the iPad broadcast the probe request packets more actively while the screen was on, as we noticed that the iPad mini was sending out probe request packets within 3 to 10 seconds when the screen was on. If the screen was off, the iPad mini sent out the probe request packets approximately in every 2 minutes, or even 4 minutes. Interestingly, when the iPad was connected to an AP that we set up using Nexus 5X WiFi tethering, we observed the iPad mini's real MAC address in the probe request.

The randomized address was changed to completely different random address, i.e., all octets of the address, when the iPad mini woke up from standby mode. The address was also changed after roughly 4 minutes since the first random address transmitted. We also observed MAC address change when we toggled the WiFi on and off. The iPad behaved in the same manner regardless of Subscriber Identity Module (SIM) card installation.

Table 6: An example of captured probe requests from iPad mini in Wireshark. The colors mark out different bursts of probe request packets, while the red boxes indicate SN reset.

No.	Time	Source	Destination	Protocol	Length	Info
1159	1124.915502	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=1, FN=0,
1160	1124.934670	76:b4:49:90:63:7d	Broadcast	802.11	190	Probe Request, SN=2, FN=0,
1161	1124.936361	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=3, FN=0,
1162	1124.958279	76:b4:49:90:63:7d	Broadcast	802.11	190	Probe Request, SN=4, FN=0,
1163	1124.978569	76:b4:49:90:63:7d	Broadcast	802.11	190	Probe Request, SN=6, FN=0,
1164	1124.980782	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=7, FN=0,
1165	1126.474246	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=72, FN=0,
1166	1126.529789	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=73, FN=0,
1167	1126.815739	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=78, FN=0,
1168	1135.800826	76:b4:49:90:63:7d	Broadcast	802.11	190	Probe Request, SN=0, FN=0,
1169	1135.802607	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=1, FN=0,
1170	1135.820500	76:b4:49:90:63:7d	Broadcast	802.11	190	Probe Request, SN=2, FN=0,
1171	1135.822576	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=3, FN=0,
1172	1135.844115	76:b4:49:90:63:7d	Broadcast	802.11	190	Probe Request, SN=4, FN=0,
1173	1135.866207	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=7, FN=0,
1174	1135.887945	76:b4:49:90:63:7d	Broadcast	802.11	190	Probe Request, SN=8, FN=0,
1175	1135.890180	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=9, FN=0,
1176	1135.908392	76:b4:49:90:63:7d	Broadcast	802.11	190	Probe Request, SN=10, FN=0,
1177	1135.909913	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=11, FN=0,
1178	1138.498576	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=16, FN=0,
1179	1138.553863	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=17, FN=0,
1180	1138.612766	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=18, FN=0,
1181	1138.667970	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=19, FN=0,
1182	1144.688418	76:b4:49:90:63:7d	Broadcast	802.11	190	Probe Request, SN=0, FN=0,
1183	1144.690081	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=1, FN=0,
1184	1144.709332	76:b4:49:90:63:7d	Broadcast	802.11	190	Probe Request, SN=2, FN=0,
1185	1144.711186	76:b4:49:90:63:7d	Broadcast	802.11	182	Probe Request, SN=3, FN=0,
1186	1144.732677	76:b4:49:90:63:7d	Broadcast	802.11	190	Probe Request, SN=4, FN=0,

Table 6 depicts several examples of captured probe request packets from iPad mini, which were captured by Wireshark. The column represents, from left to right, the number of captured packet, the time of capture since beginning of capture (in second), the source address (MAC address), the destination address, the communication protocol, packet length, and packet information.

As we can see in Table 6, the iPad mini reset the Sequence Number (SN) after broadcasting two bursts of probe request. We can see in Table 6 that each burst, with nearly identical timestamp, is grouped in

color. The red boxes mark the **SN** reset, which mark a transition of **SN** from 78 to 0, and 19 to 0.

#### 4.1.2 Nexus 5X

As opposed to iPad mini, Nexus 5X only sent out randomized **MAC** address when the screen was off. When Nexus 5X woke up from sleep, it would immediately send out 4 or 10 probe request packets with real **MAC** address. When Nexus 5X screen went off, Nexus 5X firstly sent out random **MAC** address in every 2 to 10 seconds. Normally, Nexus 5X broadcast a burst of probe request packets roughly in each 60 seconds.

Unlike iPad mini, Nexus 5X changed the **MAC** address in each burst of probe request packets, as depicted in [Table 7](#). However, the first three octets were always the same and only the last three octets were changed. We can see that **da:a1:19** appears in multiple bursts in [Table 7](#) with different last three octets, **ce:47:5f**, **00:3f:25**, and **b5:51:2c**, respectively.

[Table 7](#): An example of captured probe requests from Nexus 5X in Wireshark. The colors mark out different bursts of probe request packets. The original **MAC** address is indicated by cyan color in the last burst.

No.	Time	Source	Destination	Protocol	Length	Info
141	124.338452	da:a1:19:ce:47:5f	Broadcast	802.11	124	Probe Request, SN=359, FN=0,
142	124.422657	da:a1:19:ce:47:5f	Broadcast	802.11	124	Probe Request, SN=361, FN=0,
143	124.464727	da:a1:19:ce:47:5f	Broadcast	802.11	124	Probe Request, SN=362, FN=0,
144	124.506760	da:a1:19:ce:47:5f	Broadcast	802.11	124	Probe Request, SN=363, FN=0,
145	124.591118	da:a1:19:ce:47:5f	Broadcast	802.11	124	Probe Request, SN=365, FN=0,
146	134.256859	da:a1:19:00:3f:25	Broadcast	802.11	124	Probe Request, SN=374, FN=0,
147	134.299121	da:a1:19:00:3f:25	Broadcast	802.11	124	Probe Request, SN=375, FN=0,
148	134.341256	da:a1:19:00:3f:25	Broadcast	802.11	124	Probe Request, SN=376, FN=0,
149	134.425444	da:a1:19:00:3f:25	Broadcast	802.11	124	Probe Request, SN=378, FN=0,
150	134.467394	da:a1:19:00:3f:25	Broadcast	802.11	124	Probe Request, SN=379, FN=0,
151	134.509631	da:a1:19:00:3f:25	Broadcast	802.11	124	Probe Request, SN=380, FN=0,
152	136.655150	da:a1:19:b5:51:2c	Broadcast	802.11	124	Probe Request, SN=391, FN=0,
153	136.696397	da:a1:19:b5:51:2c	Broadcast	802.11	124	Probe Request, SN=392, FN=0,
154	136.738584	da:a1:19:b5:51:2c	Broadcast	802.11	124	Probe Request, SN=393, FN=0,
155	136.822747	da:a1:19:b5:51:2c	Broadcast	802.11	124	Probe Request, SN=395, FN=0,
156	136.864483	da:a1:19:b5:51:2c	Broadcast	802.11	124	Probe Request, SN=396, FN=0,
157	136.906981	da:a1:19:b5:51:2c	Broadcast	802.11	124	Probe Request, SN=397, FN=0,
158	140.187428	LgElectr_ca:98:2a	Broadcast	802.11	136	Probe Request, SN=408, FN=0,
159	140.207811	LgElectr_ca:98:2a	Broadcast	802.11	136	Probe Request, SN=409, FN=0,
160	140.229742	LgElectr_ca:98:2a	Broadcast	802.11	136	Probe Request, SN=410, FN=0,
161	140.272010	LgElectr_ca:98:2a	Broadcast	802.11	136	Probe Request, SN=412, FN=0,
162	140.292256	LgElectr_ca:98:2a	Broadcast	802.11	136	Probe Request, SN=413, FN=0,
163	140.356324	LgElectr_ca:98:2a	Broadcast	802.11	136	Probe Request, SN=416, FN=0,
164	140.376401	LgElectr_ca:98:2a	Broadcast	802.11	136	Probe Request, SN=417, FN=0,
165	140.398346	LgElectr_ca:98:2a	Broadcast	802.11	136	Probe Request, SN=418, FN=0,
166	140.418463	LgElectr_ca:98:2a	Broadcast	802.11	136	Probe Request, SN=419, FN=0,
167	140.440200	LgElectr_ca:98:2a	Broadcast	802.11	136	Probe Request, SN=420, FN=0,
168	140.460526	LgElectr_ca:98:2a	Broadcast	802.11	136	Probe Request, SN=421, FN=0,

[Table 7](#) depicts an example of captured probe request packets from Nexus 5X in the same format as [Table 6](#). There are four bursts of probe request packets marked in different colors. Three of the bursts are random **MAC** address, while the last burst, marked in blue color, is the Nexus 5X original **MAC** address.

As we can see from [Table 7](#), the last probe request packet of a burst has a close **SN** to the first packet of the next burst, although it is not

sequential. We observed no obvious pattern in [SN](#) change during the experiment.

#### 4.1.3 Review

Based on our findings in [MAC](#) address randomization investigation, we can minimize the side-effect of [MAC](#) address randomization by using one minute cycle length in data collection. This means we record the ambient noise and capture time-lapse images within 1 minute time interval, while we capture probe request packets in 52 seconds followed by two times of [AP](#) scanning (see [Section 3.2.5](#)).

### 4.2 CORRELATION BETWEEN CROWD COUNT AND SENSOR READINGS

In this section, we present the result of data collection using 1 minute cycle length, which consist of probe request packet capture, [AP](#) scan, ambient noise recording, and time-lapse images capture (see [Figure 2](#)). As for the ground truth estimation, we used head count of time-lapse images and device count in unique [MAC](#) address in probe request. The results indicate whether smartphone sensor readings can be an approach to estimate the level of social density or crowd count.

We collected the data from Wednesday, October 26<sup>th</sup> 2016, to Saturday, October 29<sup>th</sup> 2016. The timing of data collection is shown in [Table 3](#). During the data collection, we observed no special events in which the social density level are much higher than the normal condition, especially in public places, such as Grote Markt and Paddepoel shopping center.

In manual head counting, a person might appear more than once, i.e., the person was captured in multiple GoPro cameras, because we took time-lapse images and some people were actively moving. We make sure that the people who appear several times in different images are counted exactly once. Furthermore, sometimes we captured vehicles with very limited visual appearance of the people inside. We assume that there are a person in a car and five people in a bus. We present some examples of time-lapse images in [Appendix B](#).

We present the important results of the experiment, such as the variation of the social density level, example of ambient noise recording, correlation of [AP](#) count and sensor readings, the effect of scanning time, and all parameters used for analysis. The appendices present the detailed results, such as the sensor readings both in scatter plot and line chart ([Appendix A](#)), example of time-lapse images for each location ([Appendix B](#)), and ambient noise recording ([Appendix C](#)).

[Figure 5](#) depicts the variation of the estimated social density level in each location, e.g., the lowest value, the first quartile, the median,

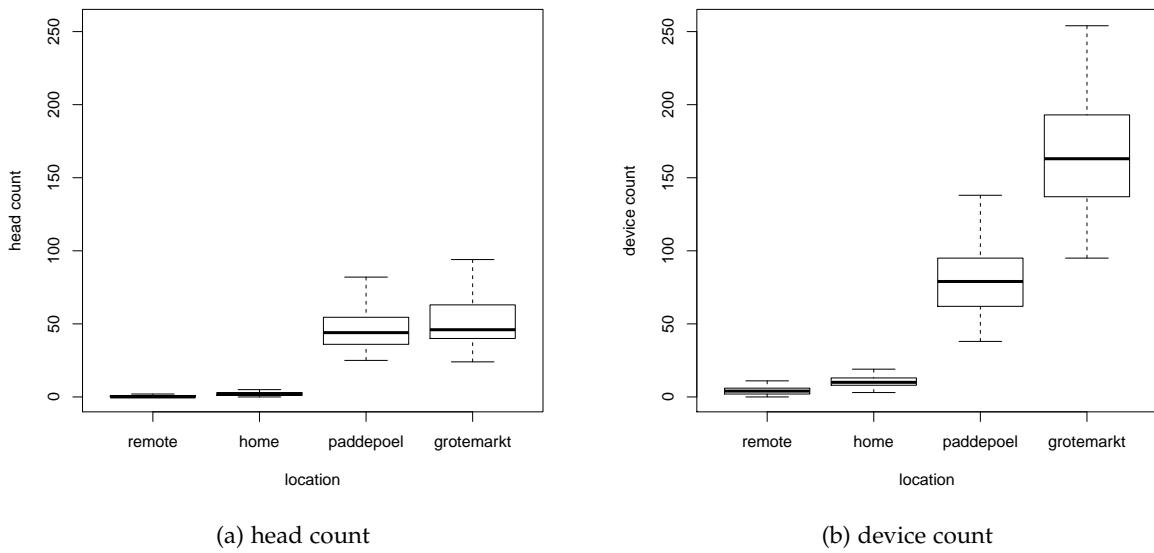


Figure 5: A box plot showing the population of estimated social density in each location by head count (5a) and device count (5b).

the upper quartile, and the maximum value. The value are estimated by two approximation, namely head count (5a) and device count (5b).

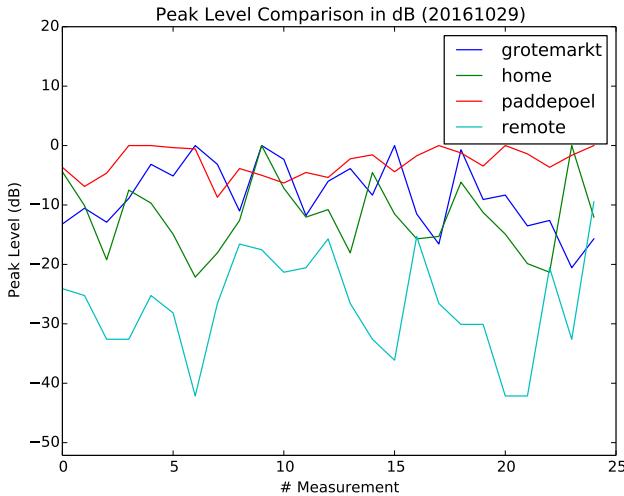
As we can see in Figure 5, the maximum value of the device count based estimation is higher than the head count based estimation. For instance, in Grote Markt, the maximum value of device count is 288, while the maximum value of head count is 115. However, both estimations are showing the same trend, which says that Grote Markt has the highest social density level and remote area has the lowest social density level.

#### 4.2.1 Ambient Noise Recordings

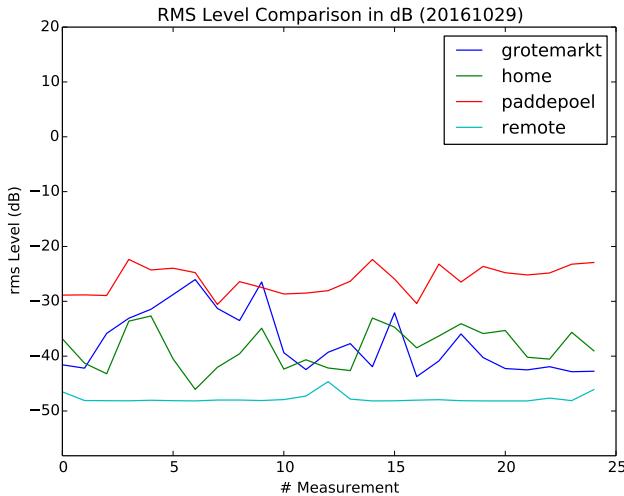
As an example of ambient noise recording, Table 8 and Figure 6 show the result of ambient noise recording in day 4, when more crowds were present than any other days. We measure the ambient noise in decibels (dB). In this unit, the closer the value to zero, the louder the sound (noise). We take two measures of ambient noise to characterize the surrounding, namely Peak Level (PKLV), which is the highest value of a total waveform, and Root Mean Square (RMS), which is the effective value or the mean of an audio recording. PKLV and RMS are positively correlated but RMS is more stable than PKLV.

Table 8: The average of RMS and PKLV of ambient noise recording in day 4.

	RMS (dB)	PKLV (dB)
Remote area	-47.74	-26.90
Home	-39.39	-12.84
Paddepoel	-26.54	-3.39
Grote markt	-36.29	-5.87



(a) peak level



(b) root mean square

Figure 6: Line chart showing the peak level (6a) and root-mean-square (6b) of the ambient noise recording in each cycle at day 4.

As we can see in the average value (Table 8) or line chart (Figure 6), more crowded place has higher ambient noise value. We see

that remote area is the quietest location, with -47.74 dB of average **RMS**, while Paddepoel is the noisiest location, with -26.54 dB of average **RMS**. As for the line chart, we can see more overlaps in **PKLV** ([Figure 6a](#)) than in **RMS** ([Figure 6b](#)). The **RMS** of remote area is stable near -50.00 dB. We can also see the same trends for the rest of the ambient noise recordings, although some overlaps are present, as shown in [Appendix C](#).

#### 4.2.2 AP and Social Density Correlation

We are particularly interested in seeing the trend and correlation of WiFi **AP** count and social density level, estimated by head count and device count. We draw a scatter plot of **AP** count vs device count and **AP** count vs head count separately, as well as device count vs head count to see the correlation of the two social density estimation. The plots are shown in [Figure 7](#), [Figure 8](#), and [Figure 9](#). We also present the line chart of the readings in [Appendix A](#).

[Figure 7](#) depicts the correlation of device count and **AP** count. The data plotted in [Figure 7](#) come from all data from four days of experiments. The location is coded in color so that we can distinguish which data come from which location. The plots of the correlation separately in each day are also available in [Appendix A](#), shown in [Figure 17](#).

As we can see in [Figure 7](#) there is a positive trend that indicates both variables have a positive correlation. This means when the **AP** count increases, the device count, which points to social density level, increases as well. The correlation coefficient  $\rho$  is 0.877, indicating a strong correlation, and the p-value is below 0.05, indicating that the result is significant. Each location forms a distinguishable cluster, although some overlaps are present. The cluster of home and remote are close and the cluster of Paddepoel and Grote Markt are adjacent. However, there is a gap that separates home-remote cluster and Paddepoel-Grote markt cluster.

[Figure 8](#) portrays the correlation between head count and **AP** count of data collected in four days of all location. [Figure 8](#) uses the same color coding as [Figure 7](#) to distinguish the location. The plots showing the correlation in separate days are available in [Appendix A](#), shown in [Figure 16](#).

We can see similar trend in [Figure 8](#). The correlation is strong, indicated by correlation coefficient  $\rho = 0.877$ , and significant, indicated by p-value which is below 0.05. However, the Grote Markt and Paddepoel clusters are overlapping each other. A gap between remote-home cluster and Grote Markt-Paddepoel cluster is also present.

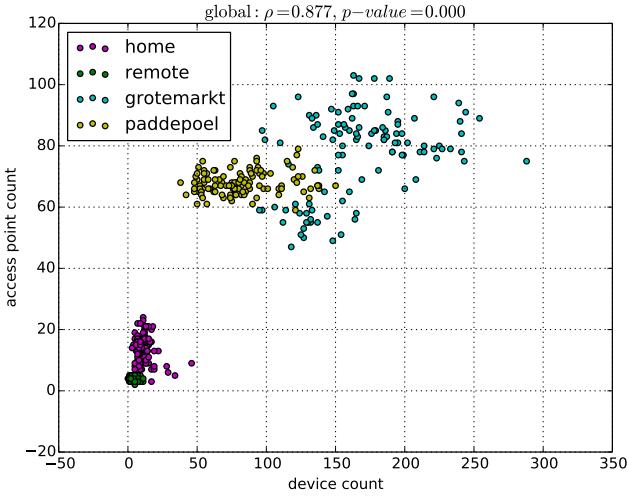


Figure 7: Scatter plot showing the correlation between *device count* and *AP count* of all collected data. The locations are marked in different color. The correlation coefficient is  $\rho = 0.877$ .

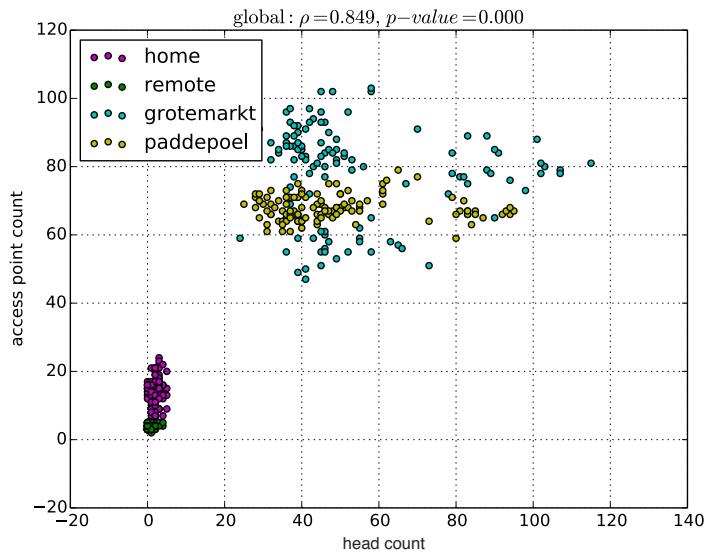


Figure 8: Scatter plot showing the correlation between *head count* and *AP count* of all collected data. The locations are marked in different color. The correlation coefficient is  $\rho = 0.849$ .

Figure 9 shows the correlation of head count and device count of all collected data. The locations are coded in color as well. Head count and device count have a strong correlation, marked by the correlation coefficient  $\rho$ , which is 0.857. A gap between remote-home cluster and Grote Markt-Paddepoel cluster is also present, although it is narrower than the gap in Figure 7 and Figure 8. Figure 18 in Appendix A presents the scatter plots showing the correlation of head count and device count in each day of the experiment.

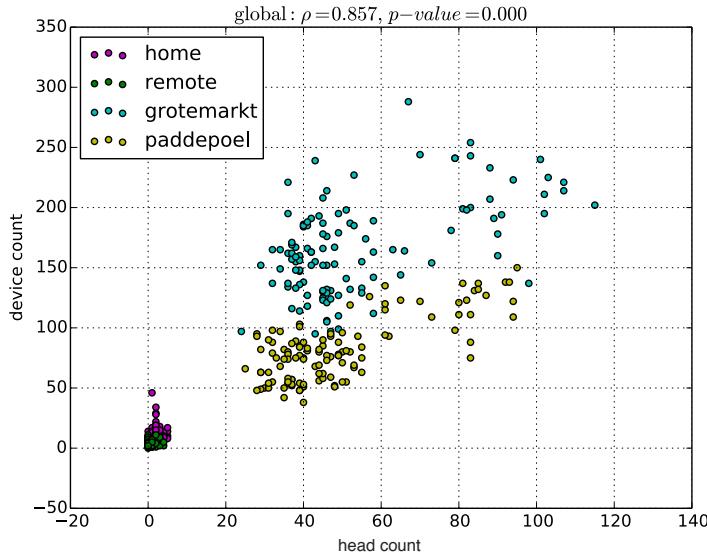


Figure 9: Scatter plot showing the correlation between *head count* and *device count* of all collected data. The locations are marked in different color. The correlation coefficient is  $\rho = 0.857$ .

As we can see in three scatter plots (Figure 7, Figure 8, and Figure 9), AP and social density level from both head count and device count have a positive correlation. The correlation is strong, which is roughly 0.8 with p-value less than 0.05.

#### 4.2.3 Effect of Scanning Time

We performed four experiments on Sunday, October 30<sup>th</sup> 2016, to investigate the effect of scanning time to the AP count and social density correlation. We collected the data for 45 minutes, consisting WiFi and audio data, in four different scanning time, namely **a)** 09:00, **b)** 12:00, **c)** 15:00, **d)** and 18:00.

Figure 10 shows the WiFi scanning result in four separate line charts. Each chart depicts the result of a certain scanning time. The blue lines, which depict the AP count, fluctuate stably around 100, while the green lines, which represent the device count, have a big variance. As we can see in Figure 10a, the green line is fluctuating below the blue line, indicating there were less device than available AP count. However, in Figure 10b, Figure 10c, and Figure 10d, the condition changes. The green lines surpass the blue lines, which indicates that the number of device was increased. Figure 10c shows an increasing trend of device count, while Figure 10d shows a decreasing trend.

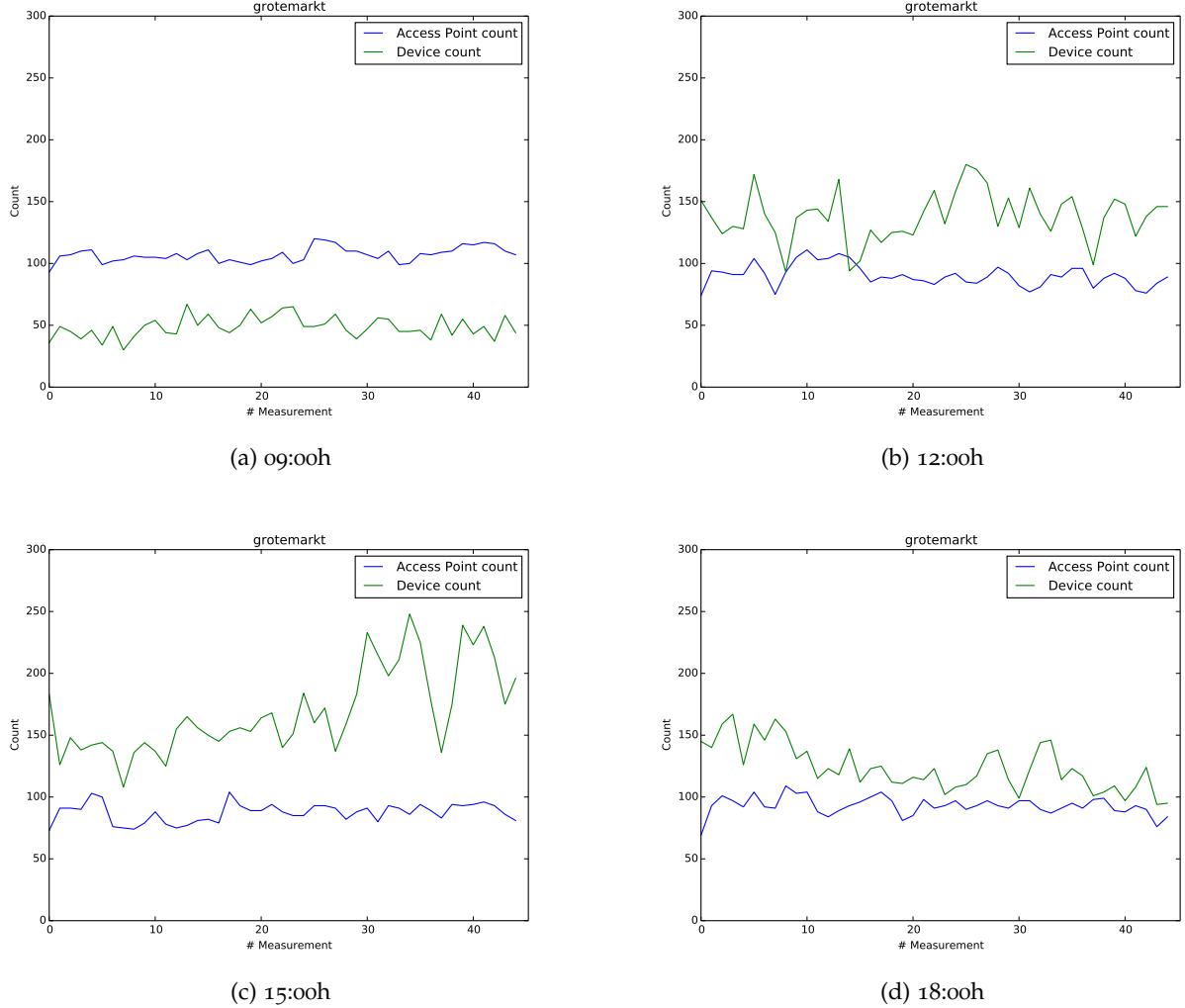


Figure 10: Line charts showing the AP count (blue) and device count (green) in different data collection time at Grote Markt, Groningen, on October 30, 2016.

Moreover, we present the ambient noise recording of the Sunday, October 30<sup>th</sup> 2016 experiment in [Table 9](#) and [Figure 11](#). If we look at the line charts ([Figure 11](#)), the lines are mostly overlapping each other, making it difficult to distinguish the noise characteristics of each scanning time. However, the ambient noise summary in [Table 9](#) presents more understandable characteristics. We see that, in [PKLV](#), the noise is increasing from 09:00h, peaking at 15:00h, and going back down at 18:00h. The ambient noise recordings are correlated with the WiFi scanning results presented in [Figure 10](#).

Table 9: The average of root-mean-square and peak level of ambient noise recording in four different scanning time.

	RMS (dB)	PKLV (dB)
09:00h	-40.51	-14.95
12:00h	-40.91	-13.08
15:00h	-37.53	-8.90
18:00h	-40.62	-12.41

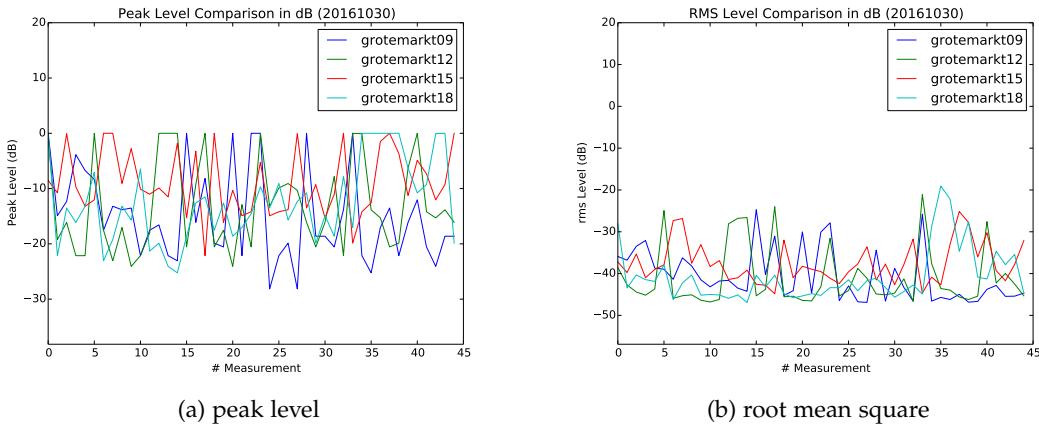


Figure 11: Line charts showing the peak level (11a) and root-mean-square (11b) of the ambient noise recording in four different scanning time.

#### 4.2.4 All Result

Figure 12 shows all data collected in four days of experiments in a scatter plot matrix. In the lower left part are the scatter plots of each parameter correlation along with the Lowess curve fitted to the data. In the upper part are the correlation coefficient and the p-value. Along the diagonal are the histograms and the labels of each parameter.

There are eight parameters extracted from all datasets of four days experiment. The AP, RSSI, Device Count (DC), and Signal to Noise Ratio (SNR) parameters are from WiFi based readings. AP is the count of available AP that a smartphone can get, while RSSI is the average of the signal strength of the AP. DC is the device count, based on unique MAC address, and SNR is the average of signal-to-noise ratio of the captured probe request packets. Speaker Count (SC), RMS, and PKLV are from recorded ambient noise, while Head Count (HC) is from manual head counting of time-lapse images.

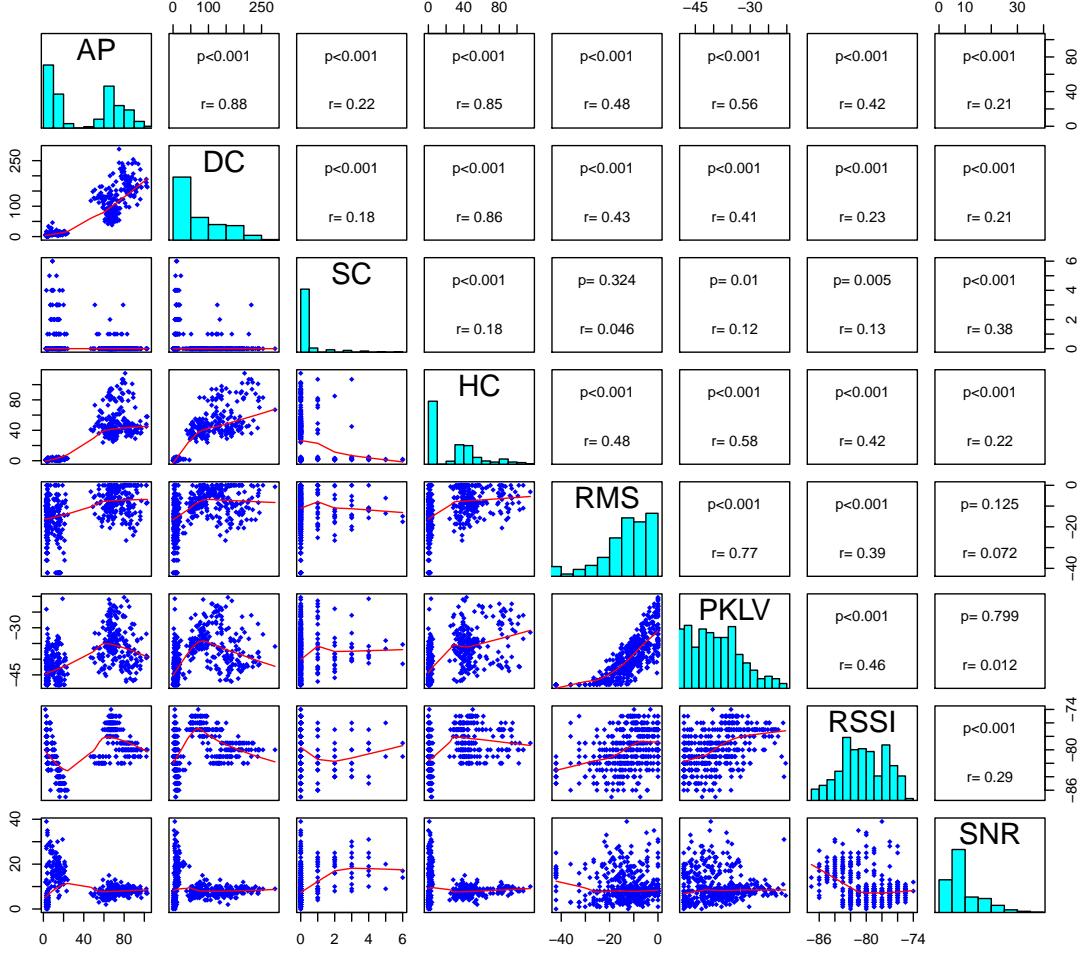


Figure 12: Scatter plot matrix showing all parameters correlation of all collected data. The top right shows the correlation coefficient  $r$  and the  $p$ -value. Along the diagonal are the histograms of each parameter.

#### 4.3 SOCIAL DENSITY PREDICTION

We construct data models so that we are able to predict the level of social density from new data. We use supervised learning techniques using three predictors, namely linear regression, k-Nearest Neighbor (**k-NN**), and Support Vector Machine (**SVM**), so that we can compare which predictor gives optimal model.

We predict the head count or device count from other parameters, e.g., **AP**, **RMS**, **PKLV**, and **RSSI**. We do not use **SC** as a predicting variable because speaker count value is almost at zero, which does not give us any meaningful insight, and we do not use **SNR** parameter as well, because we derive signal to noise ratio from probe request packet

capture in a laptop, which smartphone cannot capture. Our dataset contains 459 records.

We validate our model using 10-folds cross-validation, which is a method to assess the performance of a prediction model. The cross-validation technique involves dataset partition into complementary subsets, namely training and testing sets, in which training subsets are for performing the analysis, while testing subsets are for the validation of the resulting model. 10-folds cross-validation divides the dataset to 10 complementary subsets, in which one subset will be the testing set and the other nine subsets are the training set, interchangeably in 10 times.

We use **RMSE** as the metric for cross-validation. In this metric, the optimal model has lower **RMSE** value. The formula of **RMSE** is as follows,

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (p_i - a_i)^2}{n}} \quad (1)$$

where  $p_i$  is the predicted value,  $a_i$  is the actual value, and  $n$  is the number of cases. We implement the analysis using R [29], presented in [Appendix D](#).

#### 4.3.1 Linear Regression

Linear regression is a method of modeling the relationship between a dependent variable and an explanatory (or predicting) variable by fitting a straight line across the data. A condition where there is only one explanatory variable is called simple linear regression, while multiple linear regression involves more than one explanatory variable. The result of linear regression is a linear function (or a model) with the explanatory variables as the parameters.

To obtain a model with minimal error, we perform an exhaustive search for the best subsets of the predictors for predicting the dependent variable (head or device count) in linear regression. We use **RMSE** of 10-folds cross-validation to select the optimal model using the smallest value. [Listing 1](#) displays the implementation of linear regression analysis. We implement linear regression in R using **caret** library [20].

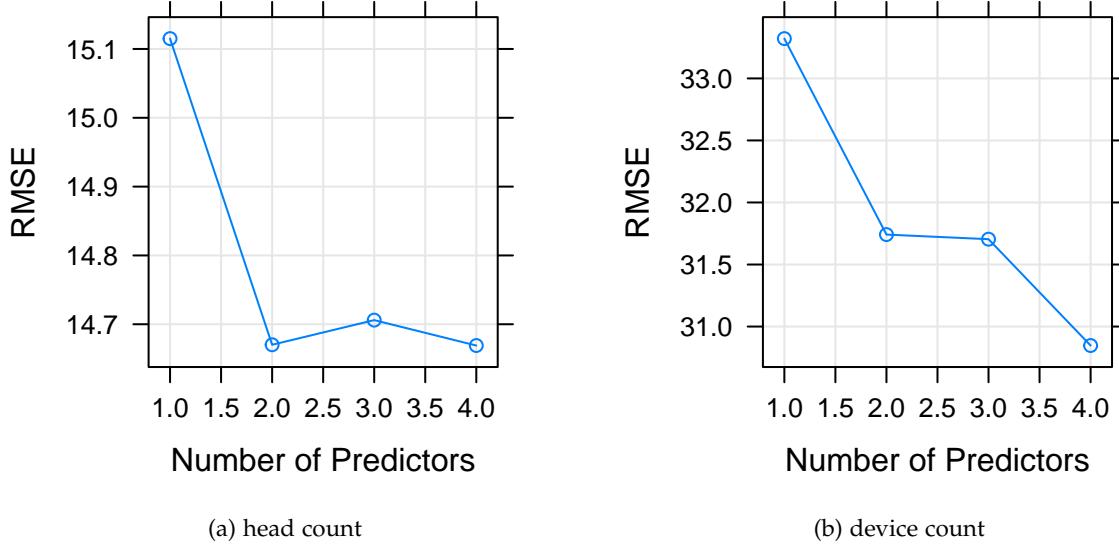


Figure 13: Tuning linear regression using best subsets combination for head count (13a) and device count (13b). The result indicates that using all predictors gives the best result, although in head count prediction using two predictors results in nearly the same error.

Figure 13 presents the tuning result. For both head count and device count, using all four predictors results in optimal model. In head count prediction, using two predictors (ap and pkv) yields in nearly identical RMSE as using four predictors. Based on the tuning, we develop a linear model to predict the head count and device count. The linear regression fit to the dataset is depicted in Table 10.

Table 10: Linear model fit to the dataset with corresponding coefficient for head count and device count model.

Term	Coefficient	
	Head count	Device count
Intercept	54.675	-367.901
ap	0.653	2.021
rms	-0.109	1.138
pkv	0.760	-1.880
rssi	0.329	-3.656

### 4.3.2 *k*-Nearest Neighbor

*k*-Nearest Neighbor ([k-NN](#)) is an algorithm that predicts numerical value based on similarity measure using distance function of *k*-nearest neighbors of the predicted value.

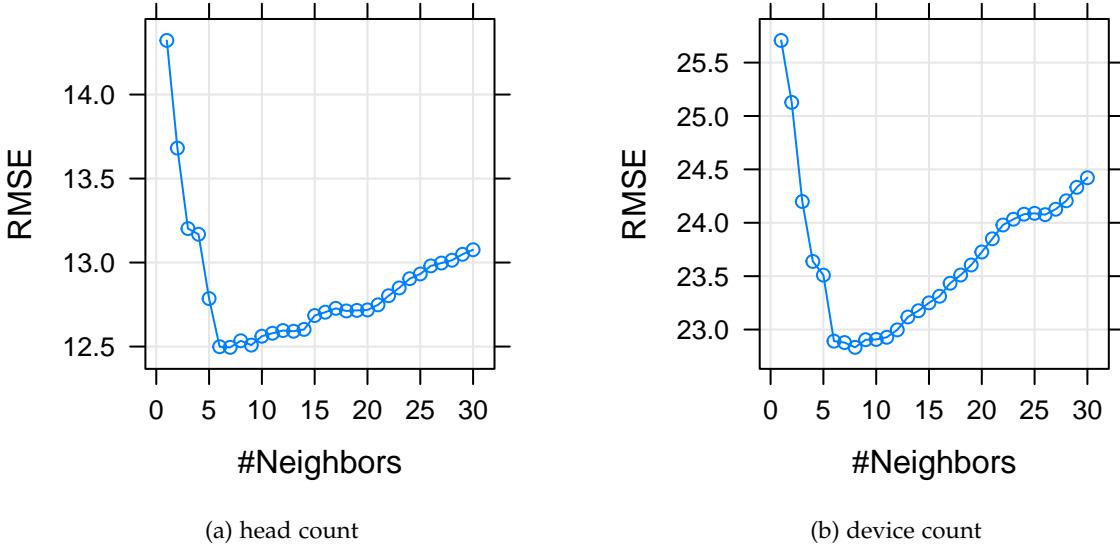


Figure 14: Tuning [k-NN](#) using  $1 \leq k \leq 30$  as the tuning parameter (#Neighbors) for head count (14a) and device count (14b). For both estimation, optimal result is obtained when  $5 \leq k \leq 10$ . The optimal result is chosen at  $k = 7$  (head count) and  $k = 8$  (device count).

We train and test [k-NN](#) regression using  $1 \leq k \leq 30$  as the tuning parameter in R using [caret](#) library [20]. Similar in the linear regression analysis, we evaluate the model using 10-folds cross-validation and we use [RMSE](#) to select the optimal model with lowest error. Listing 2 displays the implementation of [k-NN](#) regression.

Figure 14 presents the tuning result. According to Figure 14, the highest error of both estimation is when  $k = 1$  and there is an increasing trend of error as  $k$  goes up. The tuning results indicate that the optimal model for head count estimation is achieved when  $k = 7$  ( $\text{RMSE} = 12.49515$ ), while the optimal model for device count estimation is when  $k = 8$  ( $\text{RMSE} = 22.83321$ ).

### 4.3.3 Support Vector Machine

Support Vector Machine ([SVM](#)) is an estimator that works by constructing support vectors (hyperplanes) that separate the data according to a certain threshold. Most of the time, a kernel, a set of mathematical functions, is used to define the decision boundary.

We implement **SVM** to predict quantities using radial kernel with cost and  $\epsilon$  as the tuning parameters. cost parameter is used to define the width of the hyperplane, while  $\epsilon$  is used to determine how non-linear the decision boundary is. As used in **k-NN**, we validate the model using 10-folds cross-validation. We use grid search method to tune the optimal parameters. It tries to find the best combination of the two parameters in certain range of search. We implement the analysis in R with e1071 library [26]. Listing 3 displays the implementation of **SVM** prediction.

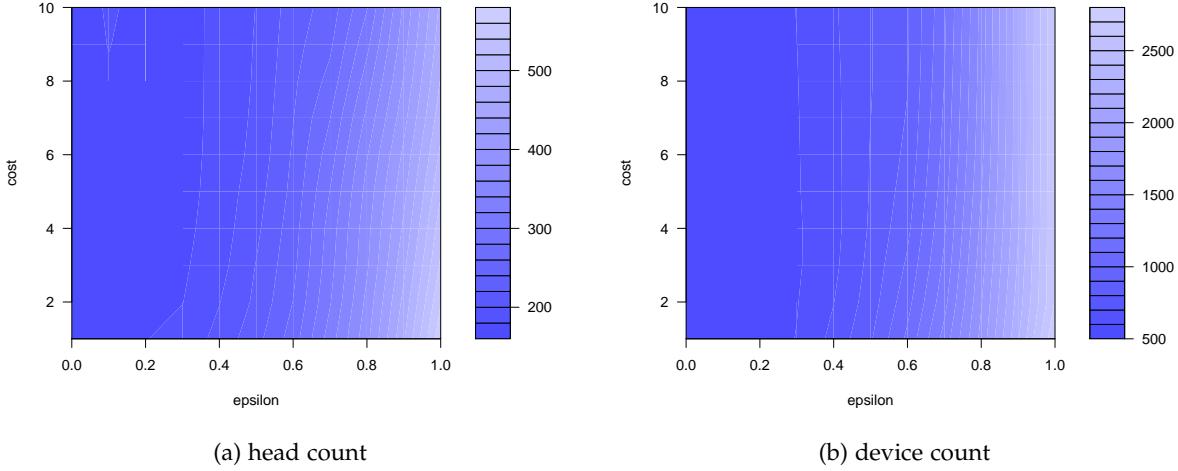


Figure 15: Tuning **SVM** using grid search technique. Two tuning parameters  $\epsilon$  and cost are represented in x and y axis, while the performance (measured using **MSE**) is represented in color. More optimal model is indicated by darker color, i.e., lower **MSE** (see color bar).

We present the result of **SVM** regression tuning in Figure 15. We focus the grid search within  $0 \leq \text{cost} \leq 10$  and  $0 \leq \epsilon \leq 1$  to reduce the computational time. Moreover, our previous grid search, which involves wider area, indicates that **SVM** performs better within that area.

We measure the performance of the **SVM** using mean squared error (**MSE**), in which best result has lower **MSE**. The best performance for head count prediction is at 171.1065 with  $\epsilon = 0$  and  $\text{cost} = 3$ , while for device count prediction is at 514.226 with  $\epsilon = 0$  and  $\text{cost} = 1$ . We can also see a trend of increasing **MSE** as the  $\epsilon$  increases.

To summarize the regression analysis and to interpret the result, we use another approach to calculate the error, which has more intuitive interpretation. We use residual error, which can be formulated as

$$\overline{\text{error}} = \frac{1}{n} \left( \sum_{i=1}^n |p_i - a_i| \right) \quad (2)$$

As formulated in the equation, residual error shows the mean of the difference between predicted and actual value. Using the optimum model of each regression technique, we perform another 10-fold cross validation to obtain residual error measurement. The optimal model has a residual error closer to zero.

Table 11: Summary of regression analysis showing the range of the data and the residual error of each regression method.

	Value Range		Residual Error		
	Min	Max	Linear	k-NN	SVM
Head count estimation	0	115	9.89	7.05	7.06
Device count estimation	0	288	23.81	13.14	13.34

**Table 11** presents the summary of the regression analysis. **k-NN** and **SVM** have better performance than the linear regression. **k-NN** and **SVM** do not have much difference of error for both head count and device count estimation. Device count estimation has wider value range and error compared to head count estimation.



# 5

## DISCUSSION

---

We discuss our findings of the experiments in the following topics, ambient noise recording, ground truth approximation, scanning time effect, sensor readings correlations, the implications of the data modeling, and the limitation of the present study.

### 5.1 AMBIENT NOISE RECORDING

We recorded the ambient noise and extracted the peak-level ([PKLV](#)) and root-mean-square ([RMS](#)) of the recordings, which are highly correlated, with  $\rho = 0.77$  (see [Figure 12](#)). Initially, we expected to see a strong correlation of ambient noise and the level of social density, i.e., the location which has high level of social density also has high value of ambient noise, but the result says otherwise. We can see in the scatter plot matrix of the dataset, [Figure 12](#), that only the correlation of head count and peak level which is more than 0.5 ( $\rho = 0.58$ ). The other correlations of social density levels (head count or device count) and ambient noise (peak level or root-mean-square) are below 0.5.

In the scatter plot of [Figure 12](#), we can see that some of the low social density values (less than 20) have high ambient noise value as well (more than -20dB), which means we also observed more noise in the area where less crowds were observable. However, we can also see in [Figure 12](#) that no high social density values (more than 50 for device count and 20 for head count) are below -30dB, which means that high social density areas have high amount of noise. We conclude that high values of ambient noise mostly indicate a high level of social density. The line charts showing the peak level and root-mean-square of the ambient noise of all locations ([Figure 6](#), [Figure 39](#), [Figure 40](#), and [Figure 41](#)) also support our conclusion. The graphs show that more crowded locations (Grote Markt and Paddepoel) have higher peak level or root-mean-square value than (Home and remote area) less crowded locations although some overlaps exist.

Furthermore, microphone sensitivity also affects the result of ambient noise recording. We used a laptop's built-in microphone to record the ambient noise. This kind of microphones (and ones installed in smartphones) is attuned to a specific (and rather narrow) range of sound intensity.

## 5.2 GROUND TRUTH APPROXIMATION

We estimate the crowd count as a first approximation of the ground truth because it is known that getting the ground truth of crowd density in public spaces is difficult [41]. We used time-lapse image based, which works by manually counting heads in the images, and WiFi's probe request based technique, works by counting unique **MAC** addresses.

Although probe request based estimation is promising, some drawbacks are also present. In this method, we are not able to distinguish the type of devices, i.e., whether it is a smartphone, tablets, or computers. Although one mostly brings a smartphone in crowded public areas [45], which means we can deduce that a smartphone means a person present, there is also a possibility that one brings more devices or no devices at all.

Furthermore, the WiFi based technique is able to detect devices through walls, which is good or bad depending on the definition of social density. In indoor social density estimation, this is a bad approximation because WiFi might detect some people but in fact there are less or no people at all inside a particular room, as they are located in other rooms nearby. This is a potential threat, for instance, if our method detects some people but actually no one is present in the room. However, in outdoor social density estimation, WiFi based technique performs better than image based technique, as WiFi can see people through an obstacle, while image-based cannot. To sum up, we have to be careful in doing indoor monitoring. As an option, we can also combine the social density estimation with ambient recording to see whether there is a noise inside a room, as empty room is mostly quite.

Compared to the probe request based estimation, time-lapse image based technique cannot detect people through walls or buildings. This might be the reason why the image based technique detects less people than the WiFi based technique (see [Figure 5](#)). Furthermore, image based technique relies heavily on assumptions when vehicles, e.g., buses or cars, are captured in the image, as vehicles have very limited visual appearance of the people inside. We assume that there are a person in a car and five people in a bus. This assumptions may slightly bias the result.

## 5.3 SCANNING TIME EFFECT

We tested the effect of scanning time to investigate whether scanning time can potentially affect the outcomes. Interesting findings about the scanning time are shown in [Figure 10](#). We see that the device count in each scanning time has different outcome, while the **AP** count remains stable no matter when the scanning was performed.

When we scanned the surroundings in the morning (09:00h), the device count was less than the AP count, as there were not so many people present. However, the trend changed when we performed the scanning at 12:00h, as the device count surpassed the AP count. We can also see an increasing trend at scanning time performed at 15:00h and decreasing trend at 18:00h. If the trend continues, there might be lower device count than AP count at night. The findings presented in [Figure 10](#) indicate that we also have to consider the scanning time.

#### 5.4 CORRELATION BETWEEN SENSOR READINGS AND SOCIAL DENSITY

The parameter which has a strong correlation with head count or device count is the AP count, as shown in the scatter plot matrix ([Figure 12](#)). The correlation coefficients of AP count vs device count, AP count vs head count, and device count vs head count are 0.87, 0.85, and 0.86, respectively. We present in detail the correlation of head count, device count, and AP count in [Section 4.2.2](#).

As we can see in the scatter plots of the correlation of AP, DC, and HC, presented in [Figure 17](#), [Figure 16](#), and [Figure 18](#), the lower social density data, as at home or remote area, seems to be more concentrated than higher social density data, as at Paddepoel or Grote Markt, and the higher social density data is widely spread. However, this does not mean that the frequency of high social density values is bigger than the low social density value. In fact, we have more data for lower social density than higher social density, as shown in the histograms of AP, DC, and HC in [Figure 12](#).

If we look at [Figure 16](#), [Figure 17](#), and [Figure 18](#), the data may also be used to perform a sort of localization, as each location shows different patterns. We can say that the AP count is bound to an area, where each area has different social density levels. Thus, this method is somewhat telling where the smartphone actually is and what the average of social density level in that area is. On the other hand, we could infer where the location of the smartphone is using our proposed technique.

In the beginning, we expect that the number of AP follows the fluctuation of the number of people in a certain location, as people might bring their own portable WiFi transmitter to assemble an ad-hoc AP, which will add up the number of AP available in the location. However, this turned out to be a rare situation, as we do not see any trend between number of people and AP count, as shown in line chart of the experiment presented in [Section A.2](#).

Moreover, if we look at the line chart, there is a fluctuation of AP count, although actually available AP count should be the same or stable across the time. This fluctuation might be caused by the instability

of radio transmission that may affect the signal strength of the WiFi AP and thus making some APs some not detected.

### 5.5 SOCIAL DENSITY PREDICTION

We have created several models to predict the social density using new data. The optimal model was achieved using k-NN predictors, although it actually performs equally well compared to SVM method. However, as opposed to linear model, k-NN works better because k-NN works by measuring the similarity of k-nearest neighbors of the predicted value, while linear model only fitting a straight line across the data. In fact, the correlation is not always perfectly linear.

The optimal model gives an residual error equals to 7.05 for head count and 13.14 for device count. The error means that, when the model is used to predict new head count data, the predicted value may be 7.05 higher or lower than the actual value, or 13.14 for new device count data. For instance, if our model predict 50 head count, it may be actually 57.05 or 42.95.

Moreover, compared to the value range of the head count or device count, which range from zero to 155 (head count) and zero to 288 (device count), the errors count 6.1% for head count and 4.6% for device count. In summary, the model is able to determine if the subject is currently in a considerably low or high social density area.

### 5.6 LIMITATIONS OF THE PRESENT STUDY

The proposed method has several limitations. The first limitation is the location. The proposed method only works at places where WiFi AP use is not restricted, i.e., people are free to set up their own WiFi AP. Some locations where the use of WiFi is restricted exist, for instance, University of Groningen complex, where eduroam is the only the available AP and the occupants are not allowed to install their own AP. If we collect data from this location, we will possibly get uncorrelated AP count and social density.

Furthermore, our dataset consists of data collected from four different location. This dataset represents the situation of selected locations but possibly not for other locations, as other locations may have different characteristics. For instance, in developing countries or other locations where the use of WiFi are not common, the number of AP in a public area may be much lower than what we have in our dataset.

As we mainly work with WiFi, the range of the proposed social density follows the maximum WiFi coverage for smartphones, which extends roughly from 20 to 50 meters. This range may be good or bad depending on the context. For outdoor situations, this range gives a good approximation as it is considerably broad enough to count social density. However, this range may be too broad for indoors. For

instance, we may get a result that says there are 20 people in total, but in fact there is only five or even less people in the room. This is because WiFi also detects people outside the room. However, we can also combine the result with ambient noise recording, as empty room are usually more quite.

The other limitation is time. Our dataset consists of data collected in daytime, ranging from 08:15h to 14:45h. Using this dataset, we are only able to approximate the level of social density if the new data is captured during the same time frame. If the new data is taken outside the time frame, our dataset is unable to tell the level of social density. This fact is based on the time of scanning investigation described in [Section 3.2.4.1](#) and [Section 4.2.3](#).

In conclusion, the results indicate that locations with high level of social density tend to have more access points. WiFi AP shows the strongest correlation with the social density level, followed by RMS that reveals a weaker correlation with social density level. Thus, we can say that it is possible to infer social density level from smartphone sensor readings. To achieve better accuracy, we may implement classification analysis instead of regression. However, regression analysis yields class-based result instead of continuous numerical result.

Furthermore, generalization of this method requires further investigation in more locations and time, as other location may reveal different patterns. Scanning time result also says that scanning time affects the social density level. Moreover, the settings and results will vary in different cities and even more in different countries.



# 6

## CONCLUSION AND FUTURE WORK

---

We have studied how a consumer smartphone can be utilized as an instrument to estimate the level of social density in the surroundings of its owner. We collected the data at several locations to see the trend of sensor readings and social density level. We also try to construct a model to predict upcoming social density levels based on sensor readings. To conclude this thesis, we go through the research questions and answer the question based on our findings. As posed in [Chapter 1](#), the main research question is

*How can we estimate the level of social density of the surroundings using smartphone as a passive behavioral monitoring scheme?*

As presented in [Chapter 2](#), literature study shows that smartphone sensors are exploitable to estimate the social density level of the surroundings. We also conducted experiments, described in [Chapter 3](#), to inspect the hypothesis. The results, presented in [Chapter 4](#), show that there is a trend that follows the relation of sensor readings and social density levels. We develop prediction models to predict the social density level in [Section 4.3](#) and we also discuss the findings and the limitations in [Chapter 5](#).

The sub-questions are defined to help answering and validating the main research question. Those were formulated as follows:

1. *What sensors are available at consumer smartphone and which of those are favorable for the social density estimation method?*

We present a summary of available sensors in smartphone in [Table 2](#). In this thesis, we make use of WiFi and microphone to record data for estimating social density level. We select these sensors as those are usable no mater how the smartphone is operated and positioned.

2. *How can we validate the sensor readings, meaning, to get the ground truth or the approximation of the ground truth?*

We use manual head counting in time-lapse images and unique [MAC](#) address counting in captured WiFi probe request packets as first approximation of ground truth. We select these techniques as those are demonstrated to have a correlation with social density in the surroundings.

3. *Can this method work everywhere? What is the scope of this method?*

Our proposed method works well at the selected locations of the experiment presented in [Table 3](#). Our method is applicable in

locations where WiFi AP use is not restricted, i.e., people are free to set up their own WiFi AP in anyway they prefer. Furthermore, our method is bound to time constraint, which limits to daytime implementation when the social density level reaches its peak. Generalization of this method requires further data collection.

### 6.1 FUTURE WORK

We plan to collect more data so that we can generalize the proposed method. The data collection includes more locations with several scanning time to see the variance of social density in time, adding more features for analysis. Furthermore, we are also interested in doing classification analysis to see whether we can achieve better performance of social density estimation. However, classification analysis means that the result will be classified into several classes instead of continuous numbers.

Moreover, we plan to enrich the smartphone data by the inclusion of other data types, for instance, GPS location and nearby Bluetooth signals. Different data type inclusion may result in better model of social density information, so that we can achieve more accurate results. Other resources or instruments may refer to using dedicated device to monitor the social density level or other data coming from third party stakeholders such as cellular operators.

Part II  
APPENDIX



# A

## ACCESS POINT COUNT CORRELATION

### A.1 SCATTER PLOTS

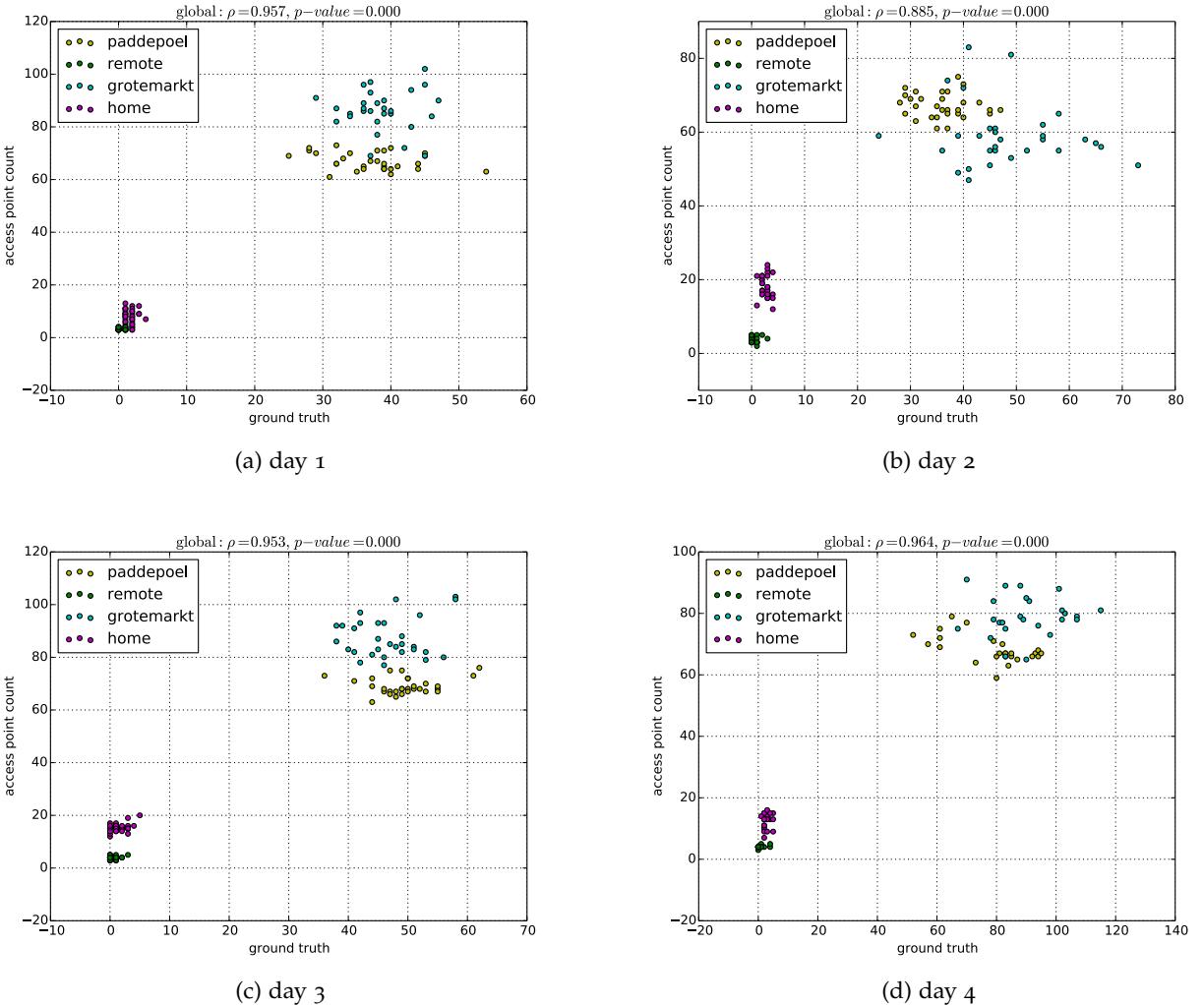


Figure 16: The scatter plots showing the correlation between headcount and AP count in four days of experiment. The location is coded in color.

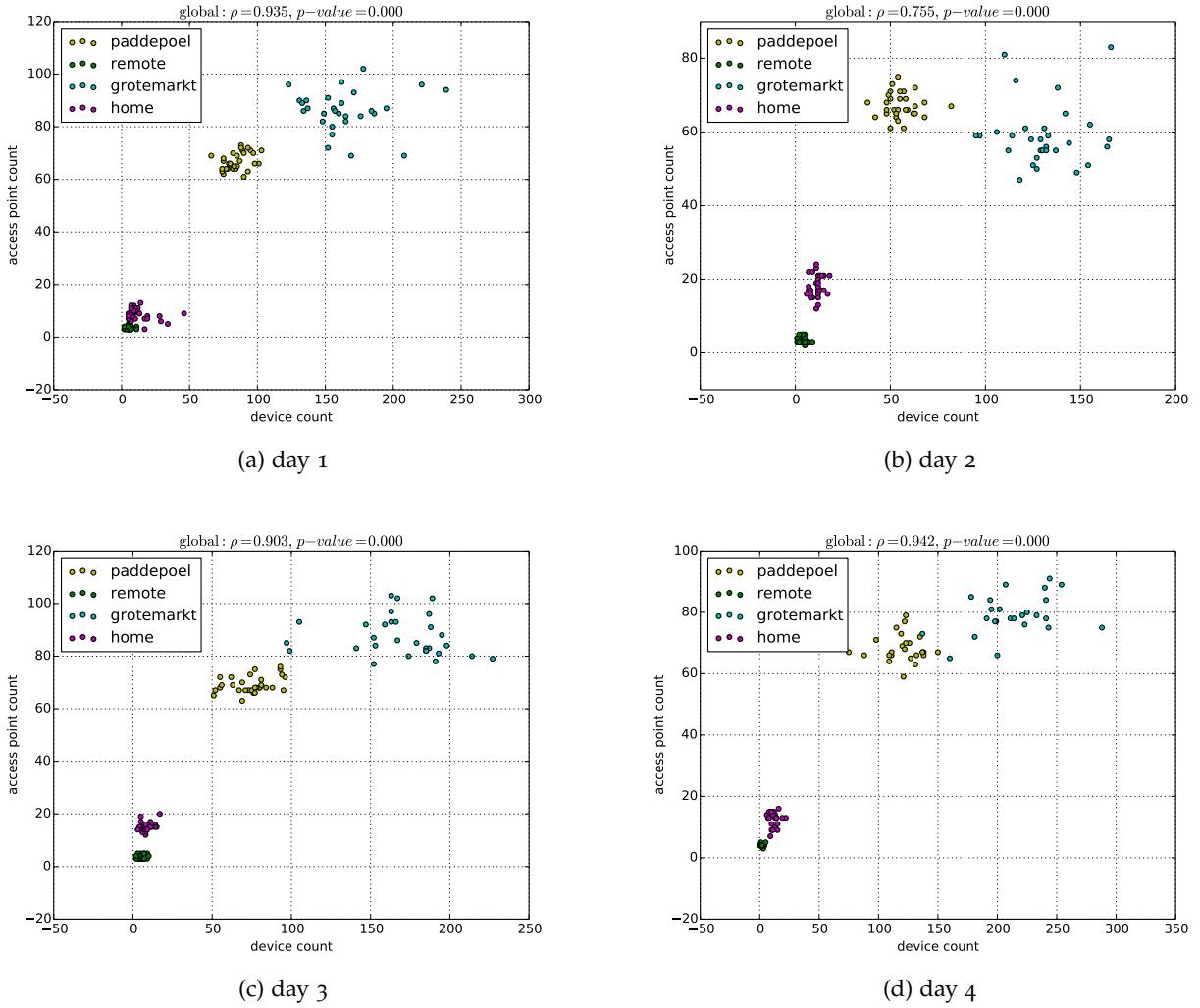


Figure 17: The scatter plots showing the correlation between device count and AP count in four days of experiment. The location is coded in color.

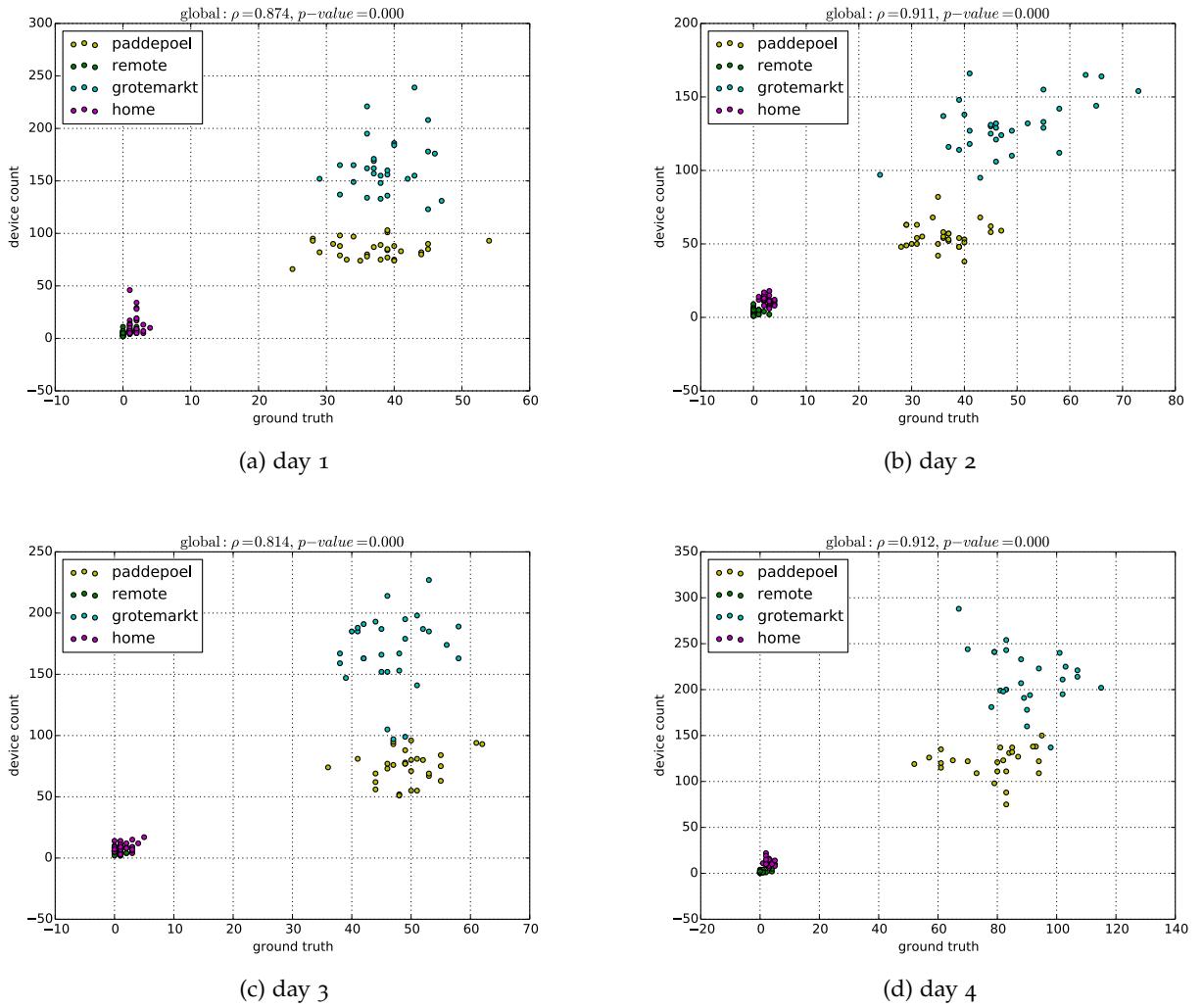


Figure 18: The scatter plots showing the correlation between device count and AP count in four days of experiment. The location is coded in color.

## A.2 LINE CHARTS

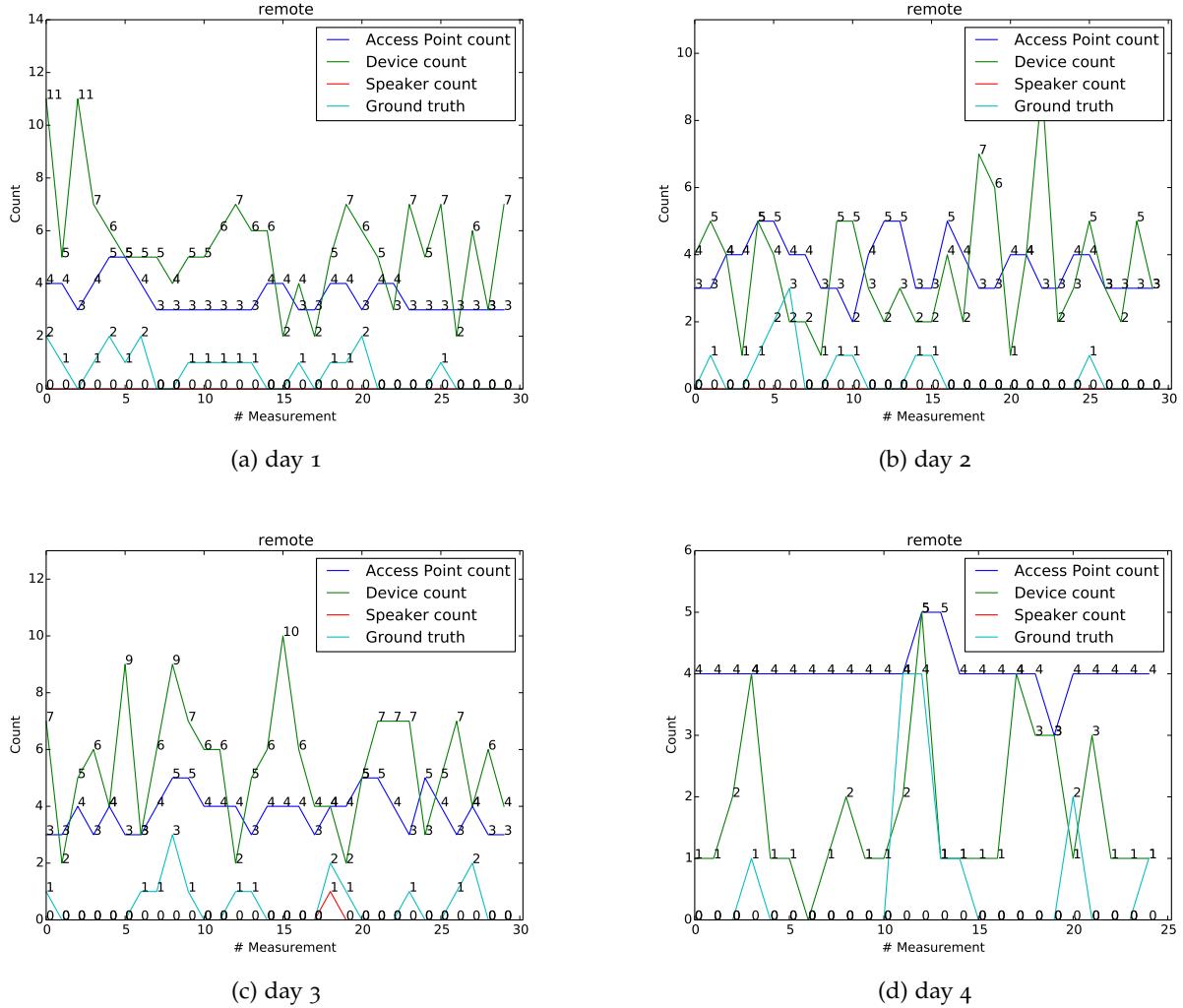


Figure 19: The line chart of sensor readings at *remote area* in four days of experiment.

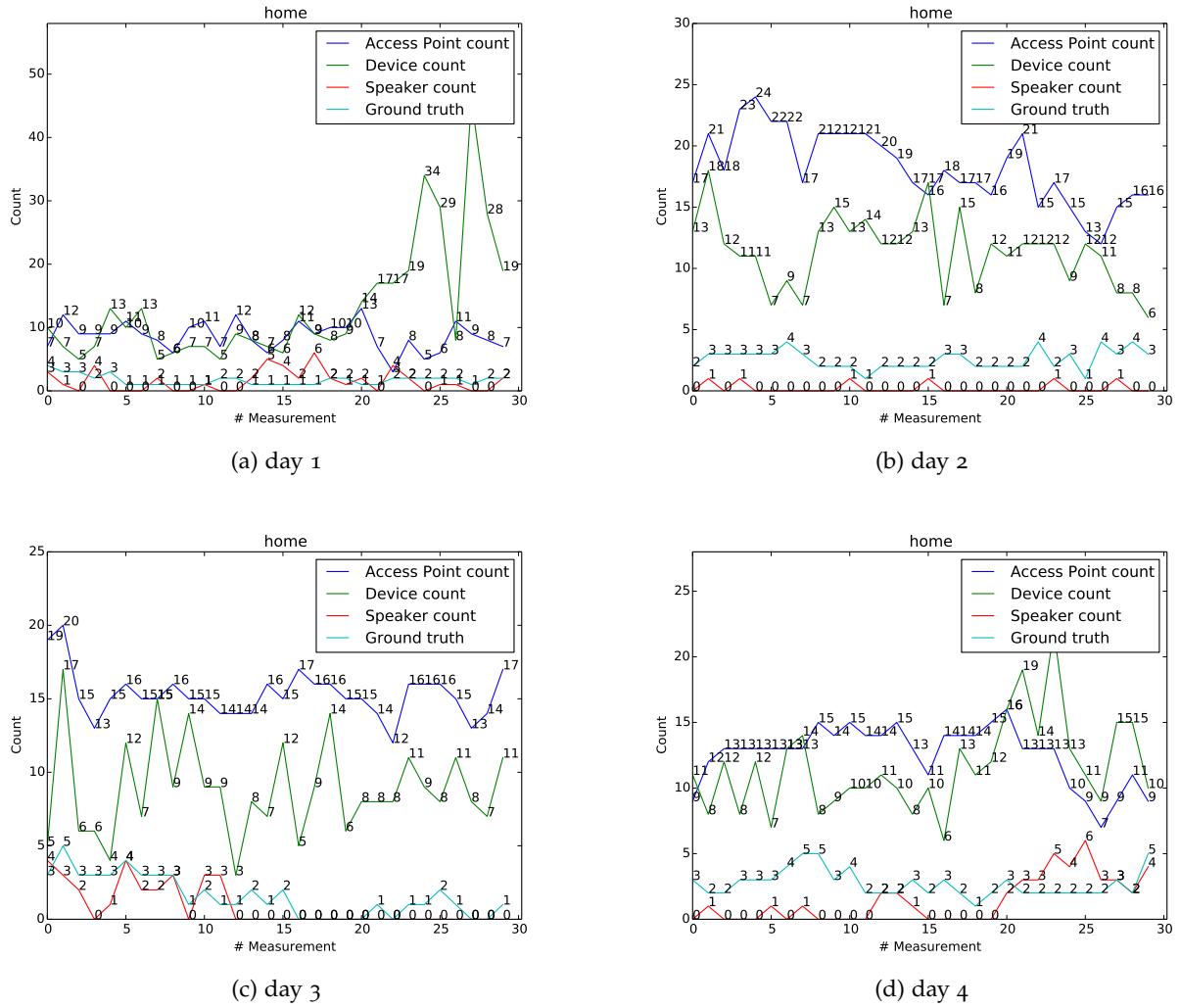


Figure 20: The line chart of sensor readings at *home* in four days of experiment.

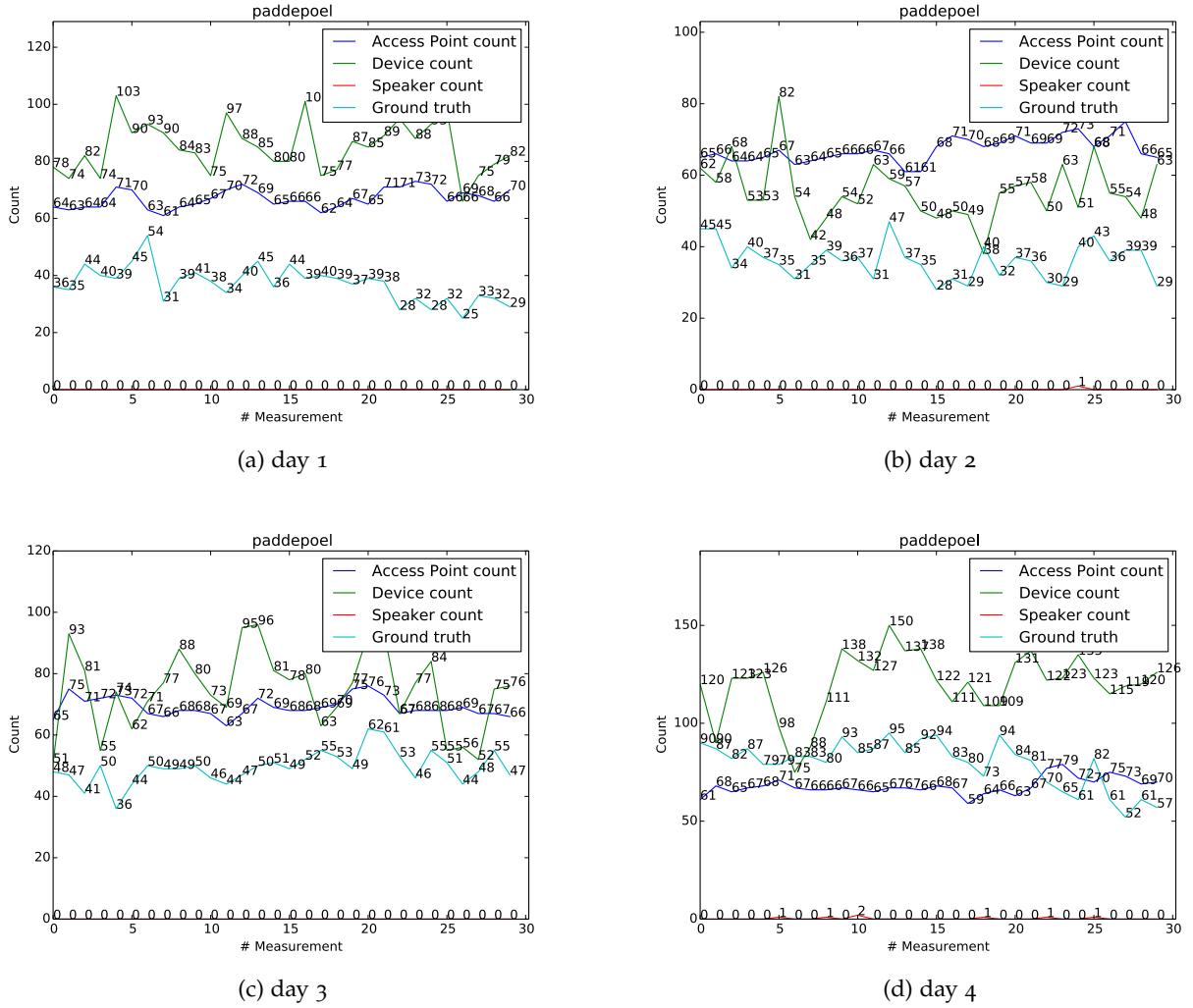


Figure 21: The line chart of sensor readings at *Paddepoel Shopping Center* in four days of experiment.

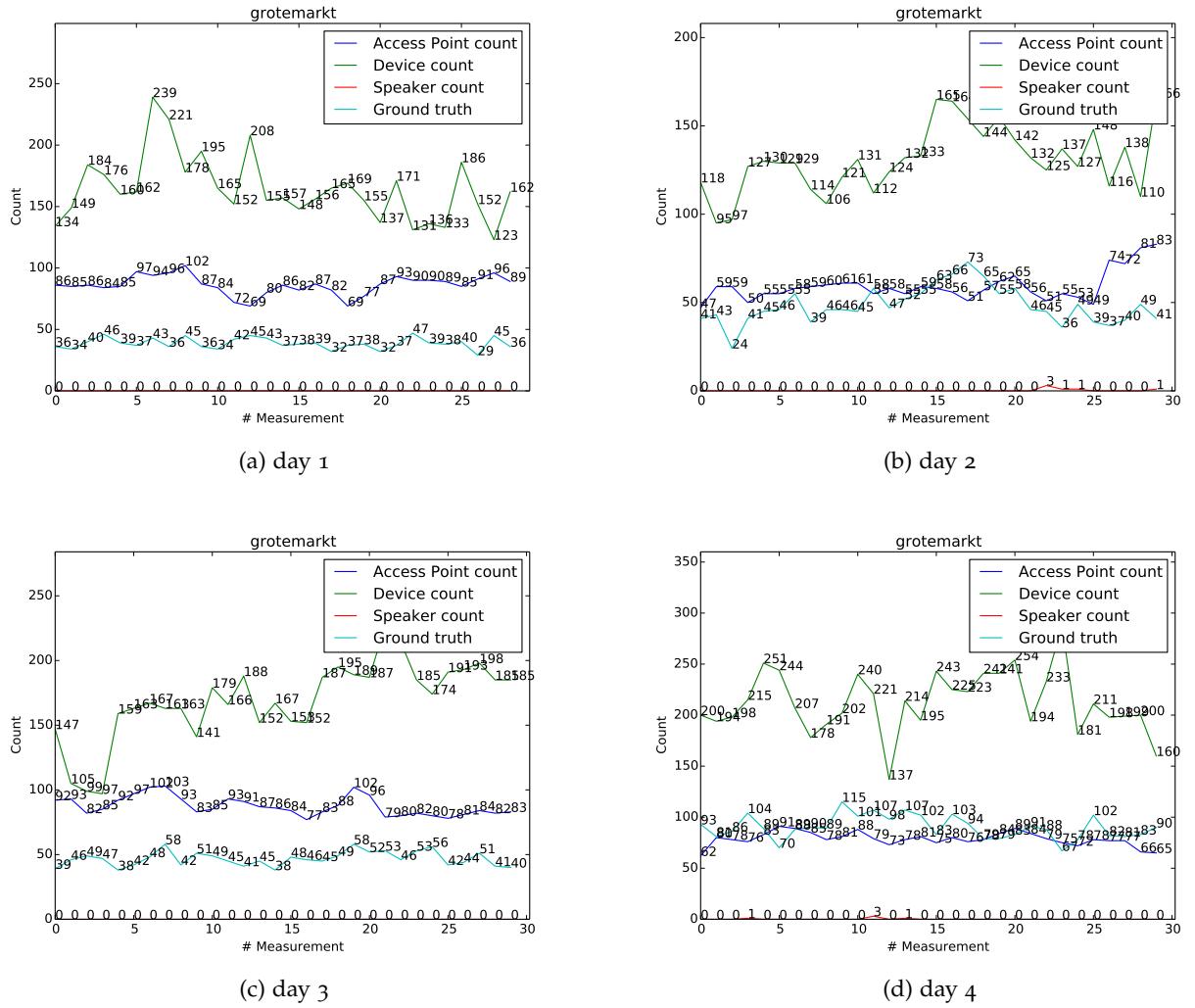


Figure 22: The line chart of sensor readings at *Grote Markt* in four days of experiment.



# B

## TIME-LAPSE IMAGES

This appendix presents an example of time-lapse images taken in each location.

### B.1 REMOTE AREA

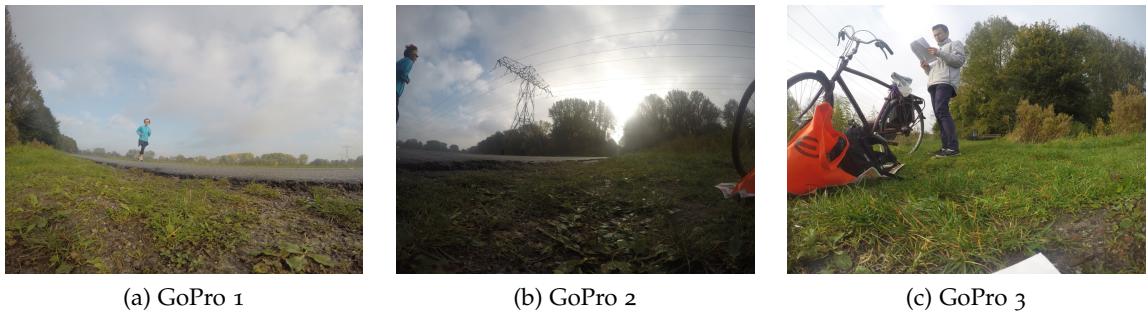


(a) GoPro 1

(b) GoPro 2

(c) GoPro 3

Figure 23: Example of time-lapse images taken at remote area in day 1.

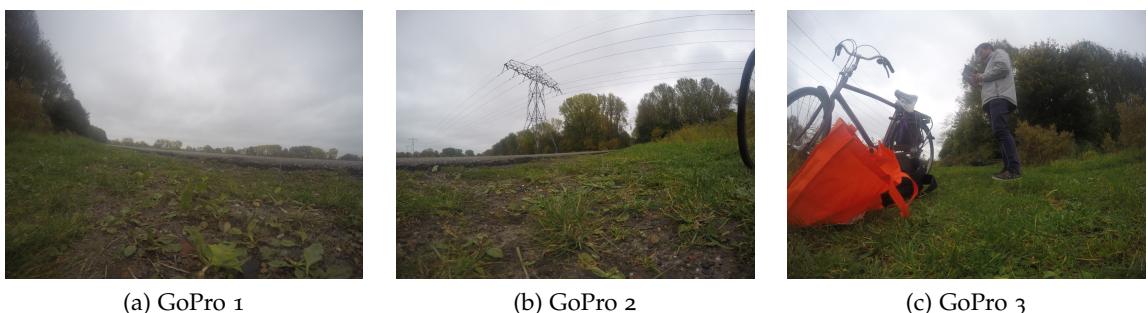


(a) GoPro 1

(b) GoPro 2

(c) GoPro 3

Figure 24: Example of time-lapse images taken at remote area in day 2.



(a) GoPro 1

(b) GoPro 2

(c) GoPro 3

Figure 25: Example of time-lapse images taken at remote area in day 3.



Figure 26: Example of time-lapse images taken at remote area in day 4.

## B.2 HOME



Figure 27: Example of time-lapse images taken at home in day 1.

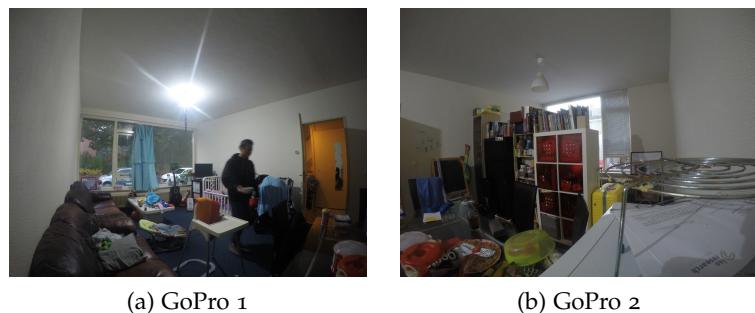


Figure 28: Example of time-lapse images taken at home in day 2.



(a) GoPro 1

(b) GoPro 2

Figure 29: Example of time-lapse images taken at home in day 3.



(a) GoPro 1

(b) GoPro 2

(c) GoPro 3

Figure 30: Example of time-lapse images taken at home in day 4.

## B.3 PADDEPOEL



(a) GoPro 1

(b) GoPro 2

(c) GoPro 3

Figure 31: Example of time-lapse images taken at *Paddepoel shopping center* in day 1.



(a) GoPro 1

(b) GoPro 2

(c) GoPro 3

Figure 32: Example of time-lapse images taken at *Paddepoel shopping center* in day 2.



(a) GoPro 1

(b) GoPro 2

(c) GoPro 3

Figure 33: Example of time-lapse images taken at *Paddepoel shopping center* in day 3.



(a) GoPro 1

(b) GoPro 2

(c) GoPro 3

Figure 34: Example of time-lapse images taken at *Paddepoel shopping center* in day 4.

## B.4 GROTE MARKT



Figure 35: Example of time-lapse images taken at *Grote markt* in day 1.



Figure 36: Example of time-lapse images taken at *Grote markt* in day 2.



Figure 37: Example of time-lapse images taken at *Grote markt* in day 3.

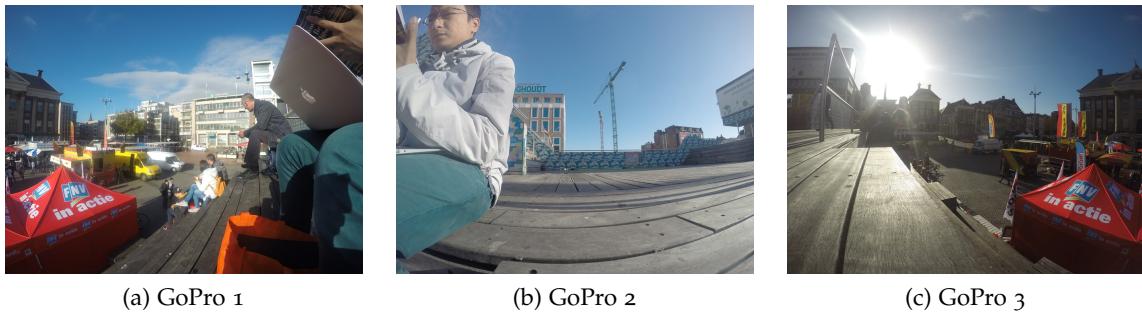


Figure 38: Example of time-lapse images taken at *Grote markt* in day 4.

# C

## AMBIENT NOISE RECORDINGS

---

### C.1 DAY 1

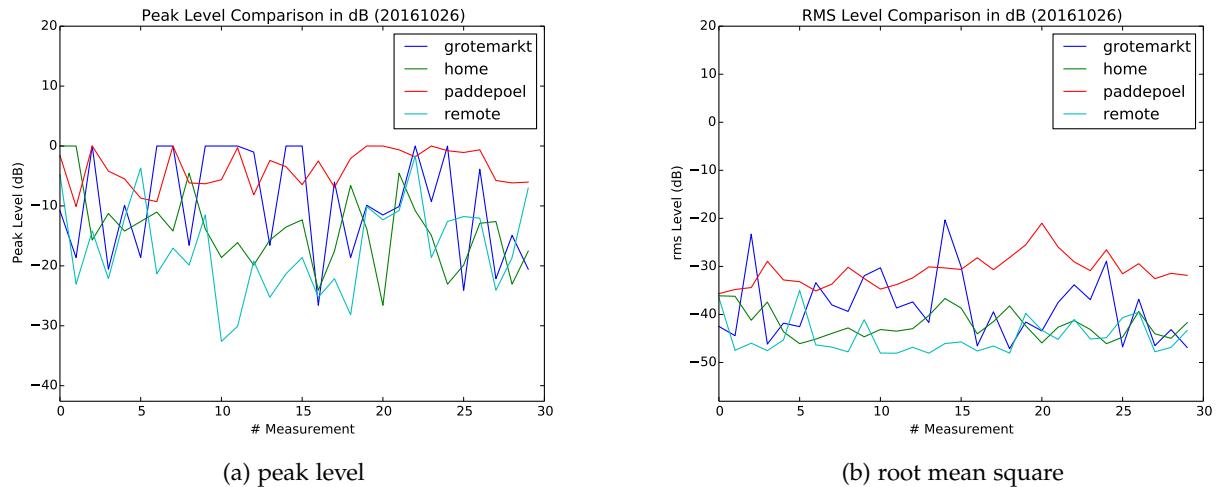


Figure 39: Line chart showing the peak level (39a) and root-mean-square (39b) in each cycle at day 1.

Table 12: The average of root-mean-square and peak level of ambient noise recording in day 1.

	RMS (dB)	PKLV (dB)
Remote area	-44.74	-17.06
Home	-42.07	-14.04
Paddepoel	-30.86	-3.74
Grote markt	-38.57	-9.67

## C.2 DAY 2

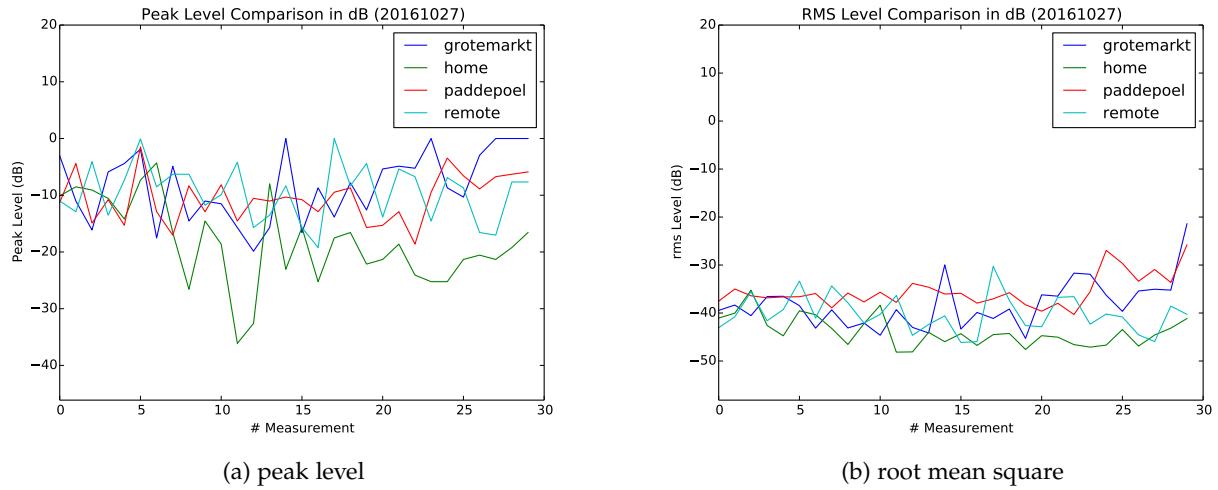


Figure 40: Line chart showing the peak level (4oa) and root-mean-square (4ob) in each cycle at day 2.

Table 13: The average of root-mean-square and peak level of ambient noise recording in day 2.

	RMS (dB)	PKLV (dB)
Remote area	-40.15	-9.53
Home	-43.89	-18.36
Paddepoel	-35.46	-10.51
Grote markt	-38.22	-8.33

## C.3 DAY 3

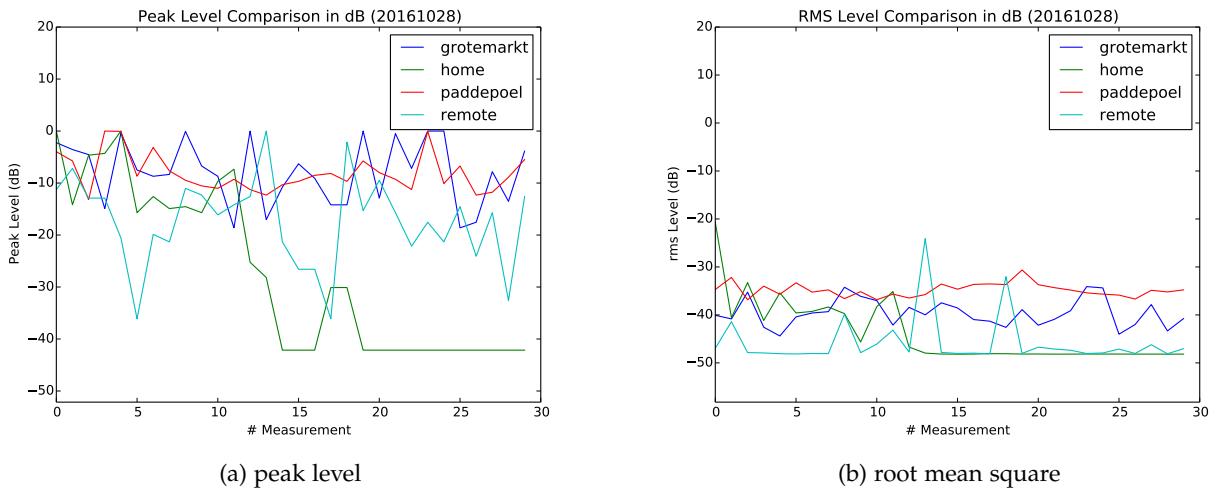


Figure 41: Line chart showing the peak level (41a) and root-mean-square (41b) in each cycle at day 3.

Table 14: The average of root-mean-square and peak level of ambient noise recording in day 3.

	RMS (dB)	PKLV (dB)
Remote area	-45.69	-17.40
Home	-43.74	-27.23
Paddepoel	-34.81	-8.08
Grote markt	-39.62	-7.92



# D

## R CODE LISTINGS

---

Table 15: Summary of R libraries in regression analysis

Regression Algorithm	R libraries
Linear regression with stepwise selection	leaps
k-Nearest Neighbors	caret
Support Vector Machine	e1071

Listing 1: Tuning linear regression with stepwise selection. Number of predictor nvmax is the tuning parameter.

```
# linear stepwise selection
library(caret)
tuning_params <- expand.grid(nvmax=seq(1,4,1))
set.seed(100)
linear.hc.stepwise <- train(hc~, data=phone_data_hc, method="
    leapSeq", trControl=fit_control, tuneGrid = tuning_params)
plot(linear.hc.stepwise)
lm(hc~, data = phone_data_hc)

tuning_params <- expand.grid(nvmax=seq(1,4,1))
set.seed(100)
linear.dc.stepwise <- train(dc~, data=phone_data_dc, method="
    leapSeq", trControl=fit_control, tuneGrid = tuning_params)
plot(linear.dc.stepwise)
lm(dc~, data = phone_data_dc)
```

Listing 2: Tuning k-Nearest Neighbors regression. Number of neighbors (k) is the tuning parameter.

```
#kNN
library(caret)
fit_control <- trainControl(method = "repeatedcv", number = 10,
    repeats = 10)
tuning_params <- expand.grid(k=seq(1,30,1))
set.seed(100)
knn.hc <- train(hc~, data=phone_data_hc, method="knn", trControl
    =fit_control, tuneGrid = tuning_params)
plot(knn.hc)

fit_control <- trainControl(method = "repeatedcv", number = 10,
    repeats = 10)
tuning_params <- expand.grid(k=seq(1,30,1))
```

```
set.seed(100)
knn.dc <- train(dc~, data=phone_data_dc, method="knn", trControl
  =fit_control, tuneGrid = tuning_params)
plotWithBars(knn.dc)
```

Listing 3: Tuning Support Vector Machine regression. Epsilon ( $\epsilon$ ) and cost are the tuning parameters.

```
# SVM for head count
library(e1071)
set.seed(100)
tuneResult <- tune(svm, hc~, data=phone_data_hc, ranges = list(
  epsilon = seq(0,1,0.1), cost = (1:10)))
plot(tuneResult)

# SVM for device count
library(e1071)
set.seed(100)
tuneResult <- tune(svm, dc~, data=phone_data_dc, ranges = list(
  epsilon = seq(0,1,0.1), cost = (1:10)))
plot(tuneResult)
```

## BIBLIOGRAPHY

---

- [1] Naeim Abedi, Ashish Bhaskar, and Edward Chung. "Tracking spatio-temporal movement of human in terms of space utilization using Media-Access-Control address data." In: *Applied Geography* 51 (2014), pp. 72–81. ISSN: 01436228. DOI: [10.1016/j.apgeog.2014.04.001](https://doi.org/10.1016/j.apgeog.2014.04.001). URL: <http://www.sciencedirect.com/science/article/pii/S0143622814000629>.
- [2] Alessio Agneessens, Igor Bisio, Fabio Lavagetto, Mario Marchese, and Andrea Sciarrone. "Speaker count application for smart-phone platforms." In: *ISWPC 2010 - IEEE 5th International Symposium on Wireless Pervasive Computing 2010*. IEEE, 2010, pp. 361–366. ISBN: 9781424468584. DOI: [10.1109/ISWPC.2010.5483789](https://doi.org/10.1109/ISWPC.2010.5483789). URL: <http://ieeexplore.ieee.org/document/5483789/>.
- [3] Nasimuddin Ahmed, Avik Ghose, Amit K. Agrawal, Chirabrata Bhaumik, Vivek Chandel, and Abhinav Kumar. "SmartEvac-Trak: A people counting and coarse-level localization solution for efficient evacuation of large buildings." In: *2015 IEEE International Conference on Pervasive Computing and Communication Workshops, PerCom Workshops 2015*. IEEE, 2015, pp. 372–377. ISBN: 9781479984251. DOI: [10.1109/PERCOMW.2015.7134066](https://doi.org/10.1109/PERCOMW.2015.7134066). URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7134066>.
- [4] Valentin Andrei, Horia Cucu, Andi Buzo, and Cornelius Burileanu. "Estimating competing speaker count for blind speech source separation." In: *2015 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*. IEEE, 2015, pp. 1–8. ISBN: 978-1-4673-7560-3. DOI: [10.1109/SPED.2015.7343081](https://doi.org/10.1109/SPED.2015.7343081). URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7343081>.
- [5] VeldMark B.V. *Winkelcentrum Paddepoel*. 2016. URL: <http://www.inwinkelcentra.nl/winkelcentrum-paddepoel/> (visited on 12/08/2016).
- [6] Anja Bachmann. "Towards smartphone-based sensing of social interaction for ambulatory assessment." In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers - UbiComp '15* (2015), pp. 423–428. DOI: [10.1145/2800835.2801642](https://doi.org/10.1145/2800835.2801642). URL: <http://dl.acm.org/citation.cfm?doid=2800835.2801642>.

- [7] Anja Bachmann, Robert Zetsche, Till Riedel, Michael Beigl, Markus Reichert, Philip Santangelo, and Ulrich Ebner-Priemer. "Identification of relevant sensor sources for context-aware ESM apps in ambulatory assessment." In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers - UbiComp '15*. New York, New York, USA: ACM Press, 2015, pp. 265–268. ISBN: 9781450335751. DOI: [10.1145/2800835.2800944](https://doi.org/10.1145/2800835.2800944). URL: <http://dl.acm.org/citation.cfm?doid=2800835.2800944>.
- [8] Marco V. Barbera, Alessandro Epasto, Alessandro Mei, Vasile C. Perta, and Julinda Stefa. "Signals from the crowd: Uncovering social relationships through smartphone probes." In: *Proceedings of the 2013 Conference on Internet Measurement Conference* (2013), pp. 265–276. DOI: [10.1145/2504730.2504742](https://doi.org/10.1145/2504730.2504742). URL: <http://dl.acm.org/citation.cfm?doid=2504730.2504742>.
- [9] Hilgo Bruining, Leo de Sonneville, Hanna Swaab, Maretha de Jonge, Martien Kas, Herman van Engeland, and Jacob Vorstman. "Dissecting the Clinical Heterogeneity of Autism Spectrum Disorders through Defined Genotypes." In: *PLOS ONE* 5.5 (2010), pp. 1–7. DOI: [10.1371/journal.pone.0010887](https://doi.org/10.1371/journal.pone.0010887). URL: <http://dx.doi.org/10.1371/\%2Fjournal.pone.0010887>.
- [10] Z Chen, Yiqiang Chen, Shuangquan Wang, and Junfa Liu. "Inferring social contextual behavior from bluetooth traces." In: *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. New York, New York, USA: ACM Press, 2013, pp. 267–270. ISBN: 9781450322157. DOI: [10.1145/2494091.2494176](https://doi.org/10.1145/2494091.2494176). URL: <http://dl.acm.org/citation.cfm?doid=2494091.2494176>.
- [11] Pedro Cunha, Daniel C. Moura, and Universidade Porto. "A scalable and privacy preserving approach for counting pedestrians in urban environment." In: *2015 12th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (2015), pp. 1–6. DOI: [10.1109/AVSS.2015.7301806](https://doi.org/10.1109/AVSS.2015.7301806). URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7301806>.
- [12] Saandip Depatla, Arjun Muralidharan, and Yasamin Mostofi. "Occupancy Estimation Using Only WiFi Power Measurements." In: *IEEE Journal on Selected Areas in Communications* 33.7 (2015), pp. 1381–1393. ISSN: 0733-8716. DOI: [10.1109/JSAC.2015.2430272](https://doi.org/10.1109/JSAC.2015.2430272). URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7102673>.

- [13] Simone Di Domenico, Giuseppe Bianchi, and Roma Tor. "A Trained-once Crowd Counting Method Using Differential WiFi Channel State Information." In: *Proceedings of the 3rd International on Workshop on Physical Analytics - WPA '16* (2016), pp. 37–42. DOI: [10.1145/2935651.2935657](https://doi.org/10.1145/2935651.2935657). URL: <http://dl.acm.org/citation.cfm?doid=2935651.2935657>.
- [14] Paul Eskes, Marco Spruit, Sjaak Brinkkemper, Jacob Vorstman, and Martien J. Kas. "The sociability score: App-based social profiling from a healthcare perspective." In: *Computers in Human Behavior* 59 (2016), pp. 39–48. ISSN: 07475632. DOI: [10.1016/j.chb.2016.01.024](https://doi.org/10.1016/j.chb.2016.01.024).
- [15] Julien Freudiger. "Short: How Talkative is your Mobile Device? An Experimental Study of Wi-Fi Probe Requests." In: *WiSec '15 Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. New York, New York, USA: ACM Press, 2015, pp. 1–6. ISBN: 9781450336239. DOI: [10.1145/2766498.2766517](https://doi.org/10.1145/2766498.2766517). URL: <http://dl.acm.org/citation.cfm?doid=2766498.2766517>.
- [16] Gemeente Groningen. *Evenementenlocatie Grote Markt*. Groningen: Gemeente Groningen, 2016, p. 15. URL: <https://gemeente.groningen.nl/sites/default/files/16-evenementenprofiel-grote-markt-br-a4-cw-maart.pdf>.
- [17] Chris Hoffman. *Is It Illegal To Root Your Android or Jailbreak Your iPhone?* 2014. URL: <https://gopro.com/support/articles/hero4-camera-battery-life> (visited on 12/08/2016).
- [18] J. D. Hunter. "Matplotlib: A 2D graphics environment." In: *Computing In Science & Engineering* 9.3 (2007), pp. 90–95. DOI: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55).
- [19] IEEE Computer Society. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. Vol. 11. March. IEEE, 2012. ISBN: 9780738172118. URL: <http://standards.ieee.org/about/get/802/802.11.html>.
- [20] Max Kuhn. Contributions from Jed Wing et al. *caret: Classification and Regression Training*. R package version 6.0-73. 2016. URL: <https://CRAN.R-project.org/package=caret>.
- [21] Pravein Govindan Kannan, Seshadri Padmanabha Venkatagiri, Mun Choon Chan, Akhihebbal L. Ananda, and Li-Shiuan Peh. "Low cost crowd counting using audio tones." In: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems - SenSys '12* (2012), p. 155. DOI: [10.1145/2426656.2426673](https://doi.org/10.1145/2426656.2426673). URL: <http://dl.acm.org/citation.cfm?doid=2426656.2426673>.

- [22] Martien J Kas, René S Kahn, David A Collier, John L Waddington, Jesper Ekelund, David J Porteous, Klaus Schughart, and Iiris Hovatta. "Translational Neuroscience of Schizophrenia: Seeking a Meeting of Minds Between Mouse and Man." In: *Science Translational Medicine* 3.102 (2011), 102mr3–102mr3. ISSN: 1946-6234. DOI: [10.1126/scitranslmed.3002917](https://doi.org/10.1126/scitranslmed.3002917). URL: <http://stm.sciencemag.org/content/3/102/102mr3>.
- [23] M B Kjærgaard, M Wirz, D Roggen, and G Tröster. *Mobile sensing of pedestrian flocks in indoor environments using WiFi signals*. 2012. DOI: [10.1109/PerCom.2012.6199854](https://doi.org/10.1109/PerCom.2012.6199854).
- [24] Lu Luo, Jun Yang, Xuan Bao, Zhixian Yan, and Yifei Jiang. "SWAN: A Novel Mobile System to Track and Analyze Social Well-being." In: *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. New York, New York, USA: ACM Press, 2015, pp. 703–712. ISBN: 978-1-4503-3575-1. DOI: [10.1145/2800835.2804395](https://doi.org/10.1145/2800835.2804395). URL: <http://dl.acm.org/citation.cfm?doid=2800835.2804395> <https://doi.acm.org/10.1145/2800835.2804395>.
- [25] A Di Luzio, Alessandro Mei, and Julinda Stefa. "Mind your probes: De-anonymization of large crowds through smartphone WiFi probe requests." In: *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9. ISBN: VO -. DOI: [10.1109/INFOCOM.2016.7524459](https://doi.org/10.1109/INFOCOM.2016.7524459). URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7524459>.
- [26] David Meyer, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071)*, TU Wien. R package version 1.6-7. 2015. URL: <https://CRAN.R-project.org/package=e1071>.
- [27] Ricardo O. Mitchell, Hammad Rashid, Fakir Dawood, and Ali Alkhaldi. "Hajj crowd management and navigation system: People tracking and location based services via integrated mobile and RFID systems." In: *International Conference on Computer Applications Technology, ICCAT 2013*. 2013. ISBN: 9781467352857. DOI: [10.1109/ICCAT.2013.6522008](https://doi.org/10.1109/ICCAT.2013.6522008).
- [28] Masayuki Nakatsuka. "A Study on Passive Crowd Density Estimation using Wireless Sensors." In: *The Fourth International Conference on Mobile Computing and Ubiquitous Networking*. 2008. URL: <http://citeserx.ist.psu.edu/viewdoc/summary?doi=10.1.1.160.9244>.
- [29] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2016. URL: <https://www.R-project.org/>.

- [30] Valentin Radu, Lito Kriara, and Mahesh K. Marina. "Pazl: A mobile crowdsensing based indoor WiFi monitoring system." In: *2013 9th International Conference on Network and Service Management, CNSM 2013 and its three collocated Workshops - ICQT 2013, SVM 2013 and SETM 2013*. IEEE, 2013, pp. 75–83. ISBN: 9783901882531. DOI: [10.1109/CNSM.2013.6727812](https://doi.org/10.1109/CNSM.2013.6727812). URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6727812>.
- [31] David Ryan, Simon Denman, Clinton Fookes, and Sridha Sridharan. "Crowd Counting Using Group Tracking and Local Features." In: *Proceedings of the 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance* (2010), pp. 218–224. DOI: [10.1109/AVSS.2010.30](https://doi.org/10.1109/AVSS.2010.30).
- [32] Lorenz Schauer, Martin Werner, and Philipp Marcus. "Estimating Crowd Densities and Pedestrian Flows Using Wi-Fi and Bluetooth." In: *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services* (2014), pp. 171–177. DOI: [10.4108/icst.mobiquitous.2014.257870](https://doi.org/10.4108/icst.mobiquitous.2014.257870). URL: <http://eudl.eu/doi/10.4108/icst.mobiquitous.2014.257870>.
- [33] André Schmidt. "Low-cost Crowd Counting in Public Spaces." In: (2014). URL: [http://andsch.com/sites/default/files/crowdcounting{\\_\}andre{\\_\}schmidt.pdf](http://andsch.com/sites/default/files/crowdcounting{_\}andre{_\}schmidt.pdf).
- [34] Maarten Van Segbroeck, Andreas Tsirtas, and Shrikanth S. Narayanan. "A robust frontend for VAD: Exploiting contextual, discriminative and spectral cues of human voice." In: *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*. August. 2013, pp. 704–708.
- [35] GoPro Support. *HERO4 Camera Battery-life*. 2016. URL: <https://gopro.com/support/articles/hero4-camera-battery-life> (visited on 12/08/2016).
- [36] GoPro Support. *HERO4 Field of View (FOV) Information*. 2016. URL: [https://gopro.com/help/articles/Question\\_Answer/HERO4-Field-of-View-FOV-Information](https://gopro.com/help/articles/Question_Answer/HERO4-Field-of-View-FOV-Information) (visited on 12/08/2016).
- [37] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S Cardoso, and Piessens Frank. "Why MAC Address Randomization is not Enough: An Analysis of WiFi Network Discovery Mechanisms." In: *ACM on Asia Conference on Computer and Communications Security*. 2016. ISBN: 9781450342339. DOI: [10.1145/2897845.2897883](https://doi.org/10.1145/2897845.2897883).
- [38] Edwin Vattapparamban. "People Counting and occupancy Monitoring using WiFi Probe Requests and Unmanned Aerial Vehicles." In: *FIU Electronic Theses and Dissertations* (2016). URL: <http://digitalcommons.fiu.edu/etd/2479>.

- [39] Mathias Versichele, Tijs Neutens, Matthias Delafontaine, and Nico Van de Weghe. "The use of Bluetooth for analysing spatiotemporal dynamics of human movement at mass events: A case study of the Ghent Festivities." In: *Applied Geography* 32.2 (2012), pp. 208–220. ISSN: 01436228. DOI: [10.1016/j.apgeog.2011.05.011](https://doi.org/10.1016/j.apgeog.2011.05.011).
- [40] Yan Wang, Jie Yang, Hongbo Liu, and Yingying Chen. "Measuring human queues using WiFi signals." In: *Proceedings of the 19th annual international conference on Mobile computing & networking*. New York, New York, USA: ACM Press, 2013, pp. 235–237. ISBN: 9781450319997. DOI: [10.1145/2500423.2504584](https://doi.org/10.1145/2500423.2504584). URL: <http://dl.acm.org/citation.cfm?doid=2500423.2504584http://dl.acm.org/citation.cfm?id=2504584>.
- [41] Jens Weppner and Paul Lukowicz. "Collaborative Crowd Density Estimation with Mobile Phones." In: *Proc. of ACM PhoneSense* (2011). URL: <http://131.107.65.14/en-us/um/redmond/events/phonesense2011/papers/CollaborativeCrowd.pdf>.
- [42] Jens Weppner and Paul Lukowicz. "Bluetooth based collaborative crowd density estimation with mobile phones." In: *2013 IEEE International Conference on Pervasive Computing and Communications (PerCom)* (2013), pp. 193–200. DOI: [10.1109/PerCom.2013.6526732](https://doi.org/10.1109/PerCom.2013.6526732). URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6526732http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6526732>.
- [43] W Xi, J Zhao, X Y Li, K Zhao, S Tang, X Liu, and Z Jiang. "Electronic frog eye: Counting crowd using WiFi." In: *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. 2014, pp. 361–369. ISBN: 0743-166X VO -. DOI: [10.1109/INFOCOM.2014.6847958](https://doi.org/10.1109/INFOCOM.2014.6847958).
- [44] Chenren Xu, Sugang Li, Gang Liu, and Yanyong Zhang. "Crowd ++: Unsupervised Speaker Count with Smartphones." In: *Ubicomp*. New York, New York, USA: ACM Press, 2013, pp. 43–52. ISBN: 9781450317702. DOI: [10.1145/2493432.2493435](https://doi.org/10.1145/2493432.2493435). URL: <http://dl.acm.org/citation.cfm?doid=2493432.2493435>.
- [45] Ooi Boon Yaik, Kong Zan Wai, Ian K.T.Tan, and Ooi Boon Sheng. "Measuring the Accuracy of Crowd Counting using Wi-Fi Probe-Request-Frame Counting Technique." In: *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)* 8.2 (2016), pp. 79–81. ISSN: 2289-8131.
- [46] Takuya Yoshida. "Estimating the number of people using existing WiFi access point in indoor environment." In: *6th European Conference of Computer Science (ECCS '15)*. 2015, pp. 46–53. ISBN: 9781618043443.

- [47] Moustafa Youssef and Ashok Agrawala. "The Horus location determination system." In: *Wireless Networks* 14.3 (2008), pp. 357–374. ISSN: 1572-8196. DOI: [10.1007/s11276-006-0725-7](https://doi.org/10.1007/s11276-006-0725-7). URL: <http://dx.doi.org/10.1007/s11276-006-0725-7>.
- [48] Yaoxuan Yuan, Chen Qiu, Wei Xi, and Jizhong Zhao. "Crowd Density Estimation Using Wireless Sensor Networks." In: *2011 Seventh International Conference on Mobile Ad-hoc and Sensor Networks* (2011), pp. 138–145. DOI: [10.1109/MSN.2011.31](https://doi.org/10.1109/MSN.2011.31). URL: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6117405>.