On the feasibility and privacy benefits of on-device data mining for opportunistic crowd-sensing and service self-provisioning

Fanourakis, Marios Aristogenis

On the Feasibility and Privacy Benefits of On-Device Data Mining for Opportunistic Crowd-Sensing and Service Self-Provisioning

THÈSE

présentée à la Faculté d'Economie et de Management de l'Université de Genéve

par

Marios Fanourakis

sous la direction de

prof. Dimitri Konstantas
prof. Katarzyna Wac

pour l'obtention du grade de

Docteur ès Économie et Management mention Systèmes d'Information

Membres du Jury de thèse:

Prof. Marcel PAULSSEN, président du jury, GSEM - IOM
Prof. Katarzyna WAC, co-directuer de thèse, GSEM - CUI
Prof. Dimitri KONSTANTAS, co-directuer de thèse, GSEM - CUI
Dr. Niels NIJDAM, GSEM - CUI
Prof. Ciaran BRYCE, HEG, Suisse
Dr. Sebastien ZIEGLER, Président IoT Forum, Suisse

Thèse no 63
Genève, 1 Novembre 2018

La Faculté d'Economie et de Management, sur préavis du jury, a autorisé l'impression de la présente thèse, sans entendre, par-là, émettre aucune opinion sur les propositions qui s'y trouvent énoncées et qui n'engagent que la responsabilité de leur auteur.

Genève, le 1 Novembre 2018

Le Doyen
Marcelo Olarreaga

Impression d'après le manuscrit de l'auteur

Résumé

Les services d'appareils mobiles deviennent de plus en plus utiles aux utilisateurs en automatisant bon nombre des tâches qu'ils auraient normalement à effectuer manuellement. Cela est facilité par l'amélioration continue de la sophistication des services et des capacités des appareils. De l'évolution du simple rappel à une multitude d'actions automatisées qui sont soit définies par l'utilisateur, soit, dans un avenir proche, apprises par les dispositifs/services eux-mêmes.

L'appareil mobile moyen comprend plusieurs capteurs standard : un accéléromètre (pour savoir quand l'écran tourne), une boussole (pour le positionnement), un GPS (pour le positionnement), un capteur de lumière (pour régler la luminosité de l'écran), un microphone et une caméra. D'autres dispositifs peuvent inclure différents types de capteurs comme la température, la qualité de l'air, la fréquence cardiaque, etc. De plus, un téléphone mobile équipé de tels capteurs se déplace avec son propriétaire et peut être utilisé pour collecter des informations contextuelles en leur nom. Cela peut provenir d'une demande explicite, en supposant une participation et une coopération humaines actives dans la détection du contexte - ce que l'on appelle la détection participative.

Il est souvent essentiel de recueillir des données (température, qualité de l'air, utilisation du téléphone, etc.) afin de créer des modèles réalistes qui pourraient nous aider à comprendre et à prédire le monde ou à vérifier les théories et les modèles développés en laboratoire. En supposant que les modèles et les algorithmes prédictifs sont en place, le niveau d'automatisation n'est limité que par sa capacité d'accéder aux flux de données et d'information ; cependant, le partage de plus en plus de données personnelles augmente le risque que la vie privée d'un utilisateur soit compromise en révélant son identité. Bien que des lignes directrices et des comités d'éthique aient été mis en place pour protéger les utilisateurs quant à la façon dont ces données sont recueillies et utilisées, il y a encore des préoccupations en matière de protection de la vie privée qui doivent être abordées dans les mécanismes fondamentaux par lesquels ces données sont recueillies et diffusées. Dans cette thèse, nous montrons que la façon la plus sûre de protéger la vie privée dans la détection opportuniste est de ne pas partager les données qui constituent une menace pour la vie privée provenant du dispositif lorsque le service ou la tâche peut très bien être exécuté sur le dispositif lui-même. Nous montrons que la plupart des données des capteurs d'un appareil doivent être manipulées avec prudence en raison de leur potentiel de menace pour la vie privée et proposons des solutions d'autoapprovisionnement des services pour mesurer l'information de localisation par triangulation sans aide et le contexte de localisation en utilisant des traces d'identification cellulaire. Lorsque les données doivent absolument parvenir à une tierce partie, nous montrons que des stratégies de mélange opportunistes peuvent en effet être efficaces, mais pas nécessairement efficaces dans le temps, pour anonymiser la source des données, mais que les données elles-mêmes doivent être protégées des attaques d'inférence en utilisant des méthodes supplémentaires de brouillage.

Abstract

Mobile device services are increasingly becoming more and more useful to users by automating many of the tasks that users would normally have to perform manually. This is facilitated by the on-going improvements of the sophistication of services and capabilities of devices. From the evolution of the simple reminder to a multitude of automated actions that are either defined by the user or, in the not-so-far future, learned by the devices/services themselves.

The average mobile device includes several sensors as a standard feature: accelerometer (to know when the screen rotates), compass (for positioning), GPS (for positioning), light (to adjust the display brightness), audio (microphone), and image (camera). Other devices may include different types of sensors like temperature, air quality, heart rate, etc. Moreover, a mobile phone with such sensors roams with its owner, and can be used to collect context information on their behalf. This can originate in an explicit request, assuming an active human participation and cooperation in context sensing – denoted as participatory sensing. It is often vital to collect data (temperature, air quality, phone usage, etc.) in order to create realistic models that might help us understand and predict the world or verify theories and models developed in lab environments. Assuming that the predictive models and algorithms are in place, the level of automation is only limited by the ability to access data streams and information; however, sharing more and more personal data increases the chance of a user's privacy being compromised by revealing their identity. Although ethical guidelines and committees have been put in place to protect people in terms of how this data is collected and how it is used, there are still privacy concerns that need to be addressed in the fundamental mechanisms by which this data is collected and disseminated.

In this thesis we show that the most secure way to proceed with privacy in opportunistic sensing is to not share data that is a privacy threat from the device when the service or task can very well be performed on the device itself. We show that most of the sensor data on a device should be handled with caution due to their potential to be a privacy threat and propose solutions for service self-provisioning for measuring location tracking information through unaided triangulation, and location context by using cell ID traces. When data absolutely needs to reach a third party, we show that opportunistic mixing strategies can indeed be effective, but not necessarily time-efficient, in anonymizing the source of the data, however the data itself needs to be shielded from inference attacks by using additional obfuscation methods.

Acknowledgements

First and foremost, I would like to thank both of my supervisors who guided me throughout this work. Thank you to Katarzyna Wac for her immense patience, priceless feedback, and invaluable support throughout my time as a doctoral student. She gave me the tools and motivation that allowed me to fulfill this task. Thank you to Dimitri Konstantas who helped me shape the context of this thesis and made sure I was on track.

A big thank you to my colleagues, fellow members of the Quality of Life research group, who helped me collect data, implement solutions, and inspire me with knowledge and ideas.

Thank you to each of the jury members who gave me valuable feedback on my thesis to make the arguments, evidence, and solutions stronger and clearer.

I am grateful to all the new friends I made during this journey who made life in Geneva a truly wonderful experience.

This thesis was supported by the European projects CaMeLi (AAL), Miraculous Life (FP7), and GrowMeUp (H2020).

Table of Contents

Chapter 1

Introduction

Mobile device services are increasingly becoming more and more useful to users by automating many of the tasks that users would normally have to perform manually. There has been an exponential increase in computation, communication, and storage capacities available on personalized mobile and miniaturized devices such as smartphones [1–3]. A simple reminder service on a calendar or agenda has evolved to not only take temporal context such as the time to be triggered but other contextual cues have been integrated such as location. This enables a use case where someone can be reminded to buy a spare light bulb when they enter the store to buy something unrelated (a Google reminder service already available to consumers). Smart home ecosystems such as Google Home or Amazon Echo are already available to consumers to control *connected* things in their home using their devices. All these connected things, which fall into the Internet of Things (IoT) nomenclature, collect and consume data in order to provide some convenience to the user. The smart features of the modern reminder and a multitude of other automated actions has been supported mostly by manual definitions that the user enters (for example, adding a location context manually for a reminder) but more recently, the capability of software to suggest actions which are automatically learned by the devices/services themselves is emerging.

What does it take to achieve such automation? Data and data mining. Data mining encompasses any method or algorithm in which knowledge is produced from the data. This includes but not limited to pattern recognition and machine learning. Actionable knowledge can then be used to implement services that provide task automation. It is worth noting that in research, it is often vital to collect data (temperature, air quality, phone usage, etc.) in order to create realistic models that might help us predict and understand the world or verify theories and models developed in lab environments. Assuming that the predictive models and algorithms are in place, the level of automation is only limited by the ability to access data streams and information; however, sharing more and more personal data increases the chance of a user's privacy being compromised by revealing their identity.

Data can be obtained through many types of sensors or surveys. Already, the average mobile device includes several sensors as a standard feature: ac-

celerometer (to know when the screen rotates), compass (for positioning), GPS (for positioning), light (to adjust the display brightness), audio (microphone), image (camera), and others. Other devices may include different types of sensors like temperature, air quality, heart rate, etc. Moreover, a mobile phone with such sensors roams with its owner, and can be used to collect context information on their behalf. This can originate in an explicit request, assuming an active human participation and cooperation in context sensing – denoted as participatory sensing [4–6]. This participatory approach usually relates to urban sensing campaigns, where mobile users manually tag specific locations with campaign-specific data, for example, uncollected garbage. In contrast, the opportunistic sensing concept builds upon sensing in an autonomic, continuous but unobtrusive way for the individual, for example, to generate air pollution maps throughout a city or provide other essential services to the user [7].

In an opportunistic setting, where the data collection is managed autonomously, user privacy must be taken into consideration. Opportunistic networks are a form of mobile ad hoc networks that exploit the human social characteristics, such as similarities, daily routines, mobility patterns, and interests to perform message routing and data sharing. In such networks, the users with mobile devices are able to form on-the-fly social networks to communicate with each other and share data objects. Although the sensing phase can be opportunistic, there are strategies for the data reporting phase which may not be opportunistic in nature. For example, there may be a centralized entity which is used to anonymize the data before the data collector receives it. As such, it is important to distinguish this kind of semi-opportunistic sensing from a *fully opportunistic* one where all parts of the sensing approach (mainly sensing and reporting) are achieved opportunistically. Since the users do not have complete control over what/when data is being collected and used, it can become a very intrusive strategy and as a result, users will be reluctant to participate in such a setting in order to keep control of their data and privacy. Thus, there is a need for a methodology of data collection that can also provide privacy guarantees to the users. Although ethical guidelines and committees have been put in place to protect people in terms of how this data is collected and how it is used, there are still privacy concerns that need to be addressed in the fundamental mechanisms by which this data is collected and disseminated. Sensitive data such as location must be abstracted or obfuscated in such a way that it cannot be linked back to a specific user while still retaining its utility.

What is Privacy? In this thesis we will adopt the definition of privacy, in the context of information technology, described by the *Common Criteria Recognition Arrangement* [8] (CC), a multinational arrangement. They describe privacy as user protection against discovery and misuse of identity by other users. We also adopt definitions described in Pfitzmann et al. [9] which closely resemble the ones of CC. According to CC, guaranteeing privacy requires the following:

Anonymity A user may use a resource or service without disclosing a user's identity. Other users or subjects are unable to determine the identity of a user bound to a subject or operation.

Pseudonymity A pseudonym is an identifier of a subject other than one of the subject's real names. It requires that a set of users and/or subjects are unable to determine the identity of a user bound to a subject or operation, but that this user is still accountable for its actions.

Unlinkability It ensures that a user may make multiple uses of resources or services without others being able to link these uses together. In general, Unlinkability of two or more items of interest (IOIs, for example, subjects, messages, actions, ...) from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not.

Unobservability It ensures that a user may use a resource or service without others, especially third parties, being able to observe that the resource or service is being used. In general, unobservability of an item of interest (IOI) means undetectability of the IOI against all subjects uninvolved in it and anonymity of the subject(s) involved in the IOI even against the other subject(s) involved in that IOI. Where undetectability of an item of interest (IOI) from an attacker's perspective means that the attacker cannot sufficiently distinguish whether it exists or not.

1.1 Motivation

Privacy has been a hot topic in recent years. The availability of automatic data collection methods and tools, propelled by continuous and rapid improvement of machine learning techniques poses a threat to the privacy of everyday users. More and more meaningful personal information can be inferred or derived from seemingly unimportant data. A malicious entity can use the data to target users in both the physical and digital world. They can sell the data to unethical organizations that use it to exploit users or criminals can gain access to the data in order to understand a user's daily routine and to plan a burglary or other criminal acts. Service providers that collect data may not only provide the intended services but some may also sell it to advertising entities and as a consequence potentially increase the vulnerability to malicious attacks that can compromise the user's security and even their well-being. This deters privacy-conscious users from using certain services that they deem to be too intrusive; it becomes a balancing act between the data they are willing to share and the services they want. It is more apparent in the phone usage case, but even the ambient light sensor can potentially be used to reveal features of a person's daily routines. At the same time, it is increasingly useful and important for societies and policy-makers to perform crowdsourced studies to understand population trends (for example, epidemiology) as well as to provide improved automated services to users. For example, it might be useful to understand if the introduction of safer cycling paths had an effect on the frequency of physical exercise in the population. It is also useful to understand when to show certain notifications to the user during their daily routines (for example, traffic information for the work commute, weather information for a trip, etc.).

1.1.1 The Mishandling or Malicious Use of Personal Data

Why care about privacy? A breach of privacy and anonymity in the data of individuals or populations can have social, political, and economical implications. Several authoritarian regimes implement diverse mechanisms for the dismissal of both privacy and anonymity to monitor their citizens ideologies and keep the status quo [10]. There have been revelations that show some of these mechanisms implemented even in regions such as the USA and Europe. The most infamous case of such mechanisms were revealed by the Snowden leaks and brought to light that the National Security Agency (NSA) in the USA was essentially spying on their citizens [11]. Besides governments, the power of large corporations cannot be dismissed. We see this with the recent Cambridge Analytica acquisition of millions of user's Facebook data was allegedly used to influence the USA presidential elections in 2016 [12]. Overall, privacy and anonymity is important not only for sustaining moral values such as freedom of speech but also individual safety. Corporations have an extraordinary amount of data on their users and are often targeted by malicious persons or entities seeking to gain access to such information by exploiting vulnerabilities in the security infrastructure that is put in place to protect this data. Other times corporations may provide a social feature that inadvertently puts their users at risk. This was the case with Strava, a fitness application, which created a public heatmap of running paths to aid users in finding a good and tested path to run on. The problem became apparent when running paths and patrol routes on military bases around the world, where soldiers also used fitness trackers, were shown on the public heatmap. This caused major security concerns but the issue was mitigated by adjusting personnel policies and the default settings of the application [13].

The ubiquitous nature of location-aware devices that we carry makes it possible for location-based services to function and collect location data from us. Even if this data is anonymized, it is relatively simple to find out who it belongs to and reveal user behaviour, preferences, and beliefs. The subsequent danger to user safety and autonomy is substantial [14]. Location is used for a variety of tasks ranging from high utility such as navigation or fitness tracking, which require a relatively high degree of accuracy, to lower utility such as social media (sharing current location) or location-based notifications, which most often do not require an accuracy better than 50 meters. In every smartphone the location is determined using a location service such as *Google Location* which will provide the geographic location in the form of GPS coordinates to any application that has the appropriate permissions. This is necessary when we consider navigation applications but it is excessive for applications that do not need exact coordinates but instead need contextual location information, like an abstract place such as home or work or tracking information like the total distance traveled. Giving such sensitive information to any application regardless of what the application actually requires to function poses an immense risk to user privacy. Google could easily expand its location service *API* to provide geographic location, contextual location, or tracking information separate from each other but it is not the case for the moment. Many solutions to provide such services exist in the literature but they are mainly used as research tools rather than a commercial solution and are often

for very specific situations. A generic and well rounded location service with the aforementioned features is yet to be made available to consumers.

1.1.2 The Value and Exploitation of Personal Data

The explosion of data in the age of information has inevitably led to its monetization and exploitation which has been called the ”new oil” [15]. In an article on Financial Times by Emily Steel et al. [16] they analyzed the data broker industry and created a tool to estimate the value of an individual's data based on certain life milestones or general facts about them and on what data points are available. General information about a person like age, gender, and location (postal code) cost a mere $0.0015 while getting more and more specific about marital status, ethnicity, consumer habits, and especially health can increase the cost significantly (for example, a data point that indicates the individual has asthma increases the cost to $0.2615). Although these are the costs associated with buying and selling personal data, they do not reflect the increased revenue that results from properly utilizing this data for advertising and market research. For example, advertisers can pay Facebook to show a certain ad to a specific demographic or people meeting some specific criteria. This strategy of targeted advertising is what creates a big portion of the personal data revenue. Another way that companies can capitalize on personal data is through market research. This strategy allows a company to gage the acceptance and usability of ideas or products on the consumers before it is made or during its initial release in order to make future improvements.

The data monetizing strategies that we mentioned allow companies such as Facebook, and Google to provide services to consumers without a subscription or other forms of monetary payment. The business model relies on consumer data to make money through various means such as advertising and, as a result, the phrase ”if you're not paying for a product, then you **are** the product” is often cited. Companies with as much influence as Facebook and Google have the upper hand but as consumers start to question their data practices, these companies may finally be impacted enough to act. Indeed, consumers are becoming more aware of this fact after recent news of data breaches or malicious use of personal data and more importantly they are becoming aware of the value of their own data and privacy. Can companies and consumers find a compromise? A couple of solutions might help to find a middle ground. The most straightforward is to directly pay users for the use of their data. A portion of the advertising profits could be passed along to consumers. Other pricing strategies are presented by Li et al. [17] and Gkatzelis et al. [18]. Another way to find a compromise is to allow users to pay a fee for privacy. Although this might seem convoluted in the sense that privacy is seen as a fundamental right and no one should have to pay for it, a consumer often-times agrees to relinquish it in exchange for signing up for a ”free” service. Companies could safeguard consumer privacy by requiring a fee for the service they provide. The price a consumer is willing to pay may vary significantly depending on the utility of the service and how companies frame the choices available to them [19–21]. This latter point must be addressed because it can manipulate consumer behaviour towards privacy in negative ways. A similar strategy is already

applied to some smartphone applications, websites, and software which gives the users a choice between receiving advertisements or paying a fee. Such a strategy can easily extend to give users a choice between the service or website collecting user data for targeted advertising or paying a fee to support the creators and maintainers of the service or website.

Although users are aware of privacy issues with some data like location [22] they may still choose to share it. This disparity between user privacy attitudes and actual behaviour is known as the privacy paradox. Many studies have been performed to understand the dynamics of this phenomenon and the general consensus is that the immediate rewards of sharing private information (for example, in social networks, applications) outweigh the privacy concerns [23, 24]. Furthermore, ignorance about which data may compromise privacy is another important factor which determines if the user is comfortable sharing some data as was the case with anonymized audio recordings [22].

In an effort to mitigate some privacy threats, the EU has introduced the General Data Protection Regulation (GDPR). This regulation provides rules for data handling and dissemination. It puts significant restrictions on privacy sensitive data such as medical records and special restrictions for data of underage persons. Many of these restrictions are relaxed or simply do not apply in the context of scientific research. Furthermore, although there are specific rules for how data should be handled by corporations, there are not many restrictions on what they can collect. The major impact of GDPR is with how entities inform users about their data, how much control the user has over this data, and with restricting the dissemination of the data to entities which the user has not expressly allowed.

1.2 Research Questions

This thesis aims to answer the following research questions:

1. *Considering all the different sensor data that can be collected on a mobile ubiquitous device, both data collectors and participants must be made aware of what privacy threats come with sharing this data. For that, we must answer the question:* **Which sensor data originating from a mobile ubiquitous device has the potential to uniquely identify a person or to otherwise reveal sensitive personal information?**

2. *With the current trends on mobile ubiquitous device processing power and storage, the option to migrate tasks from the cloud to the edge to process data and derive useful and actionable information can be seriously considered.* **Is it feasible to do data mining and provide basic services, like localization, to users without transmitting sensitive data to a cloud service from the mobile device or otherwise rely on a third party?**

3. *Opportunistic crowd-sensing leverages the mobility of participants in order to opportunistically collect data from the environment. Some such crowd-sensing schemes break from the opportunistic paradigm when it comes to privacy measures during data reporting. Other schemes utilize the paradigm*

throughout the entire crowd-sensing framework, which includes the privacy-related tasks of data reporting. Even so, they fail to prove that such an opportunistic scheme can work in a real environment with real mobility data. **Can fully opportunistic crowd-sensing still be carried out for scientific research without compromising the privacy of individual participants?**

1.3 Methodology and Contributions

To answer our research questions we performed a literature review for privacy issues concerning sensor data, we implement some of our own algorithms for location self-provisioning, and we evaluate the efficacy of common mixing techniques in a fully opportunistic environment.

Unless otherwise stated, we focus on devices which operate on the Android OS. Our reasoning is that the vast majority of mobile devices are Android devices and the well documented API of the operating system enables us to take a closer look at all the features and capabilities of a device.

For the first research question, we look into the hardware capabilities of modern smartphones and into the Android OS API to see what sensor information can be accessed by an application. Then, for each sensor, we look in the literature for how it can be used to reveal private information.

For the second research question, we first look into the technological trends of mobile devices in terms of power and storage. We review secure multi-party computation techniques that can be used for data mining. Then, we propose three algorithms in order to enable service self-provisioning for location data and location-based services. One is designed to localize a person using trilateration with the Cell IDs without knowing their locations. The other two use Cell IDs to detect the location context of a person by either using graph based approach or a relative distance matrix approach.

We answer the third question by accumulating the answers and discussions of the previous two questions and also evaluating mixing strategies with real data by proposing a simple multi-party shuffling scheme in a fully opportunistic setting. The Mobile Data Challenge (MDC) data is used to analyze the interactions between users and the effectiveness of shuffling the data in an opportunistic way.

1.3.1 Publications

Parts of this thesis have been submitted, published, and presented at scientific journals and conferences. Table 1.1 lists these publications and the chapters which they contributed to.

Publication Title	Authors	Conf./Journal	Status	Chap.
Lightweight Clustering of Cell IDs into Meaningful Neighbourhoods	M. Fanourakis, K. Wac	HetNets 2013	Published [25]	5
ReNLoc: An anchor-free localization algorithm for indirect ranging	M. Fanourakis, K. Wac	WoWMoM 2015	Published [26]	4
mQoL: Experiences of the 'Mobile Communications and Computing for Quality of Life' Living Lab	K. Wac, M. Gustarini, J. Marchanoff, M. Fanourakis, C. Tsiourti, M. Ciman, J. Hausmann, G. Pinar	HealthCom 2015	Published [27]	
Differences in smartphone usage: Validating, evaluating, and predicting mobile user intimacy	M. Gustarini, M. P. Scipioni, M. Fanourakis, K. Wac	Journal of Pervasive and Mobile Computing 2016	Published [28]	5
Using Cell ID Traces to Discover Meaningful Places	M. Fanourakis, M. Gustarini, K. Wac		Under review	5
Efficacy evaluation of opportunistic data mixing with real mobility data	M. Fanourakis, K. Wac		Under review	6
Privacy Threats from Smartphone Sensor Data	M. Fanourakis, K. Wac		Under review	3

Table 1.1: Publications.

1.4 Thesis structure

The structure of this thesis is as follows:
In chapter 2 we will present previous work that is relevant to this thesis. There, we will present some data driven services, summarize the privacy metrics that are commonly used, the strategies to attack or safeguard user privacy, and the strategies to obfuscate data with the goal of having a privacy-conscious database. We also take a look at the technological trends of mobile ubiquitous devices. In chapter 3 we analyze sensor data collected from mobile devices in order to answer the first research question of this thesis and provide some insights for the second and third research questions of this thesis. In chapters 4 and 5 we describe methods to self-provide location or location context, one of the most difficult contexts to self-provide without the use of third party services or battery-draining sensors like a GPS. These chapters provide ample proof that service self-provisioning is achievable for a multitude of situations thus answering the second research question of this thesis. In chapter 6 we present a simple strategy for safeguarding user privacy during the data reporting phase in order to evaluate the efficacy of some data mixing techniques in the literature for opportunistic settings, which, in conjunction with other described state of the art methods, answers the third research question of this thesis.

Chapter 2

Related Work

This chapter will give an overview of topics that affect or are affected by our work in this thesis as well as present the current state of the art on the problems that we plan to address. In section 2.1 we will give an overview of how data is being used in almost every service. In section 2.2 we will summarize the state of the art of how privacy is measured. In section 2.3 we present the current popular methods to obfuscate data in such a way as to provide certain levels of privacy. In section 2.5 we will describe the latest techniques that provide privacy during the data collection phase. In section 2.4 we give an overview of privacy attacks that can be used on data. In section 2.6 we present the advantages and limitations of secure computation and machine learning (i.e. homomorphic encryption). Finally, in section 2.7 we describe the edge computing paradigm and note the rapid advancement of the resources available on smartphone devices throughout the years.

2.1 Data-dependent Service Provisioning

Various data can be used to provide a service. Step count can be used by an application that motivates the user to be more active. An application can use accelerometer data to count the number of repetitions during some exercise. There are also applications that keep track of the user's menstrual cycle and predict the next occurrence. A navigation application will use your location to calculate a route to your desired destination. These, and countless more, fundamentally rely on some data to perform their intended task. We focus on localization and location context since it is more relevant to the content of this thesis (see chapters 4 and 5).

Several solutions exist for providing user location context to mobile devices, all requiring data. A GPS based method in Sila-Nowicka et al. [29] identifies significant places by looking at frequency of re-occurrence and amount of time spent in a detected *stop segment* of a user's GPS trajectory. Using the GPS, which provides direct geographical location, is a sure way to detect such places, however, as we have mentioned previously, we consider that the GPS is off-limits due to its

power consumption and poor performance in indoor environments.

PlaceLab [30] is a commonly used database that maps radio frequency (RF) beacons like cell towers and WiFi APs to geographical locations. One can perform localization by comparing the beacons in range against the database and thus estimate a probable location. CellSense [31, 32] is a fingerprint based technique to map Cell IDs into locations in two phases: an offline phase where data is collected at certain positions such as Cell IDs in range and their signal strength (RSSI), and the localization phase where a reading of neighboring Cell IDs and their RSSI is mapped to an estimated position. Similar fingerprinting techniques are not uncommon, however, they are not scalable since they require a data collection phase and do not account for network infrastructure changes. Using either PlaceLab, CellSense, or any other localization technique requires some further analysis in order to determine significant places. A spatial clustering technique can be used to cluster locations near each other into places, or a time-based clustering technique can be used such as in Kang et al. [33] where they localized using PlaceLab. This technique uses both the distance and the time between location measurements to cluster them into places.

Kim et al. [34] remedy the issues in fingerprinting techniques such as CellSense with SensLoc. They use the beacon fingerprints directly (without localization) to determine significant places. Kim et al. compare beacon fingerprints in a sliding window using the Tanimoto similarity measure and are able to detect entrance and departure from a place as well as revisited places. Chen et al. [35] use an electromagnetic (EM) propagation model to determine the distance of WiFi APs from their RSSI in InferLoc. Using the calculated AP distances they then compare them in moving windows to determine similarities and places by doing some additional clustering. Both SensLoc and InferLoc use WiFi AP measurements in order to determine significant places. The shorter range of WiFi can provide good granularity indoors but unless there is at least one WiFi AP in range this algorithm becomes unreliable. These algorithms may be able to benefit from using Cell ID data in addition to WiFi, however the vastly different signal profile between indoors and outdoors may pose issues, especially with InferLoc which relies on an EM propagation model more suitable for outdoor environments. Furthermore the WiFi transceiver of mobile devices consumes significantly more power than the GSM transceiver. A survey of fingerprint based methods (signal based and motion based) by Vo et al. [36] gives an overview of such techniques and we can conclude that although some can provide relatively accurate estimations, the use of some or a combination of GPS, WiFi antenna, accelerometer, compass is common and some may use the camera.

There are techniques which rely solely on Cell ID data (Cell ID and timestamp tuples) to reveal significant places. Yadav et al. [37] make use of Cell ID oscillations for PlaceMap to determine graph edges and then cluster Cell IDs into significant places using an oscillation threshold parameter which is based on edge weight and another threshold parameter to identify star topology in the graph.

WSN localization. It is relevant to look at wireless sensor network (WSN) localization as well when it comes to localization in the context of ad-hoc self-localization. In a typical scenario there are mobile nodes which can measure

their distance to other nodes or to other anchor points and use this information to localize themselves.

In cases when the resources available do not allow for range measurements (or angle measurements), range-free techniques can be applied such as the LHDV-HOP algorithm [38] where number of hops were used as a distance measure with an assumption that the network was uniformly distributed. Another technique is the APIT algorithm [39] which localizes nodes by checking the triangular regions that the nodes reside in to infer a smaller region from intersections of those regions.

Many algorithms rely on anchor or beacon nodes which have knowledge about their absolute or relative position from either sensors such as a GPS or a database like PlaceLab [30]. In Ramadurai et al. [40] a probabilistic approach is described that localizes nodes by utilizing range measurements from several anchor nodes. In Srinath et al. [41] they have devised a method that only requires one mobile anchor node in order to localize neighboring nodes and in turn uses in-ranging technique to then localize nodes that are not directly communicating with the anchor node. In Doherty et al. [42] they model the network as a graph and use convex optimization techniques to localize unknown nodes with respect to anchors. Carter et al. [43] used semi-definite programming technique to create an accurate highly scalable distributed localization algorithm that also has a dependency on anchors. Such localization algorithms may need several anchors to localize which makes the setup more difficult and less adaptable. Furthermore, anchor nodes need to actively communicate their locations to other nodes, something that is not always possible.

A very interesting tool is multidimensional scaling (MDS), a technique from mathematical psychology. MDS can be used to calculate relative maps of nodes based on distances between the nodes. From a graph of a network one can estimate a distance matrix by calculating the shortest path between all pairs of nodes and then apply classical MDS on the distance matrix. In Shang et al. [44] they use MDS to build relative maps and then improve the estimations of the relative map by using a few anchors to perform linear estimation. They later modified their algorithm to be more efficient and scalable in their next work by breaking up the network into subproblems [45]. In general, MDS is commonly used for building relative maps for localization in conjunction with anchors as seen in Ahmed et al. [46] and Cheng et al. [47] among others.

In order to make a more adaptable solution some have developed completely anchor-free algorithms to localize nodes. One way to do this is to make assumptions about the distribution of the nodes. Fang et al. [48] assume that the nodes are deployed in clusters that are distributed as a Gaussian distribution. They localize the nodes by using a combination of maximum likelihood estimation or small area search in conjunction with gradient descent method. Velagapalli et al. [49] adapted this algorithm for applications in non-flat terrain. Although this approach is anchor-free, it needs the assumption of the distribution of the nodes which is not always representative of their true configuration. Another approach in Jin et al. [50] is using Ricci flow which overcomes many of the problems of MDS, however this is limited to 2D cases. In Wen et al. [51] they use particle filters and Markov chain Monte Carlo methods to localize, however it requires that all sensors are at fixed

locations and that all links between sensors are bidirectional.

2.2 Privacy Metrics

There are several measures of anonymity for data content and in this section we will summarize the most widely used.

k-anonymity. A protected data set is said to satisfy k-anonymity for $k > 1$ if, for each combination of key attributes, at least k records exist in the data set sharing that combination. It is able to prevent identity disclosure but, in general, is may fail to protect against attribute disclosure. For example, Imagine that an individual's health record is k-anonymized into a group of k patients with k-anonymized key attributes values $Age =$"30", $Height =$"$180cm$" and $Weight =$"$80kg$". Now, if all k patients share the confidential attribute value $Disease =$"$AIDS$", k-anonymization is useless, because an intruder who uses the key attributes (Age, Height, Weight) can link an external identified record ($Name =$"$John\,Smith$", $Age =$"31", $Height =$"179", $Weight =$"81") with the above group of k patients and infer that John Smith suffers from AIDS (attribute disclosure) [52–55]. An attempt to mitigate this shortcoming was made by introducing p-sensitive k-anonymity [56], however, it makes the assumption that each confidential attribute takes values uniformly over its domain and when this is not the case it may cause huge data utility loss [55].

l-diversity. A data set is said to satisfy l-diversity if, for each group of records sharing a combination of key attributes, there are at least l "well- represented" values for each confidential attribute [57]. Some criticisms of l-diversity are pointed out by Li et al. [58]. As with p-sensitive k-anonymity, it may be difficult to achieve with low utility loss. Furthermore, a *skewness attack* or a *similarity attack* can be used against it.

t-closeness. A data set is said to satisfy t-closeness if, for each group of records sharing a combination of key attributes, the distance between the distribution of the confidential attribute in the group and the distribution of the attribute in the whole data set is no more than a threshold t [58]. It solves the attribute disclosure vulnerabilities of l-diversity (skewness attack, similarity attack), however, there is no clear computational procedure to enforce the t-closeness property and it limits the utility of the data severely [55]. Li et al. [59] attempt to mitigate this shortcoming by further introducing (n, t)-closeness which offers a way to relax the criteria and increase the utility.

Others. There are several other anonymity measures [60,61] that are either new or enhancements of the above mentioned, however there is little consensus in the literature about which should be used. The two most commonly used are k-anonymity and l-diversity, despite their shortcomings, because they are easier to implement and verify than other more complicated measures.

2.3 Privacy for Personally Identifiable Data

There has been substantial research and development efforts to provide certain guarantees about data privacy being collected from mobile service users. Encryption implies encoding of the original data in such a way, that only authorized parties can read it is relatively straight-forward to integrate but does not necessarily guarantee privacy. Besides encryption, common strategies involve mixing the data between multiple users, which we will describe in more detail in section 2.5, or an introduction of some form of spatio-temporal noise or abstraction to add further privacy. Additionally, in some implementations, there is the need for a trusted central entity, i.e., trusted third party (TTP) that performs certain data anonymization or abstraction tasks before the data is forwarded to the other parties. Using TTPs means that a user has to trust that the TTP (server) is secured against outside threats and that it is not malicious in any way. There are also implementations that do these tasks in a distributed 'peer to peer' manner without the need for a TTP with additional strategies to mask the data [62–66].

Data perturbation techniques such as additive random noise, multiplicative noise, and random projections are widely used to safeguard privacy by modifying the original data. Although these techniques impact the utility of the data (some more than others), this trade-off is normally accepted in exchange for increased privacy [67–70]. Some of these techniques are vulnerable to various privacy attacks which we will summarize in section 2.4.

Rule-based sharing like the one described by Choi et al. [71, 72] allows a user define their own privacy rules specifying whether they want to allow or deny sharing of data based on conditions such as current context, location, timestamp, data consumer, and sensor data themselves. This concept is a valuable addition to frameworks that aim to preserve privacy since each user may have a different sense of privacy and it allows them to personalize it. That being said, basic privacy rules should always be in place since not all users are aware of how their data can compromise their privacy.

Privacy preserving queries in databases storing population data are also a topic of interest when it comes to the privacy preservation of individual users contributing their data to these databases. When an entity wishes to retrieve some data from a database, techniques such as differential privacy are used in order to preserve the anonymity of the retrieved user data from the perspective of that query entity [73, 74]. Although such a technique is mainly applied when a centralized server is involved, it can also have applications in a distributed setting.

Location privacy. One of the most privacy sensitive types of data is location [14, 75–77] so it is not a surprise that a great majority of the research on privacy preservation focuses on it. Using a variety of obfuscation and location abstraction techniques, k-anonymity or l-diversity can be achieved. Some of these techniques include adding artificial noise to the data so that the locations of different users are more likely to intersect and provide some anonymity in numbers (increasing the number of individuals having the same data reduces the probability of identifying a user from a set of data, for example, One person who frequents a specific ice cream parlor vs. multiple people being on the same street as the ice cream par-

lor) [62]. There are other techniques that provide anonymity in numbers such as geographic tessellation, where a geographical space is divided in regions that are visited by at least some number of users and then these regions are used instead of the actual locations. Rounding techniques or even peer to peer de-centralized methods to provide k-anonymity are also implemented [78–83]. Other techniques rely on mapping the location into an abstract space, which does not resemble the real world but retains some geographic properties like relative distances. Such a technique is the mapping of a space with a Hilbert curve [84, 85].

2.4 Privacy Attacks

Having applied one of the techniques of data obfuscation does not necessarily mean that the data is safe from attacks. There exists several methods to counteract data perturbations. We will describe the most widely known attacks in this section.

To counteract additive noise perturbation of the form $Y = X + R$ one can use several methods. Eigen-Analysis can be used when the degree of correlation between the original data attributes is high relative to the noise added. MAP estimation can be used if the data and noise arose from a normal distribution. Distribution analysis can be used if there is knowledge of the distribution of the un-perturbed data. For matrix multiplicative data perturbation of the form $Y = MX$ different techniques can be used. If the attackers has knowledge of some unperturbed data and their perturbed counterparts (known I/O) and M is orthogonal they can use linear algebra and measure theory to estimate the real data. If there is known I/O and the entries of M are generated independently from a 0 mean normal distribution then the attacker can use MAP estimation. If the attacker knows some of X and M is orthogonal they can use Eigen-analysis. If M has rank n and the data attributes are largely independent and at most one is Gaussian then the attacker can use independent component analysis (ICA). The same technique can be used if M is $n \times n$ and there is weak known I/O [86]. Some more techniques are summarized by Okkaliogly et al. [87].

Anonymized location traces are vulnerable to inference attacks. An attacker can find a person's home location with relatively high accuracy and doing a reverse lookup can identify some of the participants' names. Although spatial cloaking, noise, and rounding can be used to prevent such attacks, the utility of the data can be significantly reduced and render it unusable [88, 89].

Even differential privacy can be attacked somewhat. Coarse properties of the population taken together can combine to build a model that can be applied to individuals with high accuracy although this does not violate the privacy promises of differential privacy [90].

2.5 Privacy in Data Reporting

If only aggregate information is needed from a set of data, privacy-preserving data aggregation schemes have been proposed in order to safely provide aggregate

information such as value averages, minimums, and maximums about the underlying data using homomorphic encryption or other techniques [91–96]. These aggregate values cannot be used by a researcher or other entity to train machine learning models like neural networks or SVMs which most often require the data to not be aggregated. It is advantageous to do the least amount of manipulation on the data in order to retain as much utility as possible.

A common practice when performing studies is to use pseudonyms for the participants while keeping the data at its pure form. However it has been shown overwhelmingly that this does not necessarily guarantee the privacy of the participants [66, 77, 97]. Other approaches use mix networks or mix zones to reassign pseudonyms to the participants or to mix data. These methods have been shown to be an effective way to protect the participant's identity by decoupling the data from the user who collected it [79, 98, 99]. Mix networks are well studied and quite robust at what they are designed to do, which is to shuffle the batches of data (i.e. permuting the order of the batches with respect to the pseudonyms). The limitation of this approach is that the participants who generate the data often have to trust the mix network and additionally, for many mix network designs, individual data entries remain in the original batch so that when the identity of the participant is discovered for one piece of data from a batch then the rest of the batch can be assumed to belong to that same participant. Mix zones are fixed in space and require that participants enter these zones to satisfy certain privacy aspects like k-anonymity by guaranteeing that the data is mixed among k participants making them unsuitable for opportunistic settings.

A novel approach to data privacy is slicing and mixing. First developed for wireless sensor networks, it partitions the data horizontally and then mixes it before aggregating values (SMART) [100]. It has been adapted for privacy in data publishing for human-generated data by partitioning the data both vertically and horizontally where in the former, care is taken to group highly correlated attributes together. Then these *slices* are permuted in order to break the linking between different columns [101]. Many works have extended slicing to be used in participatory sensing scenarios for privacy-preserving data aggregation [91] but only a few have looked into how the mixing might perform in real world environment with mobility data from real people (for example, Qiu et al. [96, 102] use taxi traces to simulate participants) and none to our knowledge do an analysis of the effectiveness of mixing when it comes to opportunistic peer to peer (P2P) mixing scenarios.

2.6 Secure Computation Using Homomorphic Encryption

A very promising direction in privacy is that of performing computations on encrypted data. This can be realized using a special kind of encryption called homomorphic encryption. Depending on the computational capabilities, homomorphic encryption can be categorized as partially homomorphic, somewhat homomorphic, and fully homomorphic. Partially homomorphic encryption allows for addition or multiplication operations but not both. Somewhat homomorphic allows for mul-

tiple operations but a limited number of them. Finally, fully homomorphic allows for an unlimited number of additive and multiplicative operations on encrypted data.

The first fully homomorphic encryption scheme was proposed by Gentry in 2009 [103], making it a fairly recent development. Since then many more have been developed following the basic concepts of Gentry's initial breakthrough, however there are still several challenges to overcome to fully realize an efficient and robust fully homomorphic encryption scheme [104–107]. The three main challenges that keep fully homomorphic encryption from being practical are the large size of the ciphertext in relation to the raw data, the resulting processing time required to compute the operations, and the decryption time to retrieve the results. These issues are exacerbated as the amount of data to be computed increases making it highly impractical for data mining applications in big data (or even medium data).

On the other hand, partial homomorphic encryption can be used relatively efficiently for applications that require simple operations like addition or multiplication (but not both). Indeed, it has found its way into secure aggregation schemes for crowd sensing and data mining [108, 109].

Dowling et al. [110] have implemented CryptoNets, a neural network for optical character recognition that works on encrypted data using homomorphic encryption. They use a "leveled" homomorphic encryption which allows for multiplication and addition operations but requires that one knows in advance the complexity of the arithmetic circuit that will be applied on the data. Although the prediction time takes 250 seconds per prediction, several prediction can be performed in parallel resulting in a much higher throughput.

2.7 Edge Computing

The *edge computing* paradigm is a step forward toward privacy. The idea is that data can be kept at the edge of the network (user devices) rather than the core. Processing of the data and services can be realized at the edge and very few information needs to be communicated towards the core. On the contrary, cloud computing, collects the data towards the core of the network and most processing and services are realized there. The latter is the current state of commercial computing. Most services accessible through smartphone devices utilize the cloud architecture in order to free up resources on the device (thus extending battery life) and to keep information properly synchronized between different devices of the same user. Keeping such massive amounts of data in the cloud makes it vulnerable to data breaches as we have previously mentioned. Spreading the data out to where it would be more accessible to the devices that need it greatly reduces the chances of a data breach by increasing the effort and reducing the reward of such actions.

As mobile devices are becoming more and more capable in terms of processing and storage, the barriers that prevent the edge computing paradigm from being realized are becoming easier to surpass. We only need to look at the specifications of various smartphone devices throughout the recent past to notice this progress. Below, in table 2.1, we list the Google flagship phones since 2010. We

choose this line of smartphones because it is a good indicator of the performance of medium to high-end devices in the market.

Release Date	Model	CPU	Storage	RAM
January 2010	Nexus One	1GHz	512MB+	512MB
December 2010	Nexus S	1GHz	16GB	512MB
November 2011	Galaxy Nexus	dual-core 1.2GHz	32GB	1GB
November 2012	Nexus 4	quad-core 1.5GHz	16GB	2GB
October 2013	Nexus 5	quad-core 2.26GHz	32GB	2GB
October 2014	Nexus 6	quad-core 2.7GHz	64GB	3GB
September 2015	Nexus 5x	hexa-core 1.8GHz	32GB	2GB
September 2015	Nexus 6p	octa-core (4x1.95GHz, 4x1.55GHz)	128GB	3GB
October 2016	Pixel	quad-core (2x2.1GHz, 2x1.6GHz)	128GB	4GB
October 2017	Pixel 2	octa-core (4x2.35GHz, 4x1.9GHz)	128GB	4GB

Table 2.1: Google flagship phones throughout the years.

It is clear that there is a steady improvement on the processing and storage of this line of smartphones and we can be confident that they will keep improving in the near future. Furthermore, we can expect improvements in battery technology, especially considering the increasing popularity of electric vehicles which rely on them. These improvements will enable the migration of many services from the cloud to the edge.

Chapter 3

Privacy Threats from Smartphone Sensor Data

3.1 Introduction

An average smartphone is equipped with an abundance of sensors to provide a variety of vital functionalities and conveniences. For example, the basic telephony antenna which enables the smartphone to connect to the cellular network, or the ambient light sensor which helps to automatically adjust the screen brightness to a comfortable level. The data that these sensors provide pose no threat when used for their intended purpose.

With the advent of crowd sensing, this data is collected indiscriminately in order to find trends or discover interesting correlations in the data and are often kept in large databases where malicious entities can use it for nefarious purposes by revealing the identity of the persons who generated this data. For this reason, there has been a noticeable effort in the research community to develop methods and strategies to protect the privacy of the users while still being able to collect usable data from them. These methods can introduce limitations in the utility of the data and in, some cases, a non-negligible overhead in the overall data collection and data mining processes, therefore, it is advantageous to know which data has the potential to be a threat to a user's privacy so that only that data and no other is treated with the privacy-preserving methods that have been developed.

3.1.1 Contributions

In this chapter, we seek to identify what types of sensor data can be collected on a smartphone and which of those types can pose a threat to user privacy. We identify the data types by first looking at the hardware specifications of a typical smartphone and then looking at the Android API to see what information can be retrieved from this hardware. To determine the threat level of each type we look into the literature for how this data can be used (for example, in behavioural biometrics, inference attacks, behaviour modeling, etc.). Considering the large scope

of the topic we choose to focus on novel or recent work which is based on or improves upon older work.

3.2 Smartphone Data Types and Privacy

After analyzing hardware information of popular smartphones by Samsung, Google, LG, HTC, and Huawei we present in this section the most common sensors and data available. The information is presented in no particular order.

For certain data types we equate their potential for a privacy breach with their usefulness as authentication modalities. Our reasoning is that if a particular data type is unique enough to be used for authentication purposes, then it is certainly unique enough to identify a person in a large dataset. Furthermore, if a particular data type is significantly correlated with another data type which was evaluated to be a privacy threat, then we conclude that this particular data type must also be a privacy threat to some level.

Sampling Types For each data type we also note the type of sampling that is required to derive useful information. In table 3.1 we define several sampling types that can be used to collect information from sensors. We can use these sampling types to qualify the threat-sensitivity of a particular sensor's data. For example, if some sensor only requires Type A sampling, where a single sample is enough to derive some feature, then we can conclude that it a sensor which can easily compromise an individual's privacy and should avoid sharing any of its data or be extremely selective over which data to share and to whom. On the other hand, if a sensor requires Type F sampling, several samples can be shared without compromising the privacy of an individual.

Type ID	Sampling Type Description
Type A	Single sample is enough to derive feature confidently
Type B	Single sample is enough to derive feature somewhat confidently while several samples can improve confidence
Type C	Sampling only if there is a significant change is enough to derive feature confidently
Type D	Sampling at regular intervals during a day are enough to derive feature confidently
Type E	Continuous sampling for the duration of the action is enough to derive feature confidently
Type F	Continuous sampling without limitations is needed to derive feature confidently

Table 3.1: Sampling types.

3.2.1 GPS and location

The Android API package *android.location* is a comprehensive package that integrates GPS, WiFi APs, and Cell Identity information in a proprietary method to provide an accurate location estimate. The *Location* class in this package exposes methods to provide longitude with *getLongitude()*, latitude with *getLatitude*, the accuracy of the estimate with *getAccuracy()*, the altitude with *getAltitude()*, the bearing with *getBearing()*, and even the speed with *getSpeed()*.

Location data has received a significant amount of attention from the research community in the context of privacy. It is seen as a major threat to individual privacy and at the same time its utility is undeniable as evidenced by the vast number of location-based services available. Krumm [81] has outlined some of the threats posed by location data. Someone can infer significant places like home and work, and more recently, Do et al. [111] were able to reliably characterize 10 categories of places of a person's everyday life, these included home and work as well as friend's home, transportation, friend's work, outdoor sport, indoor sport, restaurant or bar, shopping, holiday. Krumm shows examples of how pseudonymized or anonymized location data can still be used to identify the people in the data. Other information such as mode of transportation (bus, foot, car, etc.), age, work role, work frequency, and even smoking habits can also be inferred from location data. The evidence for the privacy risks of location data is overwhelming. For the GPS sensor, Type A sampling is enough to reveal the location while type C and D is enough for detecting personal and significant places.

3.2.2 Telephony

The network antenna is used to connect to the cellular network (GSM, edge, HSPA, LTE, etc.). The Android API *android.telephony* package can be used to get information such as the identity of the cell tower which the phone is connected to (Cell ID) and the signal strength to this cell using the method *getAllCellInfo()* from the *TelephonyManager* class. This class can also provide the service state with *getServiceState()*, network type with *getNetworkType*, call state with *getCallState()*, and data state with *getDataState()*.

The Cell ID can be used in conjunction with publicly available data of their locations to localize a person as demonstrated by LaMarca et al. [30]. Although a single sample is usually enough to determine an approximate location, several samples might be needed to increase confidence (Type B sampling). As such, the same privacy threats as location can be applied here. However, even without knowing the location of the Cell IDs, one can infer places such as home and work as done by Yadav et al. [37] as well as our own work [25]. Furthermore, since a person's connection traces to Cell IDs is directly related to the person's location traces, the Cell ID traces can be thought of as a quasi identifier much like location. For this, Type C or D sampling is required.

3.2.3 Bluetooth

The bluetooth antenna is used to connect to nearby bluetooth devices such as wireless headphones or a smartwatch. The Android API *android.bluetooth.le* package can be used to get a list of nearby bluetooth devices using the *startScan(...)* method of the *BluetoothLeScanner* class. This method returns a list of class *ScanResult* which includes the hardware ID of the bluetooth devices with the *getDevice()* method and the signal strength with the *getRssi()* method.

Bluetooth connections to personal devices such as headphones and smartwatch are, in general, unique to each individual, as such, they can be used as identifying information. Bluetooth devices in range (not necessarily connected to) are not as unique but provided that some of those Bluetooth devices are geographically stationary then a frequent Bluetooth device scan can also be used to cruedly localize a person as demonstrated again by LaMarca et al. [30]. Type B sampling is recommended for localization, while Type C or D is required to detect personal or significant places.

3.2.4 WiFi Antenna

The WiFi antenna is used to connect to WiFi networks. The Android API *android.net.wifi* package can be used to get a list of WiFi access points (APs) using the *startScan()* method from the *WifiManager* class. This method returns a list of class *ScanResult* which include the AP identity in the *BSSID* public field and the signal strength in the *level* public field.

WiFi connections to personal access points (APs) such as someone's home or work, much like Bluetooth, can be unique for each individual. It has also been demonstrated that WiFi APs in range and their signal strength can be used to localize a person by LaMarca et al. [30] and Redzic et al. [112] among many others [36, 113]. The same sampling requirements as Bluetooth apply for WiFi.

3.2.5 Touchscreen

The touchscreen is the main input method on a smartphone, it is used to select items on the screen, to type text, or other gestures which are out of the scope of this work. The Android API package *android.view* includes the class *View.OnTouchListener* which can be used to capture touch events. For security reasons the location of the touch is only available to the application on the foreground, but the touch event itself can still be useful information.

The dynamics of touch events (time between touches, duration of touch, pressure, etc.) are categorized as *keystroke dynamics* and they have been researched heavily for authentication and user recognition for hardware computer keyboards and more recently for smartphones [114–116]. Frank et al. [117] show that touchscreen data like navigational strokes (a subset of keystroke dynamics since they do not include typing) cannot be reliably used for authentication as a standalone but provides useful authentication features nonetheless and using this kind of data for authentication is ultimately feasible. Antal et al. [118] and Roh et al. [119] among others [120] have shown that keystroke dynamics along with additional

features that can be collected on a smartphone (accelerometer, pressure, finger area) can be used to improve the performance of authentication. Continuous sampling for the duration of the keystrokes is required for their detection (Type E sampling).

3.2.6 Microphone

The microphone is used to capture audio to facilitate a phone call or to record audio. The Android API *android.media* package includes the class *AudioRecord* which can be used to capture the audio from the microphone. For the below mentioned exploitation methods of audio signals, continuous sampling is required for the duration of the action in order to apply the methods (Type E sampling).

The audio of someone speaking can be used to recognize them. Speaker recognition is a well researched topic, low level features like short-term spectrum and mel-frequency cepstral coefficients, voice source feature estimation, formant transitions, prosodic features, and high level features such as lexicon have been used in models like vector quantization (VQ), Gaussian mixture models (GMM), support vector machines (SVM), and neural networks [121]. More recently, with the advent of deep learning, more complex and robust modeling techniques have emerged [122, 123]. Speaker recognition has reached a high enough technological maturity level that it has found commercial applications in automated home assistants such as the Google Home.

Many human activities produce characteristic sounds which can be used to recognize them. Activities such as cooking, brushing teeth, showering, washing hands, urinating, shaving, drinking, etc. have been shown to be recognizable by the sounds they produce by several researchers [124–126]. More impressively, not only can someone recognize the activity of typing on a physical keyboard but also recognize what is being typed solely from the data of a microphone [127–129].

Environmental noise features from audio recordings can be used to identify the location of the recording. Acoustic environment identification (AEI), as it is commonly known, is mostly limited to room or enclosed space environments where the geometry of the room can have noticeable effects on the reverberation of the audio. The main applications of AEI are in audio forensics where an estimation of the reverberation and background noise from a recording can be used to identify the room or even the location inside a room where the audio was recorded [130–133]. Prior measurements or estimates of the impulse response of the rooms are required for these methods since they describe how the sound reverberates in that room.

Since the room geometry can affect the audio reverberation patterns of a room, someone could use an audio recording of a sharp noise (like a hand clap) to estimate the impulse response of a room and then estimate the dimensions or even the shape of the room [134–136]. These methods are often tested under controlled environments and with specialized audio equipment so it is unclear whether a recording from a smartphone microphone would be sufficient for meaningful results.

3.2.7 Camera

There are often two cameras on a smartphone, the front facing camera and the main camera on the backside of the phone. They are used to take pictures, video, and to facilitate video calls. The Android API *android.hardware.camera2* package provides the necessary methods to retrieve data from the camera.

Pictures or video from a camera can be used in a several different ways to reveal information about the user even without the use of the file metadata. The most obvious is if the subject of the picture is the user themselves or of people related, in the social sense, to the user. If the subject of the picture is a city, a street, or a landmark, algorithms can be used to match the pictures to a location provided there is a database of prior pictures in that location [137–139]. There are also algorithms that can recognize the style of an image and match it to a known photographer [140, 141]. Since a single picture is used in these cases, Type A sampling is enough. Videos can also be used with the aforementioned techniques by treating them as sequences of still images. In addition, analyzing the device movement from a video can also be used to identify the user similar to gait recognition in other behavioural biometric identification schemes [142]. Type E sampling is required for this.

3.2.8 Environmental and Activity Sensors

There is a variety of environment and activity sensors on smartphones. Their data is exposed in the Android API *android.hardware* package with the classes *SensorManager*, *Sensor*, and *SensorEvent*. Each sensor type is assigned an integer identifier constant with an appropriate name. Among these sensors are *software sensors*, that is, sensors that do not have a direct hardware counterpart but are calculated from the outputs of one or more hardware sensors. These sensors do not require any special permissions to be accessed which makes it easy for a rogue application or website to get this data without the user's knowledge.

Many of these sensors are based on microelectromechanical systems (MEMS) technology which has been shown to be vulnerable to sensor fingerprinting [143–147]. The accelerometer, gyroscope, magnetometer, and barometer are all based on MEMS technology. The idea behind sensor fingerprinting is that minor manufacturing defects give each sensor a unique output which is composed of the true reading (acceleration, magnetic field strength, etc.) plus the bias caused by the manufacturing defect. This makes it so that someone can discriminate the devices which produce a given sensor output. To achieve this, Type E sampling is required. In the sections below we will take a look at each individual sensor for their respective privacy threats which are additional to the aforementioned sensor fingerprinting.

Accelerometer The accelerometer (TYPE_LINEAR_ACCELERATION) is a hardware sensor that measures linear acceleration. It's main uses include adjusting the display orientation to match the orientation of the physical display and as a step counter among others. To derive other more interesting information besides the

orientation of the device, Type E sampling would be required. The accelerometer can be used in a variety of ways to become a threat to one's privacy. It has found uses in indoor localization systems where GPS is not available. Together with gyroscope and/or magnetometer readings it can help to accurately track the movement of a person [112, 148, 149]. It is often used for activity recognition as well (sitting, walking, running, biking, cleaning, shopping, sleeping, cooking, etc.) [150–153]. Its applications also extend into behavioural biometrics where gait recognition uses the accelerometer to recognize a person based on how they walk or move [115,116,154]. When coupled with touch event detection it has even been used to detect what is being typed on the touch screen [155]. Therefore the accelerometer can reveal not only location, but activity patterns throughout one's daily life, the identity of someone based on how they walk, and in some cases, even what they type on their smartphone. It has been shown that auditory vibrations can be picked up by the accelerometer on modern smartphones like the iPhone 4 or a Samsung Galaxy S4 and can be used to detect hotwords (short keywords or phrases that are often used to activate voice assistants) or even what is being typed on a physical keyboard nearby [156, 157].

Gyroscope The gyroscope (TYPE_GYROSCOPE) is a hardware sensor that measures the rotation or twist of the device. It is often used in conjunction with the accelerometer to measure the orientation of the device and to aid in navigation/localization schemes. Michalevsky et al. [158] show that sounds can affect the measurements of a gyroscope to such a level that private information about the phone's environment can be revealed such as who is speaking and to some extent, what is being said. Type E sampling is required for these methods.

Magnetometer The magnetometer (TYPE_MAGNETIC_FIELD) is a hardware sensor that is mainly used to measure the Earth's magnetic field for the purpose of navigation. It has found uses in indoor localization schemes by comparing the magnetic field to previously collected magnetic field fingerprints to localize a person [159–161]. These methods require Type E sampling and prior data collection to map the fingerprint to specific locations. It is not applicable for outdoor environments since these methods rely on the structural supports of building and rooms which produce these magnetic fingerprints. For outdoor environments it can only reliably measure the orientation of the smartphone with respect to the Earth's magnetic field.

Barometer The barometer (TYPE_PRESSURE) is a hardware sensor that measures the atmospheric pressure. Not all devices are equipped with this sensor. Barometric pressure varies depending on the weather and on altitude. Baring extreme weather events, the rate of change of barometric pressure due to weather is relatively slow (less than 0.04hPa per hour for steady weather, less than 0.5hPa per hour for slow weather changes, and up to 3hPa per hour for rapid weather changes). While in a city like Geneva, Switzerland where the highest altitude is 457m and lowest is 370m, one can expect a change of approximately 0.115hPa per meter of altitude change. Based on these crude estimates it is no surprise

that the barometric pressure is often used as an altimeter and with its inclusion in smartphones it has aided in indoor navigation algorithms to determine the floor that the person is on [148, 162–164]. As such, someone with access to barometer data can learn about the altitude or floor in which a person lives and works as well as altitude variations during their commute. The specific methods vary in their sampling from Type B to Type F. For a city with many altitude variations like Geneva, it does not seem out of the realm of possibility to be able to reconstruct the commute path of a person based on barometric data, it is something worth looking into.

Proximity The proximity sensor (TYPE_PROXIMITY) is a hardware sensor that measures distance. It is mainly used to detect when the user places the device next to their ear during a phone call so that the screen can be turned off in order to save power. In most cases the sensor has a very limited range of up to 5cm and only tells you if there is something near it (less than 5cm). As such, it is only useful to know if the phone is in a pocket, bag, or next to your ear when taking a call. It does not appear to have any immediate implications to privacy.

Ambient light The ambient light sensor (TYPE_LIGHT) is a hardware sensor that measures the intensity of light. It is mainly used to automatically adjust the screen brightness to a comfortable level. Ambient light during daytime varies significantly for indoor and outdoor locales, therefore, someone can easily detect this during the daytime using this sensor [165]. Type C or D sampling would be enough to detect when the user changes from indoor to outdoor throughout the day. Kayacik [166] and Micallef et al. [167] created temporal and spatial models for light sensor readings among other sensors and their results show that the light sensor readings are among the sensors with the highest similarity between users. Based on their results they conclude that, on its own, the light sensor is not sufficient for authentication. An interesting exploit of the ambient light sensor was revealed by Spreitzer [168] where they showed that by using variations in the ambient light due to slight tilting of the smartphone while inputting a PIN they can improve their chances of correctly guessing it. They used a corpus of 50 random PINs and allowed themselves 10 guesses and managed to have an 80% success rate compared to 20% if they randomly guessed. Type E sampling during the PIN entry was used. The ambient light sensor has also found a use in indoor localization. If one has control of the LED lighting in a room they can send detectable light variations to the phone and help it to localize itself in the room [169]. Mazilu et al. [170] have also shown that it is feasible to detect room changes solely based on the ambient light sensor readings. Both of these indoor localization methods require Type E sampling.

Gravity The gravity sensor (TYPE_GRAVITY) is a software sensor that provides the direction and acceleration due to gravity. It most commonly uses the readings of the accelerometer and the gyroscope. It is directly correlated with the physical orientation of the device. The main use of this software sensor is to remove the gravity component from raw accelerometer measurements and be able to use

those measurements for other tasks that require only the linear acceleration. On their own, the gravity measurements have very little utility and therefore do not pose any apparent threat to privacy.

Step The step sensor (TYPE_STEP_COUNTER, TYPE_STEP_DETECTOR) is a software sensor that detects when the steps a user makes when walking. It uses the accelerometer readings to derive the steps. When stride length is known (distance after one step) or accurately estimated from the height of a person, step counts can be used to estimate the distance that a person has walked [171–174]. Since only one sample is needed to derive the distance, Type A sampling is enough. Although there is significant error depending on what device is being used or even depending on the speed that a person is walking, someone can roughly determine the distances to nearby destinations where the user walks to. There are no significant privacy concerns for this data since the accuracy of these measurements can have significant errors over longer distances or even at different walking speeds.

3.2.9 Summary

In table 3.2 we summarize the possible threats of each sensor noting the type of sampling that is required. Location and location features seem to be a common type of threat for most sensors. In table 3.3 we summarize the literature which was used.

Sensor	Threat Summary	Sampling Reqs
GPS	location and personal places [81, 111]	Type A for location, Type C and D for personal places
Cell ID	location and personal places [25, 30, 33]	Type B for location, Type C and D for personal places
Bluetooth	location and personal places [30], identity (from connections to personal devices)	Type B for location, Type C and D for personal places and identity
WiFi	location and personal places [30, 36, 112, 113], identity (from connections to personal devices)	Type B for location, Type C and D for personal places and identity
Touchscreen	identity (keystroke dynamics [114–120])	Type E
Microphone	identity (speaker recognition [121–123]), activity [124–126], keylogger (for physical keyboard [127–129]), location features (AEI [130–133], room characteristics [134–136])	Type E
Camera	location and location features [137–139], identity (selfies, gait recognition from video [142], author recognition [140, 141])	Type A for static pictures, Type E for video
All MEMS	identity (MEMS sensor fingerprinting [143–147])	Type E
Accelerometer (MEMS)	location (indoor navigation [112, 148, 149]), activity [150–153], PIN [155], identity (gait recognition [115, 116, 154], speaker recognition [156, 157])	Type E
Gyroscope (MEMS)	identity (speaker recognition [158])	Type E
Magnetometer (MEMS)	location (indoor localization via fingerprinting [159–161])	Type E
Barometer (MEMS)	location features (floor detection [148, 162–164])	Type B up to Type F
Proximity	None	
Ambient light	location features (indoor vs outdoor [165], indoor navigation [169], room detection [170]), PIN [168]	Type C and D for indoor/outdoor/room features, Type E for navigation and PIN
Gravity	None	
Step	distance walked (estimated from number of steps [171–174])	Type A

Table 3.2: Summary of sensors and corresponding privacy threats

Citation	Sensors used	Derived information
[111]	GPS, WiFi, Bluetooth, App	Location of home, work, other personal places
Details	**Ground truth:** User annotated data. **Methodology:** Random forest classifier. **Accuracy & Limitations:** GPS features alone gave 70.3% accuracy, adding Wifi features to previous 71.7%, adding Bluetooth features to previous 74.6%, adding app features to previous 75%. Infrequently visited places are not reliably recognized.	
[30]	WiFi, Bluetooth, Cell ID	Map of radio beacons, location of user
Details	**Ground truth:** GPS war-driving or institution databases with location of radio beacons. **Methodology:** tracker component that models signal propagation and takes into account physical environment (for example, buildings). A probabilistic Bayesian particle filter can be used to increase accuracy. **Accuracy & Limitations:** lower accuracy than GPS.	
[33]	GPS, WiFi, Bluetooth, Cell ID	Location personal places
Details	**Ground truth:** user annotated data. **Methodology:** GPS or PlaceLab estimated location was used to collect traces. Time based clustering was used on location traces to find personal places. **Accuracy & Limitations:** Does not label the personal places.	
[25]	Cell ID	Detection of personal place
Details	**Ground truth:** GPS and user annotated data. **Methodology:** graph based clustering of Cell IDs using Cell ID transition matrix populated by Cell ID oscillation events. Duration of stay in clusters and time of day indicating home or work. **Accuracy & Limitations:** limited to urban environment with relatively dense cellular tower deployment. Does not detect places with shorter durations of stay.	
[112]	WiFi	indoor location
Details	**Ground truth:** ground truth. **Methodology:** Fingerprinting of RSSI of WiFi access points at specific calibration points (CPs) and using naive Bayes to identify the three nearest CP, then using interpolation driven by the likelihoods to find the location of the user in the vicinity of those CPs (even using as few as 2 of them). **Accuracy & Limitations:** Accuracy is around 2 meters which can be significant in indoor environments even though they showed that this method is better than many others. Requires calibration measurements in advance.	
[117]	touchscreen (navigational strokes)	user identity
Details	**Ground truth:** 41 users read text and compare images on an android phone to produce natural navigational strokes. **Methodology:** 30 behavioural touch features (for example, mid-stroke area covered, direction of end to end line, start/end x, start/end y, and more). From sets of highly correlated features, only one was selected. Used kNN and SVM classifiers. **Accuracy & Limitations:** 0% to 4% error (false negative and false positive combined) which is not ideal for authentication purposes. More subjects needed to improve feature selection. Differences of screen sizes of devices needs to be taken into account.	

Continued on Next Page. . .

Citation	Sensors used	Derived information
[118]	touchscreen	user identity
Details	**Ground truth:** 42 users. Android application with its own keyboard. Nexus 7 tablet (37 users) and LG Optimus L7 II p710 phone (5 users). Users input a password 30 times (same for all). **Methodology:** features: time between key press and release, time between consecutive key presses, time between key release and next press, pressure of press, finger area of press, averages of previous values. WEKA machine learning software was used. Analyzed several classifiers. **Accuracy & Limitations:** Best classifier was random forest with 82.53% accuracy using only time based features, and 93.04% accuracy using time based features and touchscreen based features together.	
[120]	touchscreen, accelerometer, gyroscope, magnetometer	user identity
Details	**Ground truth:** 100 users typing 3 answers of at least 250 words under sitting or walking conditions. Sensor sampling at 100Hz. **Methodology:** Scaled Manhattan (SM), scaled Euclidean (SE), SVM verifiers using hand movement, orientation, grasp (HMOG) features, tap and keystroke dynamics features. **Accuracy & Limitations:** Best verifier was SM with Equal Error Rate of 10.05% for sitting and 7.16% for walking. Including HMOG features improved accuracy over only tap or keystroke dynamics. Cross-device interoperability and varying walking speeds were not explored.	
[125]	microphone	Human activity (cleaning, brush teeth, walk, drink water, etc.)
Details	**Ground truth:** Sound recordings of each activity **Methodology:** 5 random segments of 1.5 second from recording were used. Mel Frequency Cepstral Coefficients (MFCC) were extracted for each segment. Discrete time warping was used to get closest match. **Accuracy & Limitations:** Average accuracy of recognizing each of the 14 activities was 92.5% (80% lowest, 100% highest). Sound samples were recorded in a controlled environment, realistic data would improve argument.	
[129]	microphone	text typed on physical keyboard
Details	**Ground truth:** 10 minute recording of user typing in English **Methodology:** Compute Cepstrum features of each keystroke. For training, use clustering technique to separate into classes and language model correction based on HMM to label and then train a classifier. For recognition, use classifier and language model correction. **Accuracy & Limitations:** 90% of 5-character passwords in fewer than 20 attempts, 80% of 10-character passwords in fewer than 75 attempts. Classifiers user: linear classification, Gaussian mixtures, or Neural Network.	
[132]	microphone	environment (room)
Details	**Ground truth:** 30 audio recordings in 6 different acoustic environments (big classroom 1 and 2, small classroom, small seminar hall, seminar hall, small room) **Methodology:** Blind de-reverbaration was used to extract reverberant component of audio. Impulse response was estimated via hand-clap method. MFCCs were used as features, a multiclass SVM was used for classification. **Accuracy & Limitations:** 4 rooms identified with 100% accuracy, 2 rooms above 80% accuracy. Need to measure impulse response of rooms separately. Environments were based only on university campus.	

Continued on Next Page. . .

Citation	Sensors used	Derived information
[135]	microphone	room dimensions
Details	**Ground truth:** simulations of rectangular and L-shaped room. **Methodology:** Defined a cost function robust against wrong matches of TOAs. Genetic algorithm was used to minimize cost function and derive room dimensions. **Accuracy & Limitations:** room dimensions for rectangular room are within 10cm of actual size, for L-shaped room within 70cm. Should be repeated in real room. Room shape known a priori.	
[139]	camera (photo)	location
Details	**Ground truth:** 126M photos with Exif geolocations from the web. **Methodology:** Used convolutional neural network (CNN) to train with 91M images, the rest used for validation. 237 geotagged Flickr photos used to measure accuracy of model. **Accuracy & Limitations:** When using any type of photo accuracy is 8.5% for 1km radius, 24.5% for 25km, 37.6% for 200km, 53.6% for 750km, 71.3% for 2500km. Using other contextual info increased accuracy.	
[142]	camera (video)	identity
Details	**Ground truth:** 32 users recorded two 7 minute sequences with head-mounted cameras. **Methodology:** optical flow vectors computed for each frame. CNN with 2 hidden layers for classifier. **Accuracy & Limitations:** 77% accuracy for 4 second of video, 90% accuracy for 12 seconds of video. Stabilizing the video deteriorated results. Requires that camera be mounted on person. Should consider hand-held camera.	
[144, 146, 147]	MEMS (accelerometer, gyroscope, magnetometer)	device identity
Details	**Ground truth:** 3 devices on a robotic arm and moved in a predetermined pattern. For magnetometer, 9 devices were tested, a solenoid was placed around each device and a predetermined signal was produced. **Methodology:** SVM classifier was used with different kernel functions. **Accuracy & Limitations:** Over 95% accuracy to distinguish between different models, over 65% accuracy overall. The inputs to the sensors were controlled. This might not be possible to apply with data collected in the wild.	
[151]	accelerometer	human activity (walking, jogging, ascending stairs, descending stairs, sitting, standing)
Details	**Ground truth:** 29 users performing each activity several times while carrying a smartphone. **Methodology:** Split data into 10 second segments, each segment extracted features like average acceleration, standard deviation, time between peaks, etc. WEKA with decision trees (J48), logistic regression, multilayer neural networks (NN) with default settings. **Accuracy & Limitations:** NN is best with an average of 91.7% accuracy. Up/down stairs had the worst accuracy as low as 44.3% and were most often confused with each other or walking. Activity set is limited, different carrying patterns of device not taken into account.	

Continued on Next Page. . .

Citation	Sensors used	Derived information
[155]	accelerometer, gyroscope	smartphone keyboard input
Details	**Ground truth:** 10 users using a custom application for tapping icons and typing text (each letter 50 times, 19 different pangrams, and 20 times the same pangram). **Methodology:** Detect taps and extracts features of each tap (time domain and frequency domain). kNN, multinomial logistic regression, SVM, random forests, bagged decision trees are all used together in an ensemble classifier. **Accuracy & Limitations:** 90% accuracy for inferring tap locations, 80% accuracy for letters. The classifier is resource heavy and could have redundancies.	
[156]	accelerometer	text typed on physical keyboard
Details	**Ground truth:** iPhone placed on same surface as keyboard. Sentences typed were selected from the Harvard Sentences corpus. **Methodology:** Features from keypress data were used like root mean square, skewness, variance, kurtosis, FFT, MFCCs. Two neural networks were trained with with a difference in features used. **Accuracy & Limitations:** Tested with and without dictionary knowledge and with a news article from a newspaper. As much as 80% accuracy of typed content with the use of dictionary. Orientation of device, desk surface material, typing speed, ambient vibrations can affect the performance.	
[157]	accelerometer	hotword detection (for example, "okay Google")
Details	**Ground truth:** 10 users recorded saying "Okay google" and 20 common short phrases 10 times each. Each recording played through phone speakers 10 times for training at 70dB. **Methodology:** 2 second window is used and time domain and frequency domain features are extracted. For mobile scenario, a high pass filter with 2Hz cutoff is used to remove effects due to movement. A decision tree classifier is used. **Accuracy & Limitations:** 85% in static scenario, 80% in mobile scenario. The mobile scenario is very limited with just a controlled walking. More complicated movements make it vastly more difficult to recognize the hotwords.	
[158]	gyroscope	identity, speech
Details	**Ground truth:** Nexus 4, Nexus 7, Samsung Galaxy S III were used. A loudspeaker at 75dB. TIDIGITS corpus was used (recordings of 10 users speaking the 11 digits twice each). **Methodology:** 10-30ms sliding windows with time domain features and MFCCs and Short Time Fourier Transform (STFT). SVM, GMM, DTW were used as classifiers. **Accuracy & Limitations:** Over 80% accuracy for gender ID using SVM. Speaker ID ˜50% accuracy using DTW with Nexus 4 but 17% with Samsung. Speaker-independent word rec performed poorly, but improved to 65% using DTW with speaker-specific models. Results varied significantly between devices.	
[160]	magnetometer	location (indoor)
Details	**Ground truth:** Magnetic field map data collected by following serpentine pattern in a room on x-axis and then on y-axis. Test data collected following a well defined straight line, or circle path. **Methodology:** Magnetic field map data was used to generate a map of the field in the room. Test data was then matched to the map using a particle filter. **Accuracy & Limitations:** Within 0.7m of ground truth. Wi-Fi was used to get coarse location as initial condition for particle filter.	

Continued on Next Page. . .

Citation	Sensors used	Derived information
[161]	magnetometer	location (indoor)
Details	**Ground truth:** 2 users with HTC Nexus One phone. Magnetic fingerprints collected in hallways of campus buildings as users walked along the walls and pillars. **Methodology:** Magnetic field fingerprints were collected and then DTW was used on test data to match to the fingerprints. **Accuracy & Limitations:** Hallways were detected with over 90% accuracy after only less than 5 meters of walking. Users were instructed to walk close to objects that influence magnetic fields like pillars.	
[164]	barometer	location (floor in building)
Details	**Ground truth:** 63 trials at 5 different tall buildings in New York City where barometric pressure was recorded and random floors were selected. The user could choose either the staircase or elevator. **Methodology:** Calculated the change in height based on the international pressure equation. To resolve to a floor number they used calculated clusters derived from data of all visits to building (floor height could be estimated). **Accuracy & Limitations:** 65% accuracy when floor height is not known and a default 4.02m was used (98% within 1 floor), 100% accuracy if floor height has been previously estimated.	
[170]	ambient light sensor	location (indoor room detection)
Details	**Ground truth:** 3 users with Samsung Galaxy S4 collected data in their homes. Users logged room label each time they entered a new room on paper-based diary. Total of 132 hours of data **Methodology:** If sensor data feature was higher than a fixed threshold then a room change was detected. Decision trees (C4.5) were used for room identification. **Accuracy & Limitations:** Using only light sensor, accuracy was around 50%, with additional sensors like temperature and humidity the accuracy was above 60%. Random guess was 25% accuracy at best. Time of day, weather, and open windows affected the performance.	
[168]	ambient light sensor	PIN
Details	**Ground truth:** Samsung Galaxy SIII was used. 29 test runs by 10 users who entered 15, 30, or all 50 of the random PINs from 3 to 10 times. **Methodology:** Multiclass logistic regression, discriminant analysis, and K-nearest neighbor methods were used on the collected data with only light intensity and with additional RGBW information which modern light sensors include. **Accuracy & Limitations:** 80% success after 10 guesses from a set of 50 PINs. The set of 50 PINs is unrealistic as there many more possible combinations.	

Table 3.3: Summary of selected state of the art

3.3 Discussion

After reviewing each data type in section 3.2 we conclude that most of them can be used on their own to reveal something about a user be it small (for example, the floor on a building) or big (for example, the location of their home and work). Combining different data types can enhance the precision, or accuracy, or both as evidenced by several of the surveyed research in table 3.3.

On the Android OS there is a permission framework to enable an application

to explicitly request from the user if a certain data type can be used or not. Permissions that have a protection level of *normal* are automatically granted by the system while those that have protection level *dangerous* require the user's explicit permission to be allowed. In the older versions of the Android OS, the permissions were requested in batch when installing an application but on the latest Android OS version the permissions are requested individually and on an as-needed basis during the runtime of the application (i.e. until the application needs to use the microphone it will not ask for permission). Furthermore, on the latest Android OS version, a user can adjust the individual data type permissions in the settings for each application after the fact. Consequently, the user is informed about the various data types that an application uses. The permission framework does not cover the sensors in section 3.2.8 at this moment and it is unclear if it will in the future.

Data type	**permission**	**prot. level**	**comments**
Location (precise)	ACCESS_FINE_LOCATION	dangerous	
Location (approximate)	ACCESS_COARSE_LOCATION	dangerous	
Network Cell ID	ACCESS_COARSE_LOCATION	dangerous	
Bluetooth APs	BLUETOOTH_ADMIN	normal	
WiFi APs	ACCESS_WIFI_STATE	normal	
Touchscreen	No permissions are required	N/A	Touch event only outside application window or touch location available only to the application in foreground
Microphone	RECORD_AUDIO	dangerous	
Camera	CAMERA	dangerous	
Accelerometer	No permissions are required	N/A	
Gyroscope	No permissions are required	N/A	
Magnetometer	No permissions are required	N/A	
Barometer	No permissions are required	N/A	
Proximity	No permissions are required	N/A	
Gravity	No permissions are required	N/A	
Step	No permissions are required	N/A	

Table 3.4: Data types and corresponding Android OS permission requirements and protection level

Christin et al. [66] summarize countermeasures to several privacy threats: tailored sensing and user preferences, pseudonymity, spatial cloaking, hiding sensitive locations, data perturbation, data aggregation, among others. They also present important research challenges in this field that have yet to be fully addressed.

This and most such privacy research are concerned with threats in the context of data collection campaigns for research and data mining but similar principles

must also be applied to commercially available smartphone applications. Each year Google has to remove more and more malicious applications from their marketplace amounting to hundreds of thousands of applications [175, 176]. Although a lot of malicious applications are automatically filtered, some may still slip through and become available for millions of people to install. Some of these can easily collect data such as location from unsuspecting users even if they have to request the specific permission from the user. So many applications require the location permission that a user might not think twice about allowing it. In table 3.5 we list 25 of the top installed Android applications [177] along with some of the relevant permissions that are required to fully operate them [178]; the location permission is very common.

Application	**Category**	**Permissions**
Facebook	Social	Location, Camera, Microphone, WiFi
WhatsApp	Communication	Location, Camera, Microphone, WiFi
Messenger (Facebook)	Communications	Location, Camera, Microphone, WiFi
Subway Surfers	Game Arcade	WiFi
Skype	Communication	Location, Camera, Microphone, WiFi
Clean Master	Tools	Location, Camera, Microphone, WiFi
Security Master	Tools	Location, Camera, Microphone, WiFi
Candy Crash Saga	Game Casual	WiFi
UC Browser	Communication	Location, Camera, Microphone, WiFi
Snapchat	Social	Location, Camera, Microphone, WiFi
My Talking Tom	Game Casual	Microphone, WiFi
Twitter	News & Magazines	Location, Camera, Microphone, WiFi
Viber Messenger	Communication	Location, Camera, Microphone, WiFi
LINE	Communication	Location, Camera, Microphone, WiFi
Pou	Game Casual	Microphone, WiFi
Super-Bright LED Flashlight	Productivity	Camera
Temple Run 2	Game Action	WiFi
SHAREit	Tools	Location, Camera, WiFi
imo free video calls and chat	Communication	Location, Camera, Microphone, WiFi
Microsoft Word	Productivity	Camera, WiFi
Flipboard: News For Our Time	News & Magazines	
Clash of Clans	Game Strategy	WiFi
Spotify Music	Music & Audio	Camera, Microphone, WiFi
Shadow Fight 2	Game Action	WiFi
Pokemon GO	Game Adventure	Location, Camera

Table 3.5: Top downloaded apps excluding pre-installed and system applications. Only the following permissions were noted: Location, Camera, Microphone, and WiFi.

Users should always question if an application really needs a specific permission to function. For example, location can be used for navigation, to check-in to places in social media, to show local weather, to share your location with a contact, for fitness tracking, to show location-based notifications, and many more. The issue arises when an application does not need a precise location (for example, a weather application) or only needs some tracking information (for example, a fitness application). A user should not need to give more information than is needed for an application to function much like when a stranger asks for your

contact information you have the choice of giving them any of the following information depending on the intimacy level: first name, last name, email address, phone number, home address, work address, friend's address, parents' address, frequented bars, frequented shops, parents' first and last names, etc. Technically, all of these pieces of information can be considered *contact information* but you would not give out all of them if you only need to give out a first name and an email address for example. Sharing more than is necessary can feel highly intrusive. Therefore, a weather application should only get a meteorological region instead of exact coordinates, and a fitness application should only get distance and speed instead of the exact coordinates of the path you ran. There is currently no mechanism on Android OS or any other popular smartphone operating system that provides this level of abstraction when it comes to location information, location context information, or most other types of data that can be collected on a smartphone device.

This chapter reviewed the privacy threats of many data types available on a smartphone but there are still more that need to be scrutinized. Examples include other sensors like CPU temperature, and battery state data or application usage, screen state (on/off), TCP connection information. Furthermore, it is important to analyze current privacy measures to determine if they are enough to protect users from each of the threats described here or additional measures are required.

Chapter 4

Location Tracking Service Self-Provisioning

4.1 Introduction

Localization has many uses in both wireless sensor networks and ubiquitous computing. In wireless sensor networks it is often desired to know the locations of nodes in the network for several reasons: routing information efficiently through the network, mapping sensor readings and building models, navigating robots in unknown terrain, and others. One can expect similar uses in ubiquitous computing as well as some specific uses such as enabling location-aware services, tracking movement behaviors, modeling social interactions, and more. Almost all of these tasks are realized through the use of the GPS sensor available in the majority of personal ubiquitous devices, like smartphones, but several of these tasks do not require geographic location traces such as the ones provided by the GPS. For example, an application which is tracking the distance that a user has walked or ran only needs the distance travelled and not the precise geographic path the the user took. A common solution to this is to use a step counter to estimate the distance traveled which has the additional advantage of not requiring much power but distance estimation from step count has many variables that can result in large errors (gait distance of individuals, walking speed variations, and more). There are also situations where the GPS is not a reliable localization method such as when the user is indoors. This problem has motivated many different solutions, as we have previously seen in chapter 2, which are designed considering factors such as effectiveness, accuracy, power consumption, reliance on infrastructure, robustness, and ease of deployment. Very few address the aspect of privacy and none of them address privacy, ease of deployment, and power consumption simultaneously.

4.1.1 Contributions

In this chapter we introduce ReNLoc(**Re**laid ra**N**ging **Loc**alization), a range-based self-localization algorithm that addresses the issues of power consumption, reliance on infrastructure, and ease of deployment while keeping the number of assumptions about the network to a minimum. At the same time we satisfy basic privacy requirements regarding location since no geographic data needs to be communicated for the non-distributed version of this algorithm; a singular mobile node can use its own measurements to self-localize without any additional information. ReNLoc operates in an environment with three or more stationary base nodes where one or more mobile measuring nodes, able to measure their distance to base nodes and only to base nodes, use all the measurements collected from all the measurement nodes (centralized version) or only their own and neighboring measurements (distributed version) to localize themselves and the base nodes. Section 4.2.1 describes the setup of ReNLoc in detail. In section 4.3.2 we describe the methodology for the estimations in ReNLoc. In section 4.3.4 we describe the ReNLoc algorithm. In section 4.4 we show the results of simulations with ReNLoc. Lastly, in section 4.5 we make our conclusions and describe future work areas.

4.2 ReNLoc

4.2.1 Problem definition

Our approach in this chapter is tailored to networks that have two types of nodes, base nodes and measuring nodes. We wish to localize the measuring nodes, however, these nodes cannot measure their range to other measuring nodes but only to the base nodes. In the approaches that we described in chapter 2 section 2.1, nodes can communicate and range indiscriminately to any other node. In that sense, the nodes are of the same type in these approaches. Our approach is especially convenient in ubiquitous computing settings where we can readily relate the mobile ubiquitous device to the measuring nodes and the access points (APs) to the base nodes.

Let there be stationary **base nodes** and mobile **measuring nodes** that can measure their distance to the base nodes but not to other measuring nodes. The base nodes can only communicate with measuring nodes and differ from the standard anchor nodes in that they have no knowledge about their position aside from being stationary. The goal is for the measuring nodes to localize themselves relative to each other.

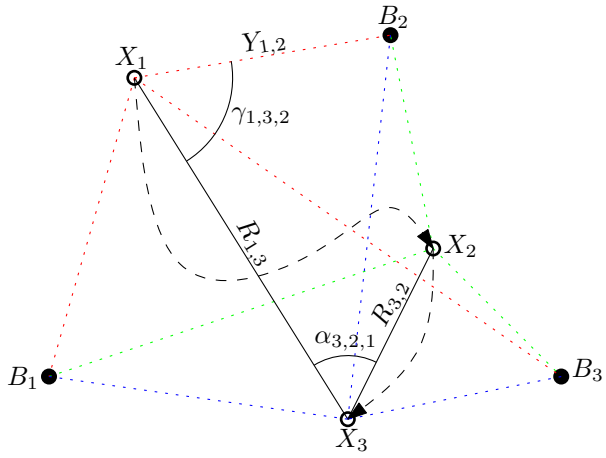


Figure 4.1: Three base nodes B_1, B_2, and B_3 along with the measuring node X for which we have distance measurements at X_1, X_2, and X_3.

The assumptions and requirements of our problem setup are as follows:

1. The base nodes are stationary (do not move)

2. There is a way to measure the distance between the measuring nodes and the base nodes

3. There are at least 3 non-colinear base nodes (for 2 dimensional problem) or 4 non-coplanar base nodes (for 3 dimensional problems)

4. There are at least 3 non-colinear measurements (for 2 dimensional problem) or 4 non-coplanar measurements (for 3 dimensional problems)

Let there be a total of N measurement instances and K base nodes. Let the matrix Y be an $N \times K$ matrix that holds the distances between the measuring nodes X_n and all stationary base nodes B at different measurement instances of X_n:

$$Y = \begin{bmatrix} \|X_1 - B_1\| & \|X_1 - B_2\| & \cdots & \|X_1 - B_K\| \\ \|X_2 - B_1\| & & & \\ \vdots & & \ddots & \\ \|X_N - B_1\| & & & \|X_N - B_K\| \end{bmatrix} \tag{4.1}$$

We will treat each measurement instance of measuring node X_n as a separate node, i.e. the m_{th} measurement of node X_n will be treated as a node X_w. So then $Y_{n,i}$ is the distance of the base node B_i from measurement node X_n. Each row of the matrix Y corresponds to one measurement instance, so lets denote then Y_n as the n_{th} row of Y which represents the n_{th} measurement instance. The matrix Y is the only information we have about this problem.

4.3 Distance estimates from a consensus of Y

Assuming that the base nodes are stationary, it becomes possible to calculate the distance between any pair of base nodes using only the distance measurements between each base node and the moving node X.

For example lets calculate $D^n_{i,j}$ the distance between B_i and B_j using measurement X_n:

$$\begin{aligned} D^n_{i,j} &= \left\| Y_{n,i} - Y_{n,j} \begin{bmatrix} cos(\theta) \\ sin(\theta) \end{bmatrix} \right\| \\ &= \sqrt{Y^2_{n,i} - 2Y_{n,i}Y_{n,j}cos(\theta) + Y^2_{n,j}} \end{aligned} \tag{4.2}$$

Since we do not know the angle θ we cannot calculate the distance between any pair of base nodes, but we can give a possible range of the distance by taking the minimum and the maximum of the result in this equation, giving us:

$$Dmin^n_{i,j} = min(D^n_{i,j}) = |Y_{n,i} - Y_{n,j}| \tag{4.3}$$

$$Dmax^n_{i,j} = max(D^n_{i,j}) = |Y_{n,i} + Y_{n,j}| \tag{4.4}$$

Putting all inter-base node minimum and maximum distances in a matrix for each measurement n:

$$Dmin^n = |(Y_n \otimes \mathbf{1}_M)' - (Y_n \otimes \mathbf{1}_M)| \tag{4.5}$$

$$Dmax^n = |(Y_n \otimes \mathbf{1}_M)' + (Y_n \otimes \mathbf{1}_M)| \tag{4.6}$$

Y_n denotes the nth row of matrix Y which contains all node to base node distances from measurement n. It is important to note that $Dmin^n$ and $Dmax^n$ are symmetric matrices and the values on the diagonal can be replaced by zeros since the distance of anything to itself is zero.

Aggregating these ranges from all the measurements can decrease the possible distance range and make the estimation more accurate:

$$Dmin_{i,j} = max(Dmin^n_{i,j}) \text{ for all } n \tag{4.7}$$

$$Dmax_{i,j} = min(Dmax^n_{i,j}) \text{ for all } n \tag{4.8}$$

Putting these values in two (symmetric) matrices $Dmin$ and $Dmax$:

$$Dmin = \begin{bmatrix} 0 & Dmin_{1,2} & \cdots & Dmin_{1,M} \\ Dmin_{2,1} & & & \vdots \\ \vdots & & \ddots & Dmin_{M-1,M} \\ Dmin_{M,1} & \cdots & Dmin_{M,M-1} & 0 \end{bmatrix} \tag{4.9}$$

$$Dmax = \begin{bmatrix} 0 & Dmax_{1,2} & \cdots & Dmax_{1,M} \\ Dmax_{2,1} & & & \vdots \\ \vdots & & \ddots & Dmax_{M-1,M} \\ Dmax_{M,1} & \cdots & Dmax_{M,M-1} & 0 \end{bmatrix} \tag{4.10}$$

Where $Dmin_{i,j}$ denotes the minimum distance between base node i and base node j and $Dmax_{i,j}$ denotes the maximum distance between base node i and base node j.

4.3.1 More constraints with the triangle inequality

The triangle inequality states that, for any triangle, the sum of the lengths of any two sides must be greater than or equal to the length of the remaining side. Suppose we have a triangle with side lengths A_1, A_2, and A_3 then:

$$A_1 \leq A_2 + A_3 \tag{4.11}$$

In our case, lets define a triangle of base nodes i, j, and k. Then we have the inter-base node distance ranges $Dmin_{i,j}$ to $Dmax_{i,j}$, $Dmin_{i,k}$ to $Dmax_{i,k}$, and $Dmin_{k,j}$ to $Dmax_{k,j}$. Using the triangle inequality we can say the following:

$$\begin{aligned} Dmax_{i,j} &\leq Dmax_{i,k} + Dmax_{k,j} \\ Dmax_{i,k} &\leq Dmax_{i,j} + Dmax_{k,j} \\ Dmax_{k,j} &\leq Dmax_{i,j} + Dmax_{i,k} \end{aligned} \tag{4.12}$$

And

$$\begin{aligned} Dmin_{i,j} &\geq max(|Dmax_{i,k} - Dmin_{k,j}|, |Dmax_{k,j} - Dmin_{i,k}|) \\ Dmin_{i,k} &\geq max(|Dmax_{i,j} - Dmin_{k,j}|, |Dmax_{k,j} - Dmin_{i,j}|) \\ Dmin_{k,j} &\geq max(|Dmax_{i,j} - Dmin_{i,k}|, |Dmax_{i,k} - Dmin_{i,j}|) \end{aligned} \tag{4.13}$$

4.3.2 Applying the geometric constraints

By looking at disparities between pairs of measurements Y_n we can get some information about the location of the base nodes relative to the measurement positions and the location of the measurements in relation to each other.

Calculating the inter-measurement distances

To do this we first calculate the distance range for each pair of measurements, the distance between measurement node X_n and X_m. We will label this distance $R_{n,m}$ and the set describing its possible values as $\mathcal{R}_{n,m}$.

$$\begin{aligned} R_{n,m} &= \|X_n - X_m\| = \\ &= \left\| Y_{n,i} - Y_{m,i} \begin{bmatrix} cos(\theta) \\ sin(\theta) \end{bmatrix} \right\| = \\ &= \sqrt{Y_{n,i}^2 - 2Y_{n,i}Y_{m,i}cos(\theta) + Y_{m,i}^2} \\ \mathcal{R}_{n,m} &= [|Y_{n,1} - Y_{m,1}|\,,\, Y_{n,1} + Y_{m,1}] \cap \cdots \\ &\quad \cdots \cap [|Y_{n,K} - Y_{m,K}|\,,\, Y_{n,K} + Y_{m,K}] \end{aligned} \tag{4.14}$$

Where we notice that $min\left(\mathcal{R}_{n,m}\right)$ is greater than or equal to 0.
If $min\left(\mathcal{R}_{n,m}\right)$ is equal to 0 then this indicates that the two measurements Y_n and Y_m are the same and therefore we can discard one of them for the equations in this section where $R_{n,m}$ is on the denominator. This will guarantee that $min\left(\mathcal{R}_{n,m}\right)$ will always be strictly greater than 0, an important point which allows for a solution to a few of the equations in this section.

Calculating the γ angles

Now we can find where the base node measurements agree for each pair (Y_n, Y_m) as we move X_m from $min\left(\mathcal{R}_{n,m}\right)$ to $max\left(\mathcal{R}_{n,m}\right)$. This will allow us to calculate the possible range of the angle $\gamma_{n,m,i}$, the angle between the base node B_i and the node-to-node vector $\vec{V}_{n,m}$ given by nodes X_n and X_m. Given $R_{n,m}$, in order to determine $\gamma_{n,m,i}$ we first find where the measurements $Y_{n,i}$ and $Y_{m,i}$ intersect:

$$f_{n,m,i}(r) = \frac{Y_{n,i}^2 - Y_{m,i}^2 + r^2}{2r} \tag{4.15}$$

And then use the inverse cosine:

$$\gamma_{n,m,i}(r) = \pm cos^{-1}\left(\frac{f_{n,m,i}(r)}{Y_{n,i}}\right) \tag{4.16}$$

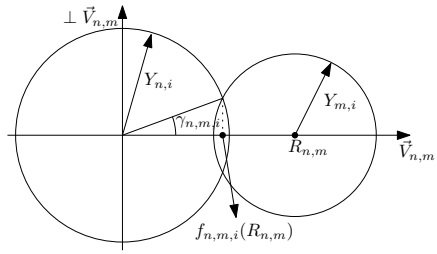


Figure 4.2: Two intersecting measurements,$Y_{n,i}$ and $Y_{m,i}$.

Let $\dot{f}_{n,m,i}(r)$ and $\ddot{f}_{n,m,i}(r)$ be the first and second derivatives of $f_{n,m,i}(r)$, respectively. Then,

$$\dot{f}_{n,m,i}(r) = \frac{1}{2} - \frac{Y_{n,i}^2 - Y_{m,i}^2}{2r^2}$$

for sufficiently small r: $f'_{n,m,i}(r) < 0$

$$\ddot{f}_{n,m,i}(r) = \frac{Y_{n,i}^2 - Y_{m,i}^2}{r^3}$$

$$\forall r > 0 : \ddot{f}_{n,m,i}(r) > 0$$

Given the range of $\mathcal{R}_{n,m}$ calculated from equation 4.14, we are guaranteed that for $r \in \mathcal{R}_{n,m}$ the measurements $Y_{n,i}$ and $Y_{m,i}$ intersect. Therefore, we can find $\mathcal{G}_{n,m,i}$, the set containing all possible $\gamma_{n,m,i}$, by using equations 4.15 and 4.16. Let $\mathcal{G}^+_{n,m,i} = \{\gamma | \gamma \geq 0, \gamma \in \mathcal{G}_{n,m,i}\}$, and $\mathcal{G}^-_{n,m,i} = \{\gamma | \gamma \leq 0, \gamma \in \mathcal{G}_{n,m,i}\}$ then,

$$\begin{aligned} \mathcal{G}^+_{n,m,i} =& cos^{-1}\left(\frac{f_{n,m,i}(\mathcal{R}_{n,m})}{Y_{n,i}}\right) \\ \mathcal{G}^-_{n,m,i} =& -cos^{-1}\left(\frac{f_{n,m,i}(\mathcal{R}_{n,m})}{Y_{n,i}}\right) \\ \mathcal{G}_{n,m,i} =& \mathcal{G}^+_{n,m,i} \cup \mathcal{G}^-_{n,m,i} \end{aligned} \tag{4.17}$$

Given the range of $R_{n,m}$, in order to determine the range of $\gamma_{n,m,i}$ we must find the minimum and maximum values of equation 4.15 with the appropriate constraint:

$$r \in \mathcal{R}_{n,m} \tag{4.18}$$

Case 1: $Y_{n,i} > Y_{m,i}$

In this case, we are guaranteed to have one minimum of equation 4.15 because it is a convex function for all $r > 0$. Given that equation 4.15 is convex, to find the minimum, we first calculate the derivative:

$$\dot{f}_{n,m,i}(r) = \frac{1}{2} - \frac{Y_{n,i}^2 - Y_{m,i}^2}{2r^2} \tag{4.19}$$

Then set it equal to 0 and solve for r:

$$\begin{aligned} \dot{f}_{n,m,i}(r_{n,m}) &= 0 \\ r_{n,m} &= \sqrt{Y_{n,i}^2 - Y_{m,i}^2} \end{aligned} \tag{4.20}$$

Applying the constraints on r described in 4.18 and keeping in mind that equation 4.15 is convex:

$$r_{min} = \begin{cases} r_{n,m} & \text{if} \quad r_{n,m} \in \mathcal{R}_{n,m} \\ min\left(\mathcal{R}_{n,m}\right) & \text{if} \quad r_{n,m} < min\left(\mathcal{R}_{n,m}\right) \\ max\left(\mathcal{R}_{n,m}\right) & \text{if} \quad max\left(\mathcal{R}_{n,m}\right) < r_{n,m} \end{cases} \tag{4.21}$$

Therefore:

$$min(f_{n,m,i}(r \in \mathcal{R}_{n,m})) = f_{n,m,i}(r_{min}) \tag{4.22}$$

And $max(f_{n,m,i}(r))$ then must occur at either $r_{max} = min\left(\mathcal{R}_{n,m}\right)$ or $r_{max} = max\left(\mathcal{R}_{n,m}\right)$:

$$\begin{aligned} &max(f_{n,m,i}(r \in \mathcal{R}_{n,m})) = \\ &\quad max(f_{n,m,i}(min\left(\mathcal{R}_{n,m}\right)), f_{n,m,i}(max\left(\mathcal{R}_{n,m}\right)) \end{aligned} \tag{4.23}$$

Case 2: $Y_{n,i} \leq Y_{m,i}$

In this case, the derivative of equation 4.15 (equation 4.19) is positive for all $r > 0$. This means that for $r \in \mathcal{R}_{n,m}$ equation 4.15 is monotonically increasing and that the minimum and maximum must be at the constraint edges:

$$min(f_{n,m,i}(r \in \mathcal{R}_{n,m})) = f_{n,m,i}(min\left(\mathcal{R}_{n,m}\right)) \tag{4.24}$$

And,

$$max(f_{n,m,i}(r \in \mathcal{R}_{n,m})) = f_{n,m,i}(max\left(\mathcal{R}_{n,m}\right)) \tag{4.25}$$

Now we can explicitly give the range of $\gamma_{n,m,i}$ in the set $\mathcal{G}_{n,m,i}$. Let $\mathcal{G}^{+}_{n,m,i} = \{\gamma | \gamma \geq 0, \gamma \in \mathcal{G}_{n,m,i}\}$, and $\mathcal{G}^{-}_{n,m,i} = \{\gamma | \gamma \leq 0, \gamma \in \mathcal{G}_{n,m,i}\}$ then,

$$\begin{aligned} &\mathcal{G}^{+}_{n,m,i} = \\ &\quad \left[cos^{-1}\left(\frac{max(f_{n,m,i}(r))}{Y_{n,i}}\right),\right. \\ &\quad \left. cos^{-1}\left(\frac{min(f_{n,m,i}(r))}{Y_{n,i}}\right)\right] \\ &\quad \text{and} \\ &\mathcal{G}^{-}_{n,m,i} = \\ &\quad \left[-cos^{-1}\left(\frac{min(f_{n,m,i}(r))}{Y_{n,i}}\right),\right. \\ &\quad \left. -cos^{-1}\left(\frac{max(f_{n,m,i}(r))}{Y_{n,i}}\right)\right] \\ &\quad \text{giving} \\ &\mathcal{G}_{n,m,i} = \mathcal{G}^{+}_{n,m,i} \cup \mathcal{G}^{-}_{n,m,i} \end{aligned} \tag{4.26}$$

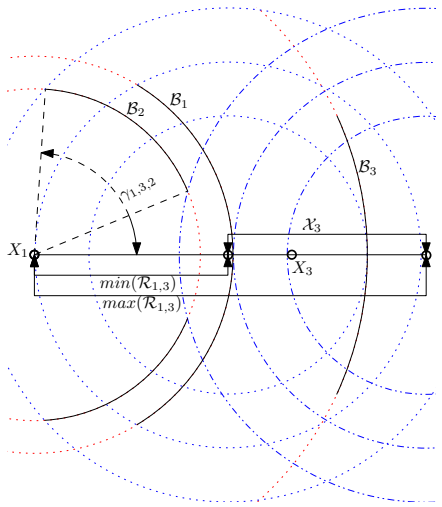


Figure 4.3: The possible configuration of the base nodes B_1, B_2, and B_3 (solid sections of circles) relative to X_1 using the measurement pair (Y_1, Y_3).

Calculating the α angles

Let $\alpha_{n,m,l}$ be the angle between $\vec{V}_{n,m}$ and $\vec{V}_{n,l}$. We can find a range for α by using the R values $R_{n,m}$, $R_{n,l}$, and $R_{m,l}$. Provided that the selected R values satisfy the triangle inequality:

$$\begin{aligned} R_{n,m} &\leq R_{n,l} + R_{m,l} \\ R_{n,l} &\leq R_{n,m} + R_{m,l} \\ R_{m,l} &\leq R_{n,m} + R_{n,l} \end{aligned} \tag{4.27}$$

Then we can use the law of cosines to calculate the angle $\alpha_{n,m,l}$:

$$\begin{aligned} F_c\left(R_{n,m}, R_{n,l}, R_{m,l}\right) = \frac{R_{n,m}^2 + R_{n,l}^2 - R_{m,l}^2}{2R_{n,m}R_{n,l}} = \\ = cos(\alpha_{n,m,l}) \end{aligned} \tag{4.28}$$

Since we do not have exact values for R we can use the set $\mathcal{R}$. We can check all the partial derivatives of equation 4.28 to get an idea about the minimum and maximum values:

$$\begin{aligned} \frac{\partial F_c\left(R_{n,m}, R_{n,l}, R_{m,l}\right)}{\partial R_{n,m}} &= \frac{R_{n,m}^2 - R_{n,l}^2 + R_{m,l}^2}{2R_{n,m}^2 R_{n,l}} \\ \frac{\partial F_c\left(R_{n,m}, R_{n,l}, R_{m,l}\right)}{\partial R_{n,l}} &= \frac{-R_{n,m}^2 + R_{n,l}^2 + R_{m,l}^2}{2R_{n,m}R_{n,l}^2} \\ \frac{\partial F_c\left(R_{n,m}, R_{n,l}, R_{m,l}\right)}{\partial R_{m,l}} &= -\frac{R_{m,l}}{R_{n,m}R_{n,l}} < 0 \end{aligned} \tag{4.29}$$

From 4.29 we notice that the maximum of the cosine (minimum of the angle) must occur at $min(\mathcal{R}_{m,l})$ and the minimum of the cosine (maximum of the angle) at $max(\mathcal{R}_{m,l})$ since the particular partial derivative is strictly negative. It is not as simple for $\mathcal{R}_{n,m}$ and $\mathcal{R}_{n,l}$, but we can make some observations about the cosine from equation 4.28 and applying the triangle inequality:

$$\begin{array}{lll} 1. & R_{n,m} - R_{n,l} \geq R_{m,l} & F_c\left(R_{n,m}, R_{n,l}, R_{m,l}\right) = 1 \\ 2. & R_{n,l} - R_{n,m} \geq R_{m,l} & F_c\left(R_{n,m}, R_{n,l}, R_{m,l}\right) = 1 \\ 3. & R_{n,m} + R_{n,l} \leq R_{m,l} & F_c\left(R_{n,m}, R_{n,l}, R_{m,l}\right) = -1 \\ 4. & R_{n,m}^2 + R_{n,l}^2 = R_{m,l}^2 & F_c\left(R_{n,m}, R_{n,l}, R_{m,l}\right) = 0 \end{array} \tag{4.30}$$

And some additional observations about the gradients in equation 4.29:

$$\begin{array}{lll} 1. & R_{n,m}^2 - R_{n,l}^2 > R_{m,l}^2 & \frac{\partial F_c(R_{n,m}, R_{n,l}, R_{m,l})}{\partial R_{n,m}} > 0 \\ 2. & R_{n,l}^2 - R_{n,m}^2 > R_{m,l}^2 & \frac{\partial F_c(R_{n,m}, R_{n,l}, R_{m,l})}{\partial R_{n,l}} > 0 \\ 3. & R_{n,l}^2 + R_{n,m}^2 < R_{m,l}^2 & \begin{array}{l} \frac{\partial F_c(R_{n,m}, R_{n,l}, R_{m,l})}{\partial R_{n,m}} > 0 \\ \frac{\partial F_c(R_{n,m}, R_{n,l}, R_{m,l})}{\partial R_{n,m}} > 0 \end{array} \\ 4. & R_{n,l} = R_{n,m} & \begin{array}{l} \frac{\partial F_c(R_{n,m}, R_{n,l}, R_{m,l})}{\partial R_{n,m}} = \\ = \frac{\partial F_c(R_{n,m}, R_{n,l}, R_{m,l})}{\partial R_{n,m}} > 0 \end{array} \end{array} \tag{4.31}$$

For R values in $\mathcal{R}$ as defined in equation4.14, lets define four significant pairs:

$$\begin{aligned} P_1 &= \left(min\left(\mathcal{R}_{n,m}\right), min\left(\mathcal{R}_{n,l}\right)\right) \\ P_2 &= \left(min\left(\mathcal{R}_{n,m}\right), max\left(\mathcal{R}_{n,l}\right)\right) \\ P_3 &= \left(max\left(\mathcal{R}_{n,m}\right), min\left(\mathcal{R}_{n,l}\right)\right) \\ P_4 &= \left(max\left(\mathcal{R}_{n,m}\right), max\left(\mathcal{R}_{n,l}\right)\right) \end{aligned} \tag{4.32}$$

Now, from the observations in equations 4.30 and 4.31 and letting $R_{m,l} = min(\mathcal{R}_{m,l})$ we can calculate the minimum of the angle $min_R(\alpha_{n,m,l})$ using equation 4.28: Letting $R_{m,l} = max(\mathcal{R}_{m,l})$ we can calculate the maximum of the angle using equation 4.28: Lets denote the set of possible $\alpha_{n,m,l}$ derived from $\mathcal{R}$ values as $\mathcal{A}_{n,m,l}^R$:

$$\mathcal{A}_{n,m,l}^R = [min_R(\alpha_{n,m,l}), max_R(\alpha_{n,m,l})] \tag{4.33}$$

We can further limit the range of α by using the previously calculated γ angle ranges. Lets denote the set of $\alpha_{n,m,l}$ derived from γ values as $\mathcal{A}_{n,m,l}^\gamma$, then:

$$\mathcal{A}_{n,m,l}^\gamma = \{\alpha | \alpha \in [0, \pi], \mathcal{G}_{n,m,i} \cap (\mathcal{G}_{n,l,i} + \alpha) \neq \emptyset \ \forall i\} \tag{4.34}$$

Putting $\mathcal{A}_{n,m,l}^R$ and $\mathcal{A}_{n,m,l}^\gamma$ together to get $\mathcal{A}_{n,m,l}$:

$$\mathcal{A}_{n,m,l} = \mathcal{A}_{n,m,l}^R \cap \mathcal{A}_{n,m,l}^\gamma \tag{4.35}$$

Note that $\mathcal{A}_{n,m,l} = \mathcal{A}_{n,l,m}$

Algorithm 1: Algorithm to calculate the minimum of the α angle according to $\mathcal{R}$.

Data: $\mathcal{R}$
Result: $min\left(\alpha\right)$
if P_2 *and* P_3 *satisfy 4.30.1 or 4.30.2* **then**
    $min_R(\alpha_{n,m,l}) = 0$
else
    $min_R(\alpha_{n,m,l}) =$
        $min\left(arccos\left(F_c\left(P_1, R_{m,l}\right)\right),\right.$
            $arccos\left(F_c\left(P_2, R_{m,l}\right)\right),$
            $arccos\left(F_c\left(P_3, R_{m,l}\right)\right),$
            $\left.arccos\left(F_c\left(P_4, R_{m,l}\right)\right)\right)$
end

Algorithm 2: Algorithm to calculate the maximum of the α angle according to $\mathcal{R}$.

Data: Y measurements
Result: R, G, A, B, X
if P_1 *satisfies 4.30.3* **then**
    $max_R(\alpha_{n,m,l}) = \pi$
else if $min\left(\mathcal{R}_{n,m}\right)^2 \leq R_{m,l}$ *or* $min\left(\mathcal{R}_{n,l}\right)^2 \leq R_{m,l}$ **then**
    $max_R(\alpha_{n,m,l}) = arccos(F_c(P_1, R_{m,l}))$
else if $min\left(\mathcal{R}_{n,m}\right) > min\left(\mathcal{R}_{n,l}\right)$ **then**
    $r_{n,l} = min\left(\sqrt{min\left(\mathcal{R}_{n,m}\right)^2 - R_{m,l}^2}, max\left(\mathcal{R}_{n,l}\right)\right)$
        and
        $max\left(\sqrt{min\left(\mathcal{R}_{n,m}\right)^2 - R_{m,l}^2}, min\left(\mathcal{R}_{n,l}\right)\right)$
    $max_R(\alpha_{n,m,l}) = arccos\left(F_c\left(min\left(\mathcal{R}_{n,m}\right), r_{n,l}, R_{m,l}\right)\right)$
else if $min\left(\mathcal{R}_{n,m}\right) < min\left(\mathcal{R}_{n,l}\right)$ **then**
    $r_{n,m} = min\left(\sqrt{min\left(\mathcal{R}_{n,l}\right)^2 - R_{m,l}^2}, max\left(\mathcal{R}_{n,m}\right)\right)$
        and
        $max\left(\sqrt{min\left(\mathcal{R}_{n,l}\right)^2 - R_{m,l}^2}, min\left(\mathcal{R}_{n,m}\right)\right)$
    $max_R(\alpha_{n,m,l}) = arccos\left(F_c\left(r_{n,m}, min\left(\mathcal{R}_{n,l}\right), R_{m,l}\right)\right)$
$max_R(\alpha_{n,m,l}) = arccos\left(F_c\left(P_1, R_{m,l}\right)\right)$

Determining the direction of $\mathcal{A}_{n,m,p}$

$\mathcal{A}_{n,m,p}$ describes the magnitude of the angle between $\vec{V}_{n,m}$ and $\vec{V}_{n,p}$ and $\mathcal{A}_{n,m,p} \subseteq [0, \pi]$. In order to calculate the base node configuration we need to also determine the direction $\mathcal{C}^p_{n,m,l}$ of $\mathcal{A}_{n,m,p}$ (i.e., if the sign should be positive or negative) given that the direction of $\mathcal{A}_{n,m,l}$ is positive.
Let

$$\begin{aligned}
\tilde{\mathcal{R}}^{p+}_{n,m,l} &= \left\| r_l \begin{bmatrix} cos(\theta_l) \\ sin(\theta_l) \end{bmatrix} - r_p \begin{bmatrix} cos(\theta_p) \\ sin(\theta_p) \end{bmatrix} \right\| = \\
&= \left\| r_l - r_p \begin{bmatrix} cos(\theta_p - \theta_l) \\ sin(\theta_p - \theta_l) \end{bmatrix} \right\| = \\
&= \sqrt{r_l^2 - 2 r_l r_p cos(\theta_p - \theta_l) + r_p^2} \\
\tilde{\mathcal{R}}^{p-}_{n,m,l} &= \left\| r_l \begin{bmatrix} cos(\theta_l) \\ sin(\theta_l) \end{bmatrix} - r_p \begin{bmatrix} cos(-\theta_p) \\ sin(-\theta_p) \end{bmatrix} \right\| = \\
&= \left\| r_l - r_p \begin{bmatrix} cos(-\theta_p - \theta_l) \\ sin(-\theta_p - \theta_l) \end{bmatrix} \right\| = \\
&= \sqrt{r_l^2 - 2 r_l r_p cos(\theta_p + \theta_l) + r_p^2}
\end{aligned} \tag{4.36}$$

For all $r_l \in \mathcal{R}_{n,l}$, $r_p \in \mathcal{R}_{n,p}$, $\theta_l \in \mathcal{A}_{n,m,l}$, and $\theta_p \in \mathcal{A}_{n,m,p}$

Then,

$$\mathcal{C}^p_{n,m,l} = \begin{cases} \{1\} & \text{if} \quad \mathcal{R}_{l,p} \cap \tilde{\mathcal{R}}^{p+}_{n,m,l} \neq \emptyset \;\; \mathcal{R}_{l,p} \cap \tilde{\mathcal{R}}^{p-}_{n,m,l} = \emptyset \\ \{-1\} & \text{if} \quad \mathcal{R}_{l,p} \cap \tilde{\mathcal{R}}^{p+}_{n,m,l} = \emptyset \;\; \mathcal{R}_{l,p} \cap \tilde{\mathcal{R}}^{p-}_{n,m,l} \neq \emptyset \\ \{-1,1\} & \text{if} \quad \mathcal{R}_{l,p} \cap \tilde{\mathcal{R}}^{p+}_{n,m,l} \neq \emptyset \;\; \mathcal{R}_{l,p} \cap \tilde{\mathcal{R}}^{p-}_{n,m,l} \neq \emptyset \\ \emptyset & \text{if} \quad \mathcal{R}_{l,p} \cap \tilde{\mathcal{R}}^{p+}_{n,m,l} = \emptyset \;\; \mathcal{R}_{l,p} \cap \tilde{\mathcal{R}}^{p-}_{n,m,l} = \emptyset \end{cases} \tag{4.37}$$

In the case when $\mathcal{C}^p_{n,m,l} = \{-1, 1\}$ or $\mathcal{C}^p_{n,m,l} = \emptyset$, one can interchange l with any $\hat{p} \in \mathcal{P}$ where $\mathcal{P} = \left\{ \hat{p} \middle| \mathcal{C}^{\hat{p}}_{n,m,l} = \{1\}, \mathcal{C}^{\hat{p}}_{n,m,l} = \{-1\} \right\}$ and recalculate $\mathcal{C}^p_{n,m,l}$ as $\mathcal{C}^p_{n,m,l} = \mathcal{C}^{\hat{p}}_{n,m,l} \mathcal{C}^p_{n,m,\hat{p}}$. One can perform this step until all $\hat{p}$ have been exhausted or until $\mathcal{C}^p_{n,m,\hat{p}} = \{1\}$ or $\mathcal{C}^p_{n,m,\hat{p}} = \{-1\}$. Consequently $\mathcal{C}^p_{n,m,l} = \mathcal{C}^{\hat{p}}_{n,m,l} \mathcal{C}^p_{n,m,\hat{p}}$.

Determining the location of base nodes

We can use $\mathcal{A}_{n,m,l}$, $\mathcal{G}_{n,m,i}$ and the measurement configuration $\mathcal{C}^p_{n,m,l}$ in order to get the base node configuration $\mathcal{D}$:

$$\begin{aligned} &\text{For any } l \neq n, m \text{ and } \mathcal{C}^p_{n,m,l} \neq \emptyset \\ \mathcal{D}^+_{n,m,i} &= \mathcal{G}_{n,m,i} \cap \left(\mathcal{G}_{n,1,i} + \mathcal{C}^1_{n,m,l}\mathcal{A}_{n,m,1}\right) \cap \cdots \\ &\quad \cdots \cap \left(\mathcal{G}_{n,N,i} + \mathcal{C}^N_{n,m,l}\mathcal{A}_{n,m,N}\right) \\ &\text{or the mirrored} \\ \mathcal{D}^-_{n,m,i} &= \mathcal{G}_{n,m,i} \cap \left(\mathcal{G}_{n,1,i} - \mathcal{C}^1_{n,m,l}\mathcal{A}_{n,m,1}\right) \cap \cdots \\ &\quad \cdots \cap \left(\mathcal{G}_{n,N,i} - \mathcal{C}^N_{n,m,l}\mathcal{A}_{n,m,N}\right) \end{aligned} \tag{4.38}$$

$$\mathcal{B}^{\pm}_{n,m,i} = Y_{n,i} \begin{bmatrix} cos(\mathcal{D}^{\pm}_{n,m,i}) \\ sin(\mathcal{D}^{\pm}_{n,m,i}) \end{bmatrix} \tag{4.39}$$

Determining the location of measurement nodes

The location of the measurement node X_p with respect to nodes X_n, X_m (node at angle 0), and X_l (non-colinear node at positive $\mathcal{A}$) $\mathcal{X}^p_{n,m,l}$ can be calculated as follows:

$$\mathcal{X}^{\pm p}_{n,m,l} = \mathcal{R}_{n,p} \begin{bmatrix} cos(\pm\mathcal{C}^p_{n,m,l}\mathcal{A}_{n,m,p}) \\ sin(\pm\mathcal{C}^p_{n,m,l}\mathcal{A}_{n,m,p}) \end{bmatrix} \tag{4.40}$$

4.3.3 Coordinate system stitching

Suppose there are two measurement nodes X_a and X_b that each have access to two different measurement sets which have three common measurement instances Y_c, Y_d, and Y_e. With those measurements they can each estimate $\mathcal{X}(a)$, $\mathcal{B}(a)$ and $\mathcal{X}(b)$, $\mathcal{B}(b)$ respectively. Since they have the common measurements of Y_c, Y_d, and Y_e one can stitch the coordinate system of X_b to the coordinate system of X_a by rotating and translating $\mathcal{X}(b)^p_{c,d,e}$ such that there is maximum overlap of the transformed $\mathcal{X}(b)^c_{c,d,e}$, $\mathcal{X}(b)^d_{c,d,e}$, $\mathcal{X}(b)^e_{c,d,e}$ and any other common nodes with $\mathcal{X}(a)^c_{n,m,l}$, $\mathcal{X}(a)^d_{n,m,l}$, $\mathcal{X}(a)^e_{n,m,l}$ and the others. The rotation and translation values can then be used on $\mathcal{B}(b)$ to stitch in the locations of any new beacons as well. Lets denote the stitched coordinate systems as $\hat{\mathcal{X}}$ and $\hat{\mathcal{B}}$.

4.3.4 The ReNLoc algorithm

In this section we will describe two variants of the ReNLoc algorithm, one centralized and one distributed.

Centralized ReNLoc

The centralized ReNLoc uses all measurements Y together to localize all nodes with respect to each other:

Algorithm 3: The centralized ReNLoc algorithm

Data: Y measurements
Result: R, G, A, B, X
1 Calculate $\mathcal{R}$ from the Y measurements
2 Calculate $\mathcal{G}$ from Y and $\mathcal{R}$
3 Calculate $\mathcal{A}$ from $\mathcal{G}$ and $\mathcal{R}$
4 Calculate $\mathcal{B}$ and $\mathcal{X}$

Distributed ReNLoc

The distributed algorithm relies on communicating measurements and current coordinate system models between neighbors and thus propagates coordinate information throughout the entire network. Each node performs the following algorithm to localize all nodes with respect to itself:

Algorithm 4: The distributed ReNLoc algorithm

Data: Y measurements
Result: R, G, A, B, X
1 Get Y measurements, $\hat{\mathcal{X}}_{c,d,e}$, and $\hat{\mathcal{B}}_{c,d}$ from neighbors. Where c, d, and e are three common measurements of the current node with each neighbor
2 Calculate $\mathcal{R}$ from the Y measurements. Calculate $\mathcal{G}$ from Y and $\mathcal{R}$
3 Calculate $\mathcal{A}$ from $\mathcal{G}$ and $\mathcal{R}$
4 Calculate $\mathcal{B}$ and $\mathcal{X}$
5 **forall** *neighbors* **do**
6 Perform coordinate stitching with $\hat{\mathcal{X}}$
7 use metrics of coordinate stitching to stitch $\hat{\mathcal{B}}$
8 **end**

4.4 Results Discussion

In order to assess the accuracy of our algorithm we performed simulations for N measuring nodes and K base nodes ranging from 3 to 30 in increments of 3. For each case, we measured the mean square error (MSE) in $R_{n,m}$ and the mean square error (MSE) in $A_{n,m,l}$ for 100 trials of randomly placed base nodes and measuring nodes in a $100m \times 100m$ area. We compare with multidimensional scaling (MDS) using the same parameters. Recall from chapter 2 that MDS is a technique from mathematical psychology, it can be used to calculate relative maps of nodes based on distances between the nodes. From a graph of a network one can estimate a distance matrix by calculating the shortest path between all pairs of nodes and then apply classical MDS on the distance matrix. An important aspect of ReNLoc to keep in mind is that it gives possible ranges of values for the estimates. In figures 4.5 and 4.9 we used the mean of the possible values ($mean(\mathcal{R}_{n,m})$ and $mean(\mathcal{A}_{n,m,l})$).

For ReNLoc, the estimates of R were calculated as described in section 4.3.2. For MDS the R estimate was calculated as the euclidean distance between the points generated from the MDS algorithm using the shortest path distance. An overview of the MSE of R with different values of N and K can be seen in figures 4.4 and 4.5, note the scale of the color graph on the right hand side.

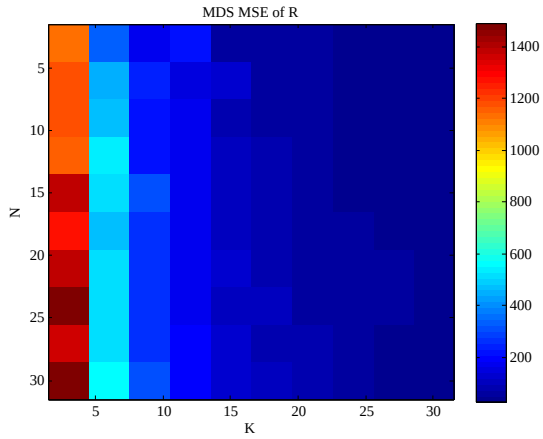


Figure 4.4: The MSE of R for MDS.

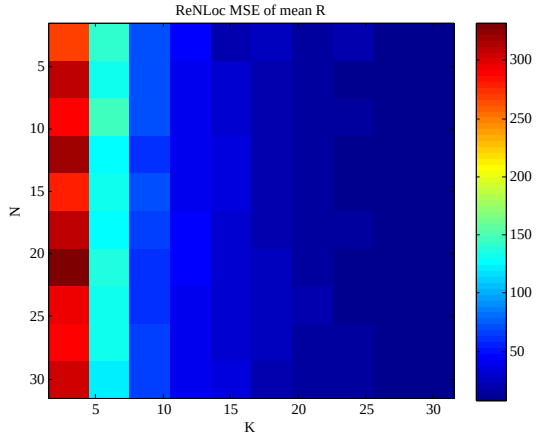


Figure 4.5: The MSE of R using the mean of each R estimate from ReNLoc.

The error of the range estimates R decreases as we increase the number of base nodes K as can be seen in figure 4.6. The reason behind this is that as we increase the number of base nodes we increase the number of constraints when calculating the R estimates in equation 4.14. Using the mean values of the R estimates from ReNLoc results in a significantly smaller MSE than MDS. In figure 4.7 the number of measuring nodes seems to have very little effect on the

accuracy of ReNLoc as opposed to the accuracy of MDS which decreases with larger N. We also notice that for $K = 30$ even the worst ReNLoc estimates are comparable to MDS while the mean ReNLoc estimates remain well below 15.

Since ReNLoc was built with this particular network architecture in mind (measuring nodes and base nodes), we can expect it to have an advantage. In fact, MDS uses the shortest path distance R for its computations. In this setup however, there are always 2 hops between measuring nodes when it comes to range measurements which has a significant effect on the accuracy of such estimates. This fact accounts for the lower accuracy of MDS when it comes to the R estimates.

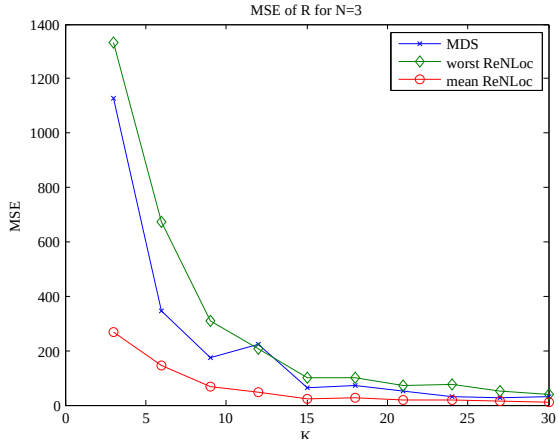


Figure 4.6: Comparison of the MSE of R at $N = 3$.

In figure 4.6, it is interesting to note that the values of the MSE for $K = 30$ are 30.04, 40.89, and 9.94 for MDS, worst ReNLoc, and mean ReNLoc respectively. This is a significant improvement over MDS especially when we see that the MSE of mean ReNLoc is well below 30 (at 22.75) starting at $K = 15$ where MDS is in fact at an MSE value of 65.31 at the same point.

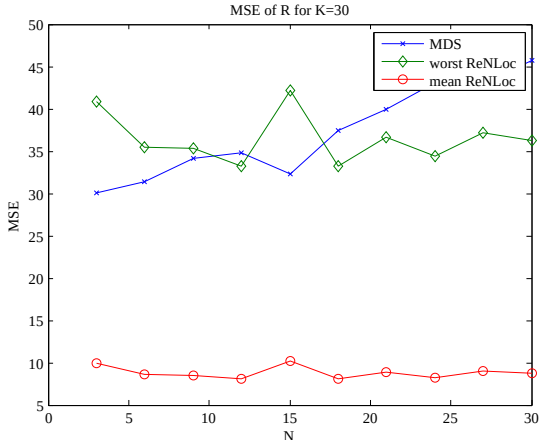


Figure 4.7: Comparison of the MSE of R at $K = 30$.

For ReNLoc, the estimates of A where calculated as described in section 4.3.2. The estimates of A for MDS where calculated using the law of cosines equation (equation 4.28) with the R estimates. An overview of the MSE of A with different values of N and K can be seen in figures 4.8 and 4.9, note the scale of the color graph on the right hand side and also that the angles were calculated in radians ($0\,rad = 0°$, $\pi\,rad = 180°$).

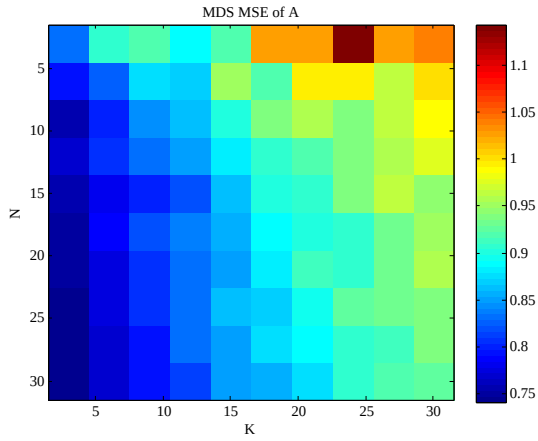


Figure 4.8: The MSE of A for MDS.

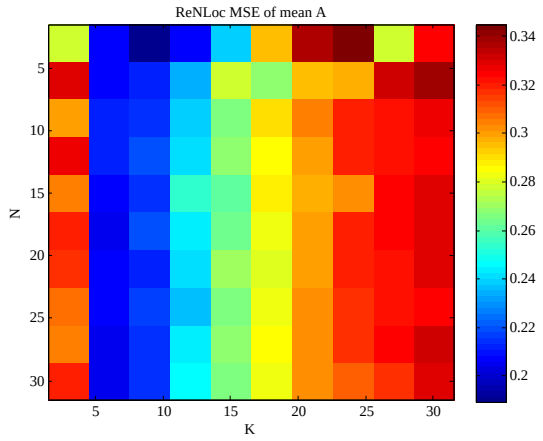


Figure 4.9: The MSE of A using the mean of each A estimate from ReNLoc.

For the angle estimates A, ReNLoc once again has much smaller MSE than the MDS algorithm. In figure 4.10 we notice that ReNLoc has an optimal number of base nodes with respect to the angle accuracy at around $K = 6$, but with MDS the error increases as K increases. We have not fully investigated the cause of this for ReNLoc, however we suspect that it is due to error accumulation from the rapidly increasing number of angles that need to be estimated with the addition of each base node K. The effects of the number of measuring nodes N has very little effect in ReNLoc, but in MDS we notice a gradual improvement in the MSE as N increases all the while staying well above the relatively low MSE of ReNLoc for the range of N that was simulated (figures 4.11 and 4.12).

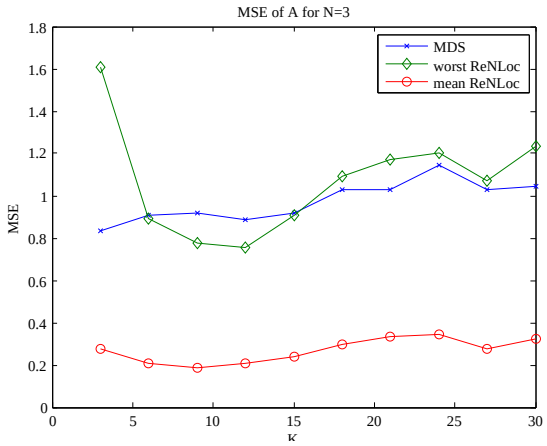


Figure 4.10: Comparison of the MSE of A at $N = 3$.

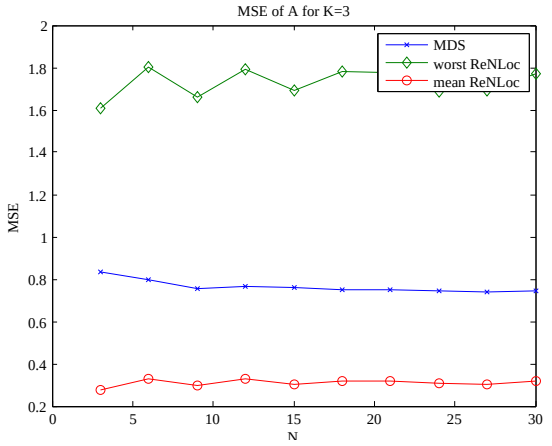


Figure 4.11: Comparison of the MSE of A at $K = 3$.

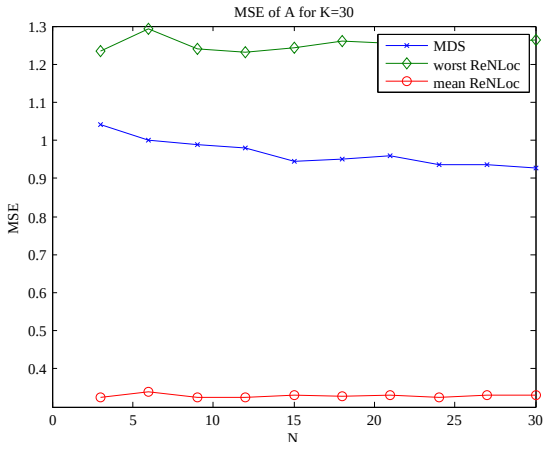


Figure 4.12: Comparison of the MSE of A at $K = 30$.

In general we notice that using the mean of the ReNLoc estimates is much more accurate than MDS in all cases. Furthermore, the number of measuring nodes N has little effect on the accuracy for ReNLoc but we notice that in MDS the MSE of A has a slight improvement. The number of base nodes K has a clear positive effect on the MSE (figures 4.4,4.5,4.8,4.9).

In figure 4.13 we see a sample of how ReNLoc results can be interpreted. Since ReNLoc outputs possible ranges of values for R and A, we can define regions where the measuring nodes and the base nodes are guaranteed to be in. Using the mean values of each estimate of R and A will produce a point with relatively small MSE in each region.

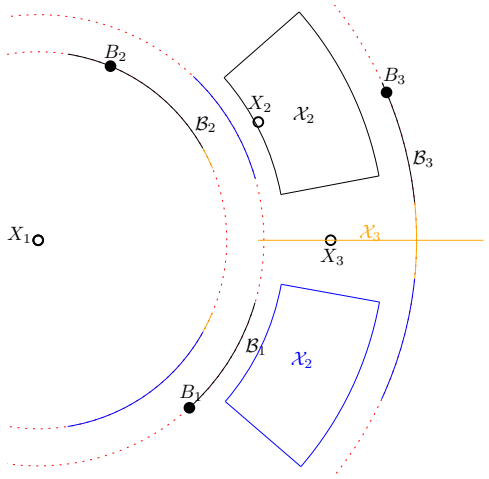


Figure 4.13: From the setup of figure 4.1 here we illustrate $\mathcal{X}_{1,3,2}$ and $\mathcal{B}_{1,3}$. Configuration is in black, blue is mirrored and orange is overlapped regions.

4.5 Discussion

In this chapter we have presented ReNLoc (**Re**laid ra**N**ging **Loc**alization), an algorithm that localizes mobile nodes that can only range to base nodes and not between themselves. We have shown that this algorithm can also be used in a distributed manner by using coordinate stitching techniques. Simulations of the centralized version of this algorithm performed significantly better compared to MDS, a commonly used algorithm for such situations.

We have addressed common issues such as power consumption, reliance on infrastructure, ease of deployment, and most importantly, privacy. Power consumption was addressed by removing the need of anchors or beacons (nodes with knowledge about their absolute position) because these nodes often rely either on sensors such as GPS, which require a significant amount of power, or on databases of the anchor node locations, making it a less adaptable solution and more prone to privacy leaks to third parties. Reliance on infrastructure is minimal since no specific setup is needed for ReNLoc making it very easy to deploy. For ubiquitous computing applications, cities may already have the necessary infrastructure in place by using GSM towers or other access points (i.e. WiFi, bluetooth) to cover the need of base nodes. It can easily be deployed in areas in which we have no knowledge about the topology of the base nodes where solutions such as PlaceLab [30] would take a non-negligible amount of effort to build the location database. In general, the infrastructure can be easily deployed by means of randomly or crudely dispersing base nodes in the desired area by means of a beacon airdrop without requiring any information about the network configuration or locations of base nodes.

Since our algorithm provides a range of possible configurations for the nodes and not exact coordinates like other techniques we are able to provide bounds on

the location of each of the nodes and understand in which direction and how far any errors might occur. Algorithms that provide exact coordinates might only give a radius of accuracy which approximate the error while ours gives exact dimensions and shape of the possible errors.

Our evaluation of ReNLoc relied on ideal conditions where exact distance measurements to base nodes were possible. Although it can give us a general understanding of the performance when compared against other algorithms, the results should be taken with a grain of salt. The significant improvement over MDS in the same ideal conditions leads us to believe that it could also show at least some improvement in a realistic simulation but until these test are performed we cannot be certain. Furthermore, our algorithm, in its current state, has many more computational steps compared to MDS which may affect power consumption on mobile devices. Nevertheless, the increasing computational power and efficiency of modern devices may give us some more room for complexity than previously possible.

4.5.1 ReNLoc application areas

Relative localization of nodes is very beneficial for many services in a range of application areas and the particular solution proposed here could be leveraged for localization of nodes in Wireless Sensor Networks [179], localization of mobile devices/nodes for authorization purposes [180] or localization of mobile robots with respect to obstacles [181], as well as for augmented and mixed-reality services [182] or services based on indoor localization [113], human mobility and tracking, for traffic management [183] or meaningful place discovery [184].

Measurement nodes in ReNLoc may measure distances using several methods: radio signals ([185] via cellIDs or WiFi APs), optically (laser beam or an infrared reflection principle), thermally or acoustically (measuring sound [186]). Measurements can also be done by using images of a scene from different perspectives [187]. We have designed and implemented this algorithm with ubiquitous computing in mind. Although no real world tests have been performed yet to evaluate the performance of ReNLoc in the wild, we propose that it can be used to measure several location-related features such as distance traveled, speed, and path structure as well as to detect significant places of users with the mobile device acting as the measuring node and GSM cell towers acting as the base node. As we have stated earlier, ReNLoc is not accurate enough to determine exact location of a user, but since it can be used with only the GSM antenna, without requiring to activate the on-board GPS sensor or use the accelerometer of the device, the power usage of the device is conserved to a minimum and can be always active as opposed to activating only at specified time intervals. Such coarse localization from ReNLoc can therefore estimate the general direction of a user's movement to some course significant places which can be used to track a user's movements between those places. Location based services relying on significant places (home, work, leisure, etc.) can therefore always be on or active without making such a big sacrifice in the battery life of the mobile device. Services like fitness tracking which require information such as distance and speed could also be realized using our algorithm. The actual application areas depend on the

performance of our algorithm in a realistic environment which includes inaccurate ranging and other noise.

In order to somewhat reduce the computational complexity of our algorithm it might be beneficial to analyze the procedure and find efficient computational analogies for some of the steps. Although approximations could be used to reduce complexity, we would like to avoid that as it would nullify some of the benefits of our current setup. We can then run tests to validate our assumption that ReNLoc has the potential to be used in the wild to track location-related information such as distance traveled, speed, path structure, and to detect significant places.

Chapter 5

Location Context Service Self-Provisioning

5.1 Introduction

Location context is a commonly used feature for location based services (LBS) and context aware services alike. For these services, location is often provided by either the use of GPS, for an accurate outdoor location, or a location service (for example, Google location services) that provides an approximate location with the help of access point (AP) map database such as WiFi and cell tower locations, or localization algorithms that rely on other sensors available on the phone like accelerometer [188]. Although the aforementioned approaches may provide a satisfying user experience in terms of location estimation, they are not always the best solution when it comes to energy consumption or privacy. From a privacy perspective, applications which only require location context should not be receiving exact GPS coordinates of the location but rather receive location context information from some service able to do so. Furthermore, location service interruptions, although rare, can propagate to the LBS and context aware services. For these reasons, we explore self-provisioning of location context information in order to address the issue of privacy and to complement or even replace, when possible, existing location services in the domain of context. Of course, location context can be readily derived using GPS traces but this would have a noticeable effect on the power consumption of the device. Therefore it would be desirable to provide location context without the use of the GPS.

Even though the majority of mobile phone devices are equipped with a multitude of sensors, the mobile connectivity antenna sensor is guaranteed to be installed. The accelerometer and compass are also fairly common sensors, however, their utility as localization tools suffer from drift errors (when dead reckoning is used) and power consumption due to the sampling rate and calculations necessary to provide useful results. The *Cell ID* is a unique identifier for the cell tower that the mobile device is currently connected to — it is 'free' information since there is no cost for the OS to provide it. Additionally, the processing of Cell ID data

streams is far less demanding than other high frequency sensor streams. Cell ID maximum handover rate is less than $10Hz$ in our experimental data compared to accelerometer sampling rate which can range from $10Hz$ to $1600Hz$. One main issue concerning the use of Cell IDs is the difference in cell tower deployment density between urban and rural areas. For example a typical urban area has a cell tower deployment with an average per-tower coverage radius of less than 500 meters whereas a rural area each tower can have a coverage radius in the kilometer range which is too large to derive meaningful information [189, 190].

5.1.1 Contributions

In this chapter, we aim to show that it is feasible to use the Cell ID traces alone to detect meaningful places for a mobile user, without the need for GPS, other sensor data, or a location service. We do this by implementing two very different clustering techniques: one based on the graph connectivity of Cell IDs with our own definition of what it means for two Cell IDs to be connected, and one density based clustering where we also define the distance metrics that can be used. We assume an urban environment with a sufficiently dense deployment of cell towers in order to impose a clear range limit on the Cell IDs.

5.1.2 Structure

The structure of this chapter is as follows: We first describe the data which we are using to validate our algorithms in section 5.2 so that we can present relevant results incrementally as we describe different parts of our approach. We define and describe Cell ID similarity measures in section 5.3. Namely, we describe how we can detect a Cell ID oscillation, we define a Cell ID adjacency matrix, and we describe how to estimate pairwise distances and how to make sure that those estimates fall into a metric or euclidean space. In section 5.4.3 we describe various techniques to derive Cell ID clusters that describe meaningful places. Specifically, in section 5.4.1 we formalize and improve upon a previously developed clustering technique that utilized Cell ID oscillation events to cluster Cell IDs. We also describe another clustering technique that utilizes Cell ID oscillation events in section 5.4.3. Finally we conclude about this work in section 5.5 and present an application of one of the clustering techniques.

5.2 Data summary

The data used in this chapter was collected from the Android OS devices of two different users for approximately two weeks each. Both of the devices were *Google Nexus 5X*. A data point was collected each time that the mobile device underwent a handover to a different Cell ID and each data tuple included the following information: Cell ID timestamp, Cell ID, location timestamp, location (from Google location service), and location accuracy (from Google location service). For user 1 there were a total of 825 handover events between 69 different Cell IDs. For user 2 there were a total of 513 handover events between 84 different Cell IDs. Both

users were in Geneva, Switzerland, for the 2 week duration of the data collection. We removed location information which had an accuracy larger than 200 meters (as estimated by the Google location service) as it could skew the velocity estimates later on. Besides that, we did not manipulate the raw data in any other way (no additional filtering or normalization).

For each Cell ID, we collect the holding times in a cumulative distribution graph and then compare the fit of an exponential and a gamma distribution (see an example in Fig. 5.1). For Cell IDs with more than 10 data entries, we find that the exponential distribution fit has a mean squared error (MSE) of 0.0851 and maximum squared error (SE) of 0.7494. The gamma distribution has an MSE of 0.0233 and a maximum SE of 0.2225. Based on our analysis we find that the gamma distribution is more representative of the CDF of the data. Given this information, we can conclude that the process under which the Cell IDs undergo a handover is a Markov process with a gamma distribution. This information can prove useful in the case when we need to estimate the holding time of a particular cluster of Cell IDs based only on the knowledge of the holding time of individual Cell IDs. Such a case can arise when we want to save computing resources to calculate cluster holding times after the clusters have experienced a change (i.e. estimating cluster holding times after any clique changes in the maximal clique method in section 5.4.1). The theory and mathematics of how to estimate the holding times of an aggregation of Markov states is described in a publication by Rubino and Sericola [191].

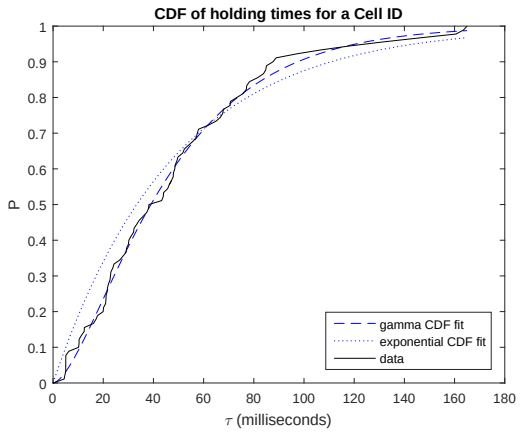


Figure 5.1: The holding time cumulative distribution of a Cell ID and the corresponding fitted exponential and gamma distribution.

5.3 Cell ID Similarity Measures

To cluster Cell IDs into groups that represent meaningful places we need to define some measures for similarities, connectivity, or distance between Cell IDs. Kang et al. [33] achieved this in their location measurements using both spatial distance and temporal distance as their similarity metrics. Fanourakis et al. [25] and Yadav et al. [37] achieved this by analyzing graph connectivity patterns between Cell IDs and deriving simple similarity metrics. In this section we will describe similarity measures that can be used when only Cell ID and timestamp observations are available.

5.3.1 Cell ID Oscillation

The *Cell ID oscillation* event used by Fanourakis et al. [25] and Yadav et al. [37] can be described as follows: assume there is a Cell ID sequence $\mathcal{S} = \{c_b, c_1, ..c_i, ..., c_j, c_b\}$, then a possible Cell ID oscillation event is detected when two observations of a base cell c_b enclose a set of Cell IDs c_i in the sequence S where $c_i \neq c_b$. Furthermore we will define the oscillation time $\delta_b \in \Delta = \{\delta_1, ..., \delta_n, ..., \delta_N\}$ of base cell c_b be the time between the two observations of c_b. When δ_b is below a predefined oscillation time threshold τ_b, then this event is a Cell ID oscillation. These events can occur even if the user is not moving out of range and back from a particular Cell, they can be caused by network load, small time signal fading, and inter-network (2G to 3G, 4G, etc. and vice versa) handoffs [192].

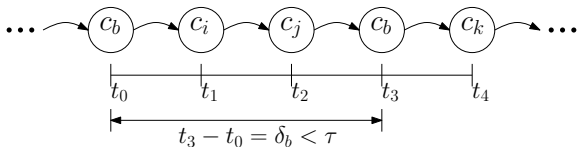


Figure 5.2: A sequence of Cell ID measurements where Cell ID c_b has oscillated.

The oscillation time threshold can be the same for all cells, $\tau_1 = \tau_2 = ..., \tau_N = \tau$, it can also depend on specific properties of each cell like the *holding time* of that cell or it can be defined using the properties of all the cells involved in a potential oscillation event. Let $\mathcal{T}^o = \{\tau_1, \tau_2, ..., \tau_N\}$. Furthermore, let $\mathcal{H}_i = \{h_{i,1}, h_{i,2}, ..., h_{i,K}\}$ be the collection of holding times of cell c_i where each holding time $h_{i,k}$ is defined as the time it takes after each observation of c_i to transition to an other cell. Then, we can define a function $f_o(\cdot)$ that can take as arguments either the set of holding times of a specific cell or the sets of holding times for any cell involved in a potential oscillation event and give us an oscillation threshold τ_i. We empirically determine an appropriate function upon analysis of our data.

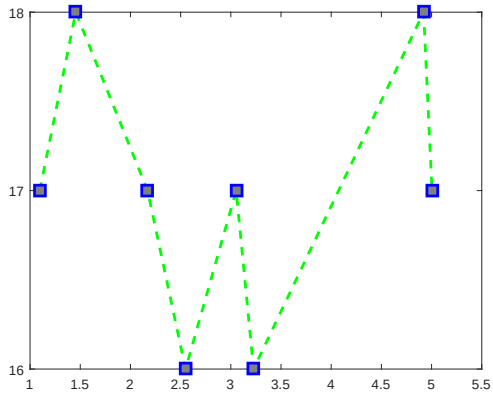


Figure 5.3: Cell ID oscillation between cells 16, 17, and 18 in the data of user 1. Horizontal axis represents time in minutes.

In figure 5.2 we illustrate a cell oscillation event and the parameters that are involved with detecting it. In figure 5.3 we see actual oscillation events that occurred in the data of user 1. In this figure there are oscillations between cell IDs 17 and 18, as well as 16 and 17.

5.3.2 Cell ID Adjacency Matrix

We represent the Cell ID weighted adjacency matrix as A, where A_{ij} is equal to the number of times that cells c_i and c_j have transitioned between each other. We also define a restricted adjacency matrix $\hat{A}$ where we use a stronger criterion to signify an edge which better suits our problem. Specifically, we signify a graph edge when two Cell IDs are involved in a cell oscillation event. An edge is created between all pairs of c_b and c_i $(i.e. \{c_b, c_1\}, ..., \{c_b, c_j\})$ if $\delta_b < \tau_b$. The edge weight between two Cell IDs is equal to the number of times that this pair of Cell IDs has been involved in a cell oscillation event.

5.3.3 Cell ID Pairwise Distances

Unless we know the physical locations of the cell towers for the Cell IDs or the physical location where the Cell ID was observed we cannot determine the spatial distance between Cell IDs. Since we have restricted our available data in this work to only Cell ID and timestamp observations (barring the use of databases such as PlaceLab and the general lack of such location information from the network operators) it is not possible to directly determine the spatial distance between Cell IDs. Nevertheless, we can get some measure of distance using the temporal data. Intuitively, the longer the time between two Cell ID observations the farther they should be from each other so we can use the difference between the observation times of c_i and c_j $(|t_i - t_j|)$ as a measure. In the rest of this section we explore different possibilities for estimating the velocity of the user between consecutive

Cell ID observations (sections 5.3.3, 5.3.3, and 5.3.3) and then we show, in a simple process, how to derive the distance between pairs of Cell IDs (algorithm in figure 5). We also provide some reasoning and methods to make sure that the resulting pairwise distances are in a metric or euclidean space (sections 11 and 11).

Velocity from holding times

If we naively assume a constant velocity v then we can calculate the physical distance between Cell IDs by multiplying $|t_i - t_j|$ with this constant v. In reality, v can range from $0km/h$ when the user is stationary to over $100km/h$ when the user is in a vehicle, however, we cannot directly measure this given the limited information that we have. Our strategy to estimate the velocity relies on the *holding time* h of a Cell ID which is related to the rate of change of Cell IDs. We assume that the longer a person has stayed connected to the same Cell ID (suggesting a lower rate of change between Cell IDs) then the slower his velocity will be when transitioning to the next Cell ID. Therefore, to get an estimate on the velocity, we can use an appropriate function $V(\bar{t})$ with domain $0 < h < \infty$, range $0 < V(h) < v_{max}$ where v_{max} is the maximum velocity, and $\frac{\partial V(h)}{\partial h} \leq 0 \, \forall h$.

We empirically determine an appropriate function upon analysis of our data by plotting the holding time (τ) against the ground truth velocity (v). First we obtain a model for the upper bound of the velocity (v_{up}) and then a model for the lower bound of the velocity (v_{low}). We conclude that the actual velocity should be within those two bounds. The exponential model $v = ae^{b\tau} + ce^{d\tau}$ was empirically found to fit best with our data.

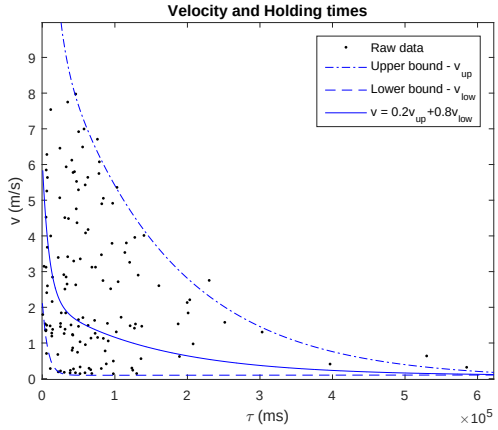


Figure 5.4: 1-cell holding time vs velocity

Using this model we fit the upper bound of the velocity v_{up} for the Cell ID holding times with the parameters $a = 11.55$, $b = -8.912E^{-5}$, $c = 10.42$, $d = -6.549E^{-6}$ and the lower bound of the velocity v_{low} with the parameters $a = 2.019$, $b = -1.367E^{-4}$, $c = 9.518E^{-2}$, $d = 1.246E^{-7}$.

Velocity from adjusted holding times

Due to Cell ID oscillations, the holding time h may not accurately represent the user's movements. Therefore we also define an *adjusted holding time* $\bar{h}$ that takes this into account by extending the holding time as long as two observations of the same Cell ID are within the Cell ID oscillation threshold τ.

Velocity from n-cell holding times

For now, we have been looking at holding times of single Cell IDs, however it may be beneficial to look at holding times of groups of Cell IDs as well. We will call these *n-cell holding times*, h^n, and we define them as the amount of time that the Cell ID has not changed from the last n Cell IDs. For example, consider the sequence in Fig. 5.5: The 1-cell holding times for Cell ID c_0 can be calculated as $h = t_1 - t_0$ and $h = t_4 - t_3$, for Cell ID c_1 as $h = t_3 - t_1$. The 2-cell holding time for the pair of Cell IDs $\{c_0, c_1\}$ is calculated as $h^2 = t_4 - t_0$. Determining a suitable n can depend on several factors: density of cell tower deployment, attenuation profile of the location where the measurements are taking place, among others. We will empirically determine a suitable value for n upon analysis of our data.

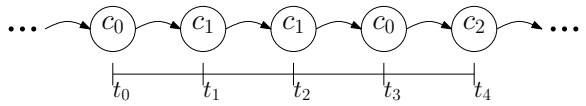


Figure 5.5: A sequence of Cell ID measurements.

For the n-cell (n=5) holding times we notice again that the exponential model is a good fit and calculate the upper bound parameters to be $a = 10.31$, $b = -4.832E^{-5}$, $c = 9.314$, $d = -5.562E^{-6}$ and the lower bound parameters $a = 2.006$, $b = -1.548E^{-4}$, $c = 9.298E^{-2}$, $d = -5.812E^{-7}$. The raw data and the fitted models are illustrated below.

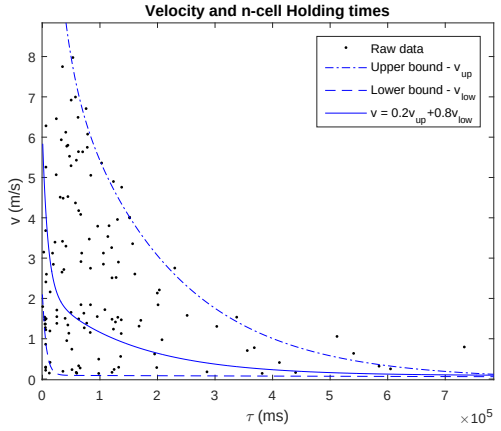


Figure 5.6: n-cell (n=5) holding time vs velocity

Using the velocity models depicted in Fig. 5.4 and Fig. 5.6, we estimate the velocity of the user as they experience handovers. We empirically find that using the estimate $v = 0.05v_{up} + 0.95v_{low}$ results in the most reliable velocity estimate and we present a small sample of the resulting velocity estimates in Fig. 5.7. Using the standard holding times, we notice that there are many regions throughout the Cell ID sequence where a non-zero velocity was estimated when in fact there was no actual change in location (see a little past the middle of the graph in Fig. 5.7). The n-cell holding times with an empirically determined n of 5 seems to filter out most of these outliers while only missing very few of the non-zero velocity situations (see the end of the graph in Fig. 5.7). Overall, these estimates look to be approximating the real velocity with some errors, however, as we will see in the Cell ID distance analysis these estimates are still desirable and useful.

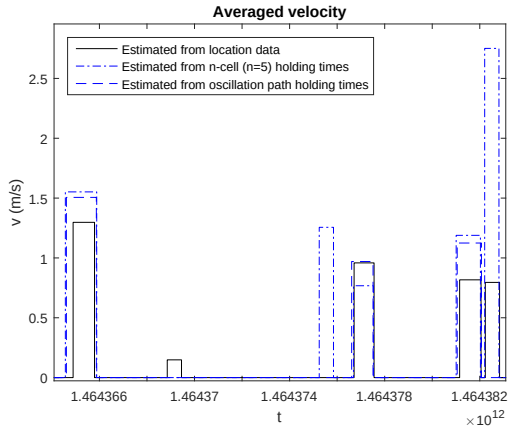


Figure 5.7: The velocity estimate of a section of the data sequence where the velocity of regions with continuous nonzero velocity are averaged.

Having estimates for velocity, we define a pairwise distance matrix D where $D_{i,j}$ is the distance between c_i and c_j and is updated as shown in the algorithm in Algorithm 5. We can use this pairwise distance matrix to cluster Cell IDs that are near each other, or even map the Cell IDs using techniques such as multidimensional scaling (MDS).

Metric Distance Matrix

Our pairwise distance matrix D is not guaranteed to be metric and can cause some clustering techniques, which work best with metric distance matrices, to have major inconsistencies or to not work at all. A metric space is defined as a space in which the following properties hold:

1. $D_{i,j} \geq 0$
2. $D_{i,j} = D_{j,i}$
3. $D_{i,j} = 0 \implies x = y$

Algorithm 5: $D_{i,j}$ updates

Data: Cell ID observations
Result: Cell ID Distance Matrix
$D_{i,j} = \infty \; \forall i, j$ **repeat**
  observe cell ID c_i at time t
  $t_i \leftarrow t$
  estimate $v_{i,j}$
  $d_{i,j} \leftarrow (t_i - t_j)v_{i,j}$
  if $d_{i,j} < D_{i,j}$ **then**
    $D_{i,j} \leftarrow d_{i,j}$
    $D_{j,i} \leftarrow D_{i,j}$
  end
  $c_j \leftarrow c_i$
until *No more observations*

4. $D_{i,j} \leq D_{i,k} + D_{k,j}$

Based on the algorithm in Fig. 5, for the distance matrix D, we can only guarantee the first three properties (1-3) but not the fourth which corresponds to the triangle inequality. Even if our distance estimates are accurate, the path from one point to another is influenced by the city infrastructure, among other physical factors, which can cause D to be non-metric (for example, one-way streets in Fig. 5.8). To estimate $\hat{D}$, the metric D, we can use a shortest path algorithm such as the Floyd-Warshall algorithm which will force the triangle inequality to hold for all i, j, k tuples.

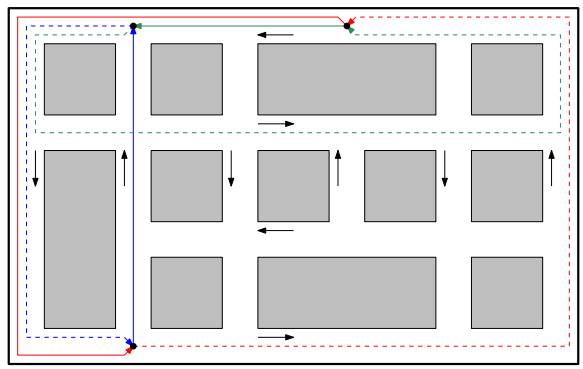


Figure 5.8: Urban setting with one-way streets resulting in non metric distances. The solid lines represent shortest paths (dashed lines are alternative, longer, paths). The sum of the blue and green paths (the paths connecting the two vertically and the two horizontally aligned points respectively) is shorter than the red path (the path connecting the two diagonal points), thus failing the triangle inequality.

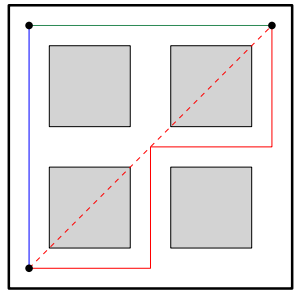


Figure 5.9: The dashed diagonal red line is the Euclidean path, while the solid red line connecting the two diagonal points is a realistic path which is not Euclidean.

Euclidean Distance Matrix

Some clustering algorithms require that the data is derived from a euclidean space but the distance matrix D is not guaranteed to belong to such a space due to the city infrastructure (among other factors) which can limit the available paths between two points (see Fig. 5.9). There are a few ways to approximate D as a euclidean distance matrix (EDM) $\hat{D}$. One way, is to project the eigenvectors of D to a euclidean space and use these projections to re-estimate D [193]. Another way, is to utilize the commonly used multidimensional scaling method to approximately map the pairwise distances to a euclidean space. Yet another way, which provides some mathematical guarantees about cluster preservation, is to use constant shift embedding (CSE) [194]. There are several such methods in the literature, each with its own advantages. It is often advantageous to apply these methods to the metric distance matrix $\hat{D}$ instead of D directly.

Cell ID Kernel Similarity Function

When working with similarity measures it is not uncommon to apply a kernel function in order to either normalize the data to a certain domain or to enrich the data with some properties of the kernel. Some common kernel functions that we can use include the Gaussian kernel, $K(D_{i,j}) = e^{-\frac{D_{i,j}^2}{2\sigma}}\ \sigma > 0$, or the Laplacian kernel, $K(D_{i,j}) = e^{-\alpha D_{i,j}}\ \alpha > 0$.
Depending on how we want to use the data after the kernel has been applied, specific kernel functions should be preferred. A Gaussian kernel might be desired if we require that the input data in our clustering algorithm be smooth. A Laplacian kernel is much less smooth than the Guassian but has similar effect on the data. Both are in the family of radial basis function kernels which accentuate radial structures (circles, spheres, and hyperspheres) in the data.

Distance Matrix from data

Now that we have estimated the velocity as described above, we can estimate the distances between Cell IDs using the process in algorithm 5 in section 5.3.3. This way we have the shortest distance between any two Cell IDs. In Fig. 5.10,

we show the ground truth distance matrix as calculated using the location data in conjunction with the Cell ID sequence. All distances are in meters and blue color indicates closeness while the yellow indicates Cell IDs being further away. In Fig. 5.11, we show the estimated distance matrix as calculated based on the n-cell holding times. We notice that the two have very similar structure. We see that there are two groups of Cell IDs (first group being approximately the Cell IDs $0-30$, $70-90$ and second group in the middle with approximately the Cell IDs $30-70$) in both distance matrices. The boundaries of the groups are less clear in the estimated distance matrix, however it yields very useful information nonetheless since we can exploit the structural similarity of the estimated distance matrices with the ground truth to derive realistic Cell ID clusters.

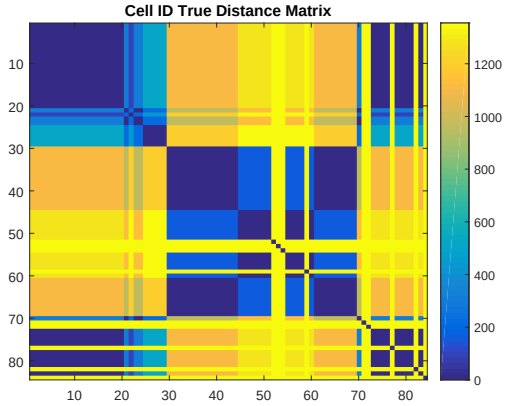


Figure 5.10: Ground truth distance matrix from real location data

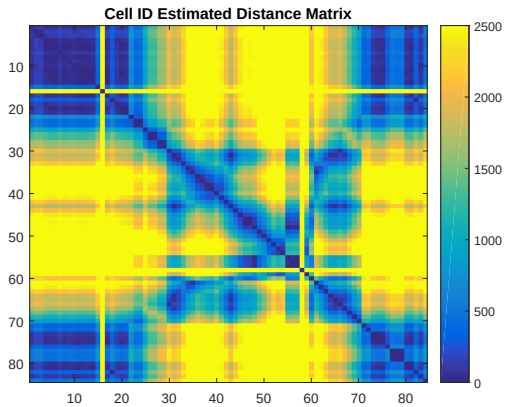


Figure 5.11: n-cell holding time estimated distance matrix.

5.4 Cell ID Clusters

5.4.1 Clusters From Maximal Cliques

As we have previously mentioned, graph connectivity information between Cell IDs can be used as a similarity metric. A possible method to determine the Cell ID clusters from graph connectivity information is to find all the maximal cliques of the graph represented by the adjacency matrix A or $\hat{A}$. This method was used in the work of Fanourakis et al. [25] with some promising results and we modify it here with a more sophisticated cluster labeling procedure to further improve it. We also propose a method to avoid using complicated labeling procedures.

A *clique* is a set of nodes (i.e. Cell IDs) where each pair of nodes in this set is connected with an edge. A maximal clique is a clique which is not a subset of any other clique. Since our graph has weighted edges (recall that edge weights are determined by the number of cell oscillation events that a pair of Cell IDs experience), we can use the edge weights to filter out possible outliers by setting an edge weight threshold e_{th}. The maximal clique that determines a Cell ID cluster is a maximal set of nodes where each pair of nodes in this set is connected with an edge of at least weight e_{th}. Let $\mathcal{G} = \{G_1, ...G_m, ..., G_M\}$ be the set of all maximal cliques of the graph (i.e. the set of all viable Cell ID clusters). It is important to note that a single node can belong to multiple clusters as depicted in Fig. 5.12.

1	1	0	0
1	**1**	1	1
0	1	**1**	1
0	1	**1**	1

Figure 5.12: Example of maximal cliques in an adjacency matrix where one cell belongs to two maximal cliques.

5.4.2 Assigning Cluster IDs

Assigning a single cluster ID to a Cell ID in the sequence is not trivial because a Cell ID can belong to multiple clusters. Therefore, we must chose the cluster $G_m \in \mathcal{G}$ that best represents the Cell ID and its surroundings. To do so, we need to specify a measure representing the *goodness of a cluster* by assigning weights to the recent Cell IDs in the sequence and to the Cell ID members of the clusters. We will then use these weighted values in a similarity measure and apply that in two versions of a self contained clustering algorithm defined in section 5.4.2.

Weights of Cell IDs in the sequence Let $\mathcal{G}^n$ be the set of all clusters containing Cell ID c_n. Furthermore, let $\mathcal{C} = \{c_1, ..., c_n, ..., c_N\}$ be a list of all observed Cell IDs (each unique Cell ID only appears **once** in this list) and $\mathcal{T} = \{t_1, ...t_n, ...t_N\}$ be a list of time in minutes since the first Cell ID measurement that each Cell ID was observed (i.e. t_n is the most recent time that Cell ID c_n was observed since we

started collecting data). Then the weight $w_T(c_i, m)$ for $c_i \in \mathcal{C}$ and $t_i < t_n$ relative to a cluster $G_m \in \mathcal{G}^n$ is defined similarly to a low pass filter as follows:

$$w_T(c_i, m) = \frac{1}{1 + \left(\frac{|t_n - t_i|}{|G_m| f}\right)^{2n}} \tag{5.1}$$

where $0 < f < 2$ and n is equivalent to the order of a low pass filter. Note that $0 < w_T(c_i) \leq 1$. Fig. 5.13 illustrates these weights using various parameters.

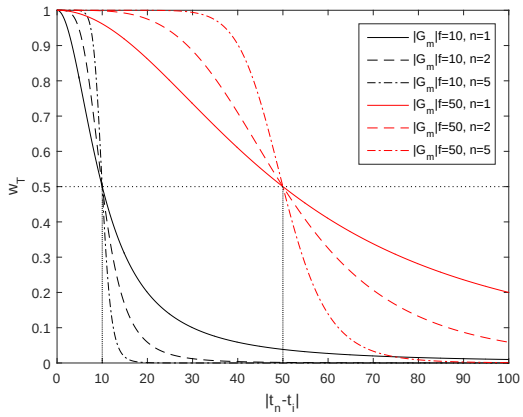


Figure 5.13: $w_T(c_i, m)$ with various parameters. The size of the cluster, $|G_m|$, along with f determine the 50% attenuation point while the order, n, determines the slope around the 50% attenuation point.

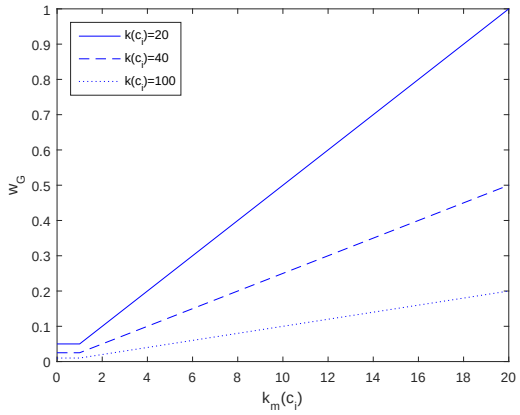


Figure 5.14: $w_G(c_i, m)$ with various parameters. The value of $k(c_i)$ determines the slope of the function.

Weights of Cell IDs in the clusters To assign weights $w_G(c_i, m)$ to the Cell IDs $c_i \in G_m$ we use the following equation:

$$w_G(c_i, m) = \frac{max(k_m(c_i), 1)}{k(c_i)} \tag{5.2}$$

where $k(c_i) > 0$ is the number of times that Cell ID c_i has been observed in total, $k_m(c_i) \geq 0$ is the number of times that Cell ID c_i has been assigned the cluster G_m in the sequence, and $0 \leq w_G(c_i) \leq 1$. Fig. 5.14 illustrates these weights using various parameters.

Choosing the best cluster Now that we have the weights of both the Cell IDs in the cluster and the Cell IDs in the sequence relative to each cluster size we can calculate the similarity of each cluster to a Cell ID c_n in the sequence. Let $S(m)$ be the similarity of $G_m \in \mathcal{G}^n$ to c_n in the sequence, then the cluster assignment is determined as follows:

$$\arg\min_m S(m) = \sum_{c_i \in G_m} \left(w_T(c_i, m) - w_G(c_i, m)\right)^2 |G_m|^{-1} \tag{5.3}$$

The factor of $|G_m|^{-1}$ in equation 5.3 gives more importance to larger clusters. Putting it all together, we formulate the Cell ID clustering and labeling algorithms in Algorithm 6 and Algorithm 7 which use the cell transition adjacency matrix A and cell oscillation adjacency matrix $\hat{A}$ respectively.
In the pseudocode in Algorithm 6, when a Cell ID is observed (line 5), the time of the observation is logged (line 15), then the transition matrix A is updated (lines 17-18). If there is a change in A, the new cliques are calculated (line 18). Then, to label the Cell ID with a cluster, the similarity of the observation $S(m)$ is calculated against all clusters m and the most similar cluster is selected (lines 20-21). If two sequential events are separated by more than the $timeout$ time (line 16) then there may have been some information loss and therefore we should not associate those two events.

In the pseudocode in Algorithm 7, when a Cell ID is observed (line 5), the time of the observation is logged (line 16) and the time since the last observation of that Cell ID is calculated (line 15). If there is a cell oscillation event (line 18), the special adjacency matrix is updated and then the cliques are updated as well (lines 19-20). The Cell ID is assigned a cluster (lines 22-23) in the same way as in the pseudocode in Algorithm 6. Note that $\tau_n \in \mathcal{T}^o$ from the pseudocode in Algorithm 7 is less than or equal to the $timeout$ from the pseudocode in Algorithm 6.

Assigning Cluster IDs After Cluster Merging To simplify the labeling process we can merge similar or overlapping clusters together such that a Cell ID can only belong to one cluster. The procedure for merging the clusters is simple: any two or more clusters containing one or more Cell IDs which are the same are merged into one cluster. Labeling the Cell ID in the sequence with these updated clusters becomes trivial since a Cell ID can only belong to one cluster.

Algorithm 6: Clustering and labeling w/ transition matrix

Data: Cell ID observations
Result: Cluster-labeled sequence

$\mathcal{C}, \mathcal{T}, \mathcal{S}, \mathcal{G} \leftarrow \emptyset$
$A \leftarrow \mathbb{R}^{0\times 0}$
$e_{th} \leftarrow z \in \mathbb{Z}_{\geq 0}$
repeat
  observe Cell ID $\tilde{c}$ at time $\tilde{t}$
  if $\tilde{c} \notin \mathcal{C}$ **then**
    $\mathcal{C} \leftarrow \{\mathcal{C}, \tilde{c}\}$
    $A \leftarrow \begin{bmatrix} A & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$
    $\mathcal{T} \leftarrow \left\{\mathcal{T}, \tilde{t}\right\}$
    $\mathcal{G} \leftarrow \{\mathcal{G}, \tilde{c}\}$
  end
  $\tilde{c} \equiv c_n \in \mathcal{C}$
  $\mathcal{S} \leftarrow \{\mathcal{S}, \tilde{c}\}$
  $t_n \in \mathcal{T} \leftarrow \tilde{t}$
  if $t_n - t_k < timeout$ **then**
    $A_{n,k} \leftarrow A_{n,k} + 1$
    update $\mathcal{G}$
  end
  $m \leftarrow \arg\min_m S(m)$
  assign cluster ID m to $\tilde{c} \in \mathcal{S}$
  $k \leftarrow n$
until *No more observations*

Algorithm 7: Clustering and labeling w/ oscillation matrix

Data: Cell ID observations
Result: Cluster-labeled sequence

$\mathcal{C}, \mathcal{T}, \mathcal{S}, \Delta, \mathcal{T}^o, \mathcal{G} \leftarrow \emptyset$
$\hat{A} \leftarrow \mathbb{R}^{0\times 0}$
$e_{th} \leftarrow z \in \mathbb{Z}_{\geq 0}$
repeat
  observe Cell ID $\tilde{c}$ at time $\tilde{t}$
  if $\tilde{c} \notin \mathcal{C}$ **then**
    $\mathcal{C} \leftarrow \{\mathcal{C}, \tilde{c}\}$
    $\hat{A} \leftarrow \begin{bmatrix} \hat{A} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$
    $\mathcal{T} \leftarrow \{\mathcal{T}, \tilde{t}\}$
    $\Delta \leftarrow \{\Delta, 0\}$
    $\mathcal{T}^o \leftarrow \{\mathcal{T}^o, -1\}$
    $\mathcal{G} \leftarrow \{\mathcal{G}, \tilde{c}\}$
  end
  $\tilde{c} \equiv c_n \in \mathcal{C}$
  $\mathcal{S} \leftarrow \{\mathcal{S}, \tilde{c}\}$
  $\delta_n \in \Delta \leftarrow \tilde{t} - t_n$
  $t_n \in \mathcal{T} \leftarrow \tilde{t}$
  update $\tau_n \in \mathcal{T}^o$
  if $\delta_n < \tau_n$ **then**
    $\hat{A}_{n,k} \leftarrow \hat{A}_{n,k} + 1 \; \forall k \mid t_n - \delta_n \leq t_k \leq t_n$
    update $\mathcal{G}$
  end
  $m \leftarrow \arg\min_m S(m)$
  assign cluster ID m to $\tilde{c} \in \mathcal{S}$
until *No more observations*

5.4.3 Clusters From Oscillation Paths

Consider an oscillation event in the sequence; we can say that all cells involved in this oscillation event can form a cluster together. In the case when multiple oscillation events are intertwined with each other then all the cells involved are a cluster as well. For example, let $\mathcal{S} = \{c_a, c_b, c_c, c_d, c_b, c_e, c_d, c_f, c_g\}$ be a part of the Cell ID sequence. We notice that there are two oscillation events, one involving cell c_b and another involving cell c_d. We also notice that these two events overlap. Thus, we define an *oscillation path* as the combination of overlapping oscillation events. In our example this would result in an oscillation path starting from the first occurrence of cell c_b and end at the last occurrence of cell c_d and the resulting cluster would contain the cells c_b, c_c, c_d, and c_e.

The same labeling techniques used in the maximal clique method can also be used here.

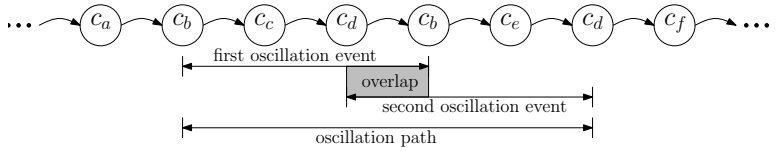


Figure 5.15: An example of an oscillation path.

5.4.4 Clusters From Pairwise Distances

There are many clustering methods in the literature that rely on pairwise distance information. In this section we will summarize a few of the most popular methods and techniques that can be used with the pairwise distance matrices D, $\hat{D}$, and $\tilde{D}$ that we defined in a previous section.

Agglomerative Clustering A ”bottom-up” approach of hierarchical clustering where each observation (in our case Cell ID) starts as its own cluster, and pairs of clusters are merged as one moves up the hierarchy. Different distance measures can be used to determine the distance between two clusters (not necessarily the distance between two Cell IDs but rather two groups/clusters of Cell IDs).

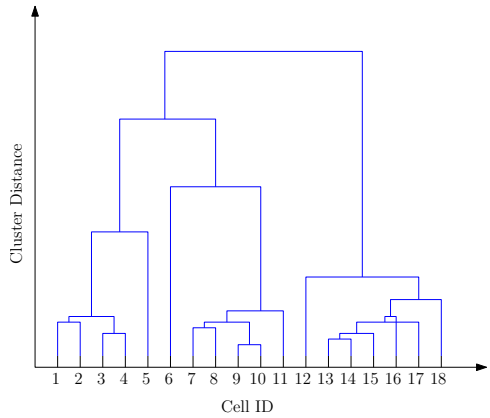


Figure 5.16: A dendrogram where clusters are connected depending on a cluster distance measure.

Density-Based Clustering Using the distance matrix that was estimated in section 5.3.3, we can use a density based clustering method to cluster Cell IDs that are in close proximity to each other and thus may represent a single place that the user has visited. A widely used and built-upon density-based clustering method is DBSCAN [195]. Given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors) , marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away). It requires two parameters to be provided: ϵ specifying the maximum distance for two points to be considered as neighbors, and $MinPts$ specifying the minimum number of neighbors that a point needs to have in order to be considered as a core point. This and other density-based clustering methods are versatile and work well with oddly shaped clusters (for example, rings, crescents, etc.). Furthermore, it is a trivial matter to apply DBSCAN to the pairwise distance matrices that we have defined in this work.

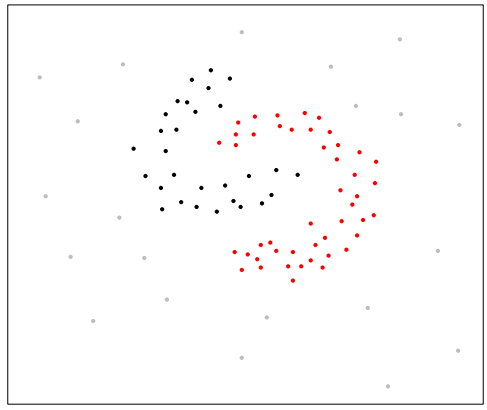


Figure 5.17: DBSCAN algorithm was used to determine the clusters. A cluster represented by black points and another by red points. Gray points are outliers.

Spectral Clustering These techniques make use of the eigenvalues of a similarity matrix such as D. A common spectral clustering technique is the *normalized cuts* algorithm which uses the symmetric normalized Laplacian of D and then computes the eigenvalues and eigenvectors, finally k-means can be used to find clusters in this spectral space.

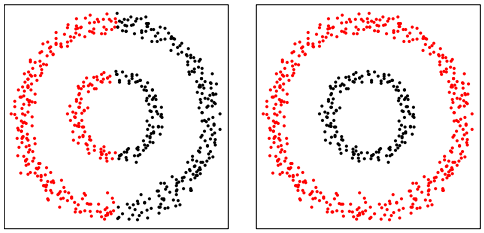


Figure 5.18: On the left the result of the k-means clustering algorithm of some data (black data points belong to one cluster and red data points belong to another cluster) and on the right the result of the k-means clustering algorithm performed on the spectral domain of the same data (and then plotted back into the original domain).

5.4.5 Cell ID Clusters in the Data

We first filtered out *transitional Cell IDs*, that is Cell IDs that are only observed for brief instances while the user is moving from one place to another, from the data since they could not represent a significant place. Through empirical observation, we conclude to label a Cell ID as transitional if its maximum holding time is less than 5 minutes, and if its median holding time is less than 2 minutes. This meant that for user 1, 16 of the Cell IDs were transitional and for user 2, 20. To recover a ground truth for the clusters, we used the DBSCAN algorithm to find the clusters from the ground truth distance matrix (the one calculated from the location data). We empirically find that setting the parameter ϵ to 100 (meters) and $MinPts$ to 4, results in a good clustering. The DBSCAN algorithm gives us two clusters, one with Cell IDs in the regions $1 - 17$, 73, and 81, and the other with Cell IDs in the region $56 - 58$ (seetable 5.1) which confirms our earlier observation about possible groups of Cell IDs when we looked at the distance matrix. Inspecting the locations where these Cell IDs were detected, and by cross checking with the user, we conclude that the two clusters represent the user's work place and home respectively.

Method	Clusters	Difference
Ground truth	$\{1, 2, 4, 6, 8, 10, 12, 13, 14, 15, 16, 73, 81\}$ $\{56, 57, 58\}$	N/A
Distance matrix	$\{1, 4, 6, 8, 10, 12, 13, 14, 73, 81\}$ $\{56, 57\}$	$\{-2, -15, -16, -73\}$ $\{-58\}$
Graph	$\{1, 10\}$ $\{4, 6, 8, 12, 13, 14, 73, 81\}$ $\{56, 57\}$	$\{+1, +10\}$ $\{-1, -2, -10, -15, -16\}$ $\{-58\}$

Table 5.1: Comparison of clusters

Performing DBSCAN, with the same parameters as before, on the n-cell estimated distance matrix we find that the resulting clusters are in the same regions (see table 5.1). However, there are 4 Cell IDs missing from the first cluster and 1 from the second cluster compared to the ones found using the ground truth distance matrix. Since we used the same clustering algorithm and parameters, we suspect that discrepancies in the distance matrix calculation due to inaccuracies in the velocity estimates are the cause of these differences in the cluster. We can also compare the clusters from the ground truth distance matrix with those discovered using the maximal clique method in section 5.4.1. We notice that, when compared with the ground truth, the first cluster is split into two clusters with the second cluster missing only 3 Cell IDs. The last cluster is identical with the n-cell clusters.

5.5 Discussion

Although cloud services and cloud computing are gaining traction in recent years, they can be major sources of privacy concerns for users. The continuous improvements in processing power and memory in mobile devices allows us to perform many privacy sensitive tasks on the device itself with little compromise. In this chapter we defined similarity metrics to describe Cell ID distances in a privacy-conscious manner. We then used these metrics in order to estimate clusters which represent significant places for a user in a real life study. Based on our results we can conclude that the similarity metrics that we described are useful since they yield clusters that are similar to the ones produced when using real distance measures obtained through the location service of the mobile device. Using our distance metrics we identified the same number of clusters as when using the real distance measures (two clusters) and these clusters were almost identical (a maximum difference of 4 Cell IDs). Our estimated distance matrix, although the distances were overestimated, had a very similar structure to the ground truth distance matrix, i.e. a constant scaling of the estimated distance matrix would result in an accurate representation of the Cell ID pairwise distances. The maximal clique method produced similar results. The main reason one might choose the maximal clique method over the distance based method would be because the maximal clique method is more lightweight in terms of computations and memory,

however, one might want to use the distance based method because it provides distance estimates. Finally, we show that it is possible to introduce such a privacy-conscious technique to provide important location context information to location based services in a realized application.

The most significant cause of errors is a direct result of the inaccuracies of the velocity estimates. In our future work we plan to further improve the velocity estimates by using more sophisticated models that take into account not only the holding time at that instant but also other more long term metrics about the holding times and their patterns, therefore improving the clustering accuracy. Considering that the Cell ID clusters are not an indicator of geographical position but rather a signifier of an abstract place where a mobile user is in (such as home, work, etc.), the application of this research is limited to services which need a location context input without compromising the user's privacy or using a significant amount of resources on the device. Such services include but are not limited to: automatic user interface modification to match the location context of the user, tracking of time spent in home, work, or other significant place, providing reminder notifications based on location context. Other, research-oriented applications of automated location context discovery are abundant such as in Stals et al. [196] where they correlated user emotions with the places they visited.

The maximal clique method was deployed in a mobile phone application with the aim of serving as an additional feature to be used in a machine learning algorithm that can detect the intimacy state of the user in the work of Gustarini et al. [28]. The application used semantic location and time spent in that location. The maximal clique algorithm was used to determine the semantic location (home, work) of the user instead of using the GPS sensor so as to conserve battery usage and preserve user privacy. Overall, the maximal clique algorithm along with the time spent in the detected clusters helped to define intimate vs non-intimate environments. Since intimacy detection should not invade user privacy (it is counterintuitive to invade intimacy to detect it) the use of this algorithm, which did not require access to the user's location data (a very intimate set of data), helped achieve the goals of the experiment.

Chapter 6

Efficacy Evaluation of Opportunistic Data Mixing

6.1 Introduction

Mobile crowd sensing leverages the number of user-companioned devices, including mobile phones, wearable devices, and smart vehicles, and their inherent mobility to collect information such as location, personal and surrounding context, noise level, and more [197]. The users, acting as sensors, have a certain expectation of privacy about the data they might be sharing and often do not trust that is it possible to hide their identity while at the same time provide usable data [22]. Providing data privacy in crowd sensing or other participatory data collection context has been an important task that ensures that the participants privacy is protected (for example, data cannot be traced to the individual) while the data is being collected at large scales without bias stemming from privacy-aspects (for example, participants switching off their phone in certain contexts).

There are several elements of the data collection process that can be exploited to reveal sensitive information about the participants: the data communication channel, the reporting of the data, and the data itself. The communication channel can be exploited by man in the middle attacks where someone can intercept the message, read it and potentially manipulate it, and then relay it to its original destination. Securing the communication channel from third parties that might want to intercept the data can be achieved using data encryption techniques. In the reporting stage, each participant sends their data to the entity collecting the data (a researcher or a company), from here on referred to as the *data collector*. As a result, the data collector can easily know which data belongs to each participant by looking at which of the participants sent it. Giving pseudonyms to the participants can help mitigate this but it is still not completely safe. The data collector will still know that a certain batch of data belongs to a certain pseudonym which can be compromising depending on the content of the data. Even one piece of identifiable data will allow the data collector to know that all the data in that batch with the same pseudonym belongs to the same user. For this reason,

mix networks were introduced in the data reporting process. These networks mix the batches of data from each participant and send it to the data collector which then has no way to directly trace the source of a batch of data. The data itself can be used in inference attacks where an attacker analyzes the data and is able to cross reference with public data or maliciously obtained data (ex. spying) in order to identify the participant who generated this data. For this, there has been a lot of research in data obfuscation which provides some level of anonymity (k-anonymity, l-diversity, t-closeness, and others). Chapters 2.2 and sec 2.3 describe privacy metrics and data obfuscation techniques in more detail and provide more information and shortcomings of the currently available anonymity measures.

A generic data collection scheme is shown in figure 6.1 where users collect data in some environment and then send the data through an optional mix network that can either be a geographical zone in their environment or a separate network. The data is eventually communicated to the data collector who may chose to use data obfuscation techniques to provide privacy to the users. The communication channel is represented by arrows.

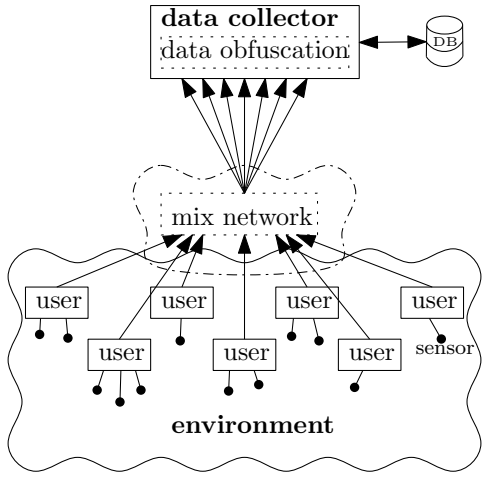


Figure 6.1: A diagram of a generic data collection scheme.

Other privacy and security related requirements are outlined by Giannetsos et al. [63] however, these are out of the scope of this chapter. These include privacy-preserving resilient incentive mechanisms and fairness (users should receive credits and rewards for their participation without associating themselves with the data or the tasks they contributed), communication integrity confidentiality and authentication (all entities should be authenticated and their communications should be protected from any alteration and disclosure to unauthorized parties), authorization and access control (participating users should act according to the policies specified by the sensing task), data-centric trust (Mechanisms must be in place to assess the trustworthiness and the validity of user submitted data), and accountability (entities should be held accountable for actions that could disrupt the system operation or harm users).

6.1.1 Contributions

This chapter concerns itself with enhancing the privacy of participants during the reporting phase of a data collection scheme. Although mix networks have helped in this regard, they often rely on some centralized entity or specific location in order to mix the batch of data. The concept of slicing and mixing, which is described in chapter 2.5, allows for novel mixing strategies that can be implemented in an opportunistic sense but there is little proof as to their performance and effectiveness with real world data.

We will focus specifically on evaluating the efficacy of mixing the data, as part of a slicing and mixing strategy, in a fully opportunistic way with the goal of achieving a uniform distribution of the data among all participants. We assume that participants are mobile and generate data by using sensors or answering surveys and that they regularly cross paths with at least one other participant in order to exchange data in a peer to peer manner. Furthermore, we require that all participants together form a connected graph with respect to who they meet. The mixing strategy we use is very basic so as to provide baseline results that can later be used to evaluate more complex strategies.

In section 6.2 we introduce and describe our data mixing scheme, in section 6.3 we verify the mixing scheme in a simulated environment. Finally, in 6.4 we summarize the results and provide design implications.

6.2 Privacy-conscious Data Shuffling

From here on we will refer to the participants (users in Figure 6.1) who are generating the data as *sensor nodes* or just *nodes*. A simple opportunistic mixing scheme consists of the following steps:

1. Nodes perform their normal daily routines while also collecting data.
2. At some point all the nodes have finished collecting data.
3. Nodes continue performing their normal daily routines, this time **without** collecting data.
4. Node i comes *into communication range* with another node j.
5. Each node randomly *selects a subset of their data* which will be sent to the other.
6. The nodes *exchange the data* between each other.
7. Keep performing steps 3 through 6 until certain stopping criteria is reached.

Note that steps 2 and 3 can be removed, however, in order to keep our evaluations manageable we will keep them. Furthermore, as we mentioned earlier, we require that the nodes form a connected graph so that the entirety of the data can be uniformly shuffled. If there are any disconnected subsets of nodes the data will have no way to be communicated between those subsets, only within them.

In contrast, other schemes which use mix networks instead of opportunistic peer to peer slicing and mixing, either compress steps 3 to 7 in one step where the nodes send their data to a centralized mix network, or require that in step 4 the exchange occurs in a predefined area where at least k nodes regularly enter at some point of their routine.

In figures 6.2 through 6.5 we illustrate a simple scenario with two participants. In this example scenario, a researcher wants to collect noise pollution data in a region of a city but it would be too costly to install sensors throughout this region. The researcher's solution is to equip a small number of citizens with a noise and GPS location sensor that automatically collects data and let the participants go about their daily routine.

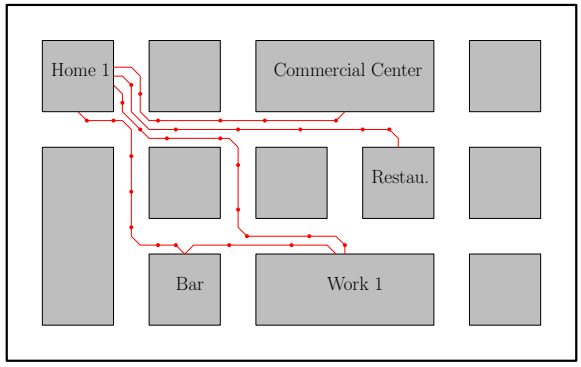


Figure 6.2: An example of a participant who collects data (solid dots) while they are moving along paths (lines) performing their regular routine.

In figure 6.2 a participant commutes between a set of places like home, work, commercial center, bar, restaurant. The red lines represent the paths they use to go between those places during their daily routine. At the same time, the participant is collecting data. The solid red dots represent the places where they have collected data.

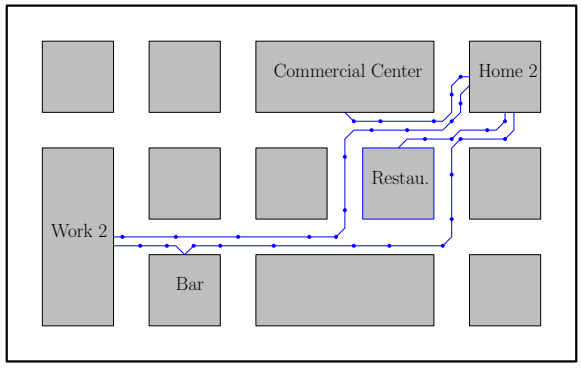


Figure 6.3: An example of another participant.

In figure 6.3 we see the paths and data points of another participant. If each of the participants would directly send their individual data, the researcher could easily figure out where each participant lives, works, shops, and goes out to. To solve this privacy issue, it would help if the data were mixed together so that the researcher does not know which specific participant collected a certain data point.

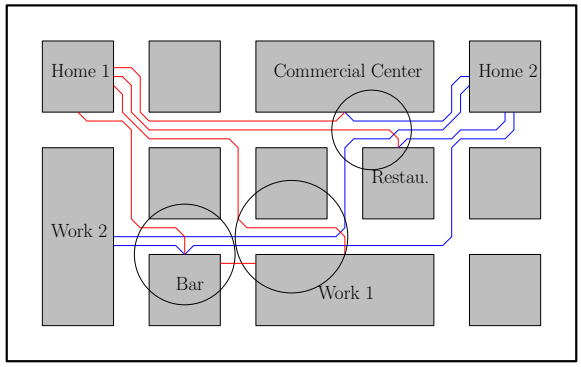


Figure 6.4: The paths of the two participants displayed on the same map. Circled areas indicate regions where they might be able to exchange data.

In order to mix the data, the participants could send it to a trusted entity whose purpose would be to mix the data before reporting it to the researcher. But what if the participants do not trust such a solution? In figure 6.4 we overlayed the paths that the two participants take in their daily routines. We notice that there are certain regions (circled areas) where they come into relatively close proximity with each other. We can take advantage of this to implement an opportunistic mixing strategy where the users mix their data between each other whenever they are close enough that they can wirelessly transmit their data. The detection of proximity and transmission of data would be managed automatically by the sensor device software so that the participants do not have to actively perform this task or even be aware of it.

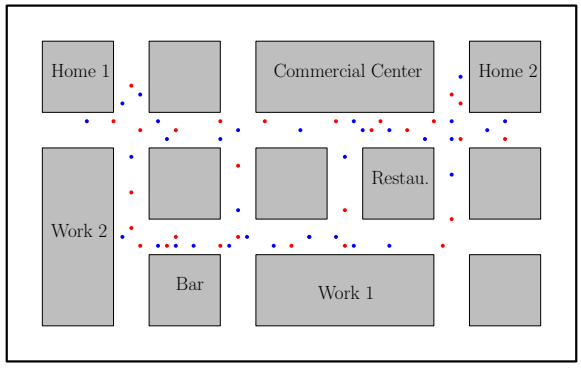


Figure 6.5: Data ownership after mixing.

After a thorough mixing of the data, each participant can then transmit their data to the researcher. In figure 6.5 we see which data each participant sent to the researcher, red for the first participant, and blue for the second participant. As we can see, the researcher will have a good sample coverage of the region and also have a much harder time to figure out where each of them lives, works, etc. Because we only used two participants in this example scenario, the researcher can still deduce some sensitive information. For example, they can claim that one of the participants lives in "Home 1" with a 50% probability. This is why it is also important to involve an adequate number of participants in any scientific study.

6.2.1 Data exchange

When two nodes come within communication range of each other, each node randomly selects **half** of the total number of data, M, that they have in their possession to exchange with the other. This value can be adjusted individually on each node to adjust their data exposure and either reduce or increase the potential amount of personal data that they might share at each transaction depending on the privacy requirements and/or trust metrics. The specifics of this adjustment are not explored in this paper and we keep the amount of data that each node exchanges at $\frac{M}{2}$ as it is optimal for reaching a uniform distribution of data in the least number of shuffles. This fact can be easily verified by looking at the number of ways there are to choose k data from M given by the binomial coefficient which can be calculated using the equation below:

$$\binom{k}{M} = \frac{M!}{k!\,(M-k)!} \tag{6.1}$$

Each shuffle becomes more random as the binomial coefficient increases in value. If we set k to be some fraction $x \in [0, 1]$ of the total data M, $k = xM$, then we can analytically verify that the value of x which maximizes the result in the equation above is $\frac{1}{2}$.

6.2.2 Stopping criteria

There are two different stopping criteria that can be used to signify that the data has been sufficiently shuffled (uniformly distributed) and that it is safe to send it to the data collector. The first one is based on each nodes perception of how well the data is mixed. Each node can keep track of the data that they come into contact with and measure the probability that they encounter some specific piece of data. Since the data may be encrypted, the nodes must keep track of the encrypted data or a shorter hashed version of the encrypted data which can come paired up with the encrypted data. Once the probability is close to being uniform across all data, then they can stop the shuffling process since this indicates a near uniform mix. This might work well when there are not many nodes, but as the number of nodes increases, the time it takes to verify the uniformity of the mix also increases. The second set of stopping criteria is based on the properties of the graph like closeness centrality. If, in addition to data, the nodes exchanged information about

their graph connections (nodes that they have previously encountered) or there is prior knowledge about the graph, then they can estimate the number of exchanges that they need to perform before the data is near uniformly mixed.

Closeness centrality to estimate stopping criteria. Closeness centrality is a measure of the degree to which an individual node is near all other nodes in the network. In order for each node to calculate its closeness centrality it needs to know its distance to all other nodes. This is trivial when there is global knowledge about the graph, however, it may not always be the case, especially when there is no trusted party to provide this information. When there is no prior graph knowledge, each node i needs to communicate its personal adjacency matrix A^i in addition to the data at each exchange. A^i should initially indicate which nodes are directly connected (one hop) along with the edge weight as it is calculated by the node (in this case, edge weight is equivalent to the number of times that the node has encountered each of its one hop neighbors). Then the node can update A^i by combining all the personal adjacency matrices it has acquired (A^j, A^k, etc.) from other nodes. To update A^i when the node receives another node's personal adjacency matrix A^j we perform the following operation:

Algorithm 8: Procedure to combine A^i with A^j

Data: A^i, A^j
Result: updated A^i
if $A^i_{k,l} = \emptyset$ *and* $A^j_{k,l} \neq \emptyset$ **then**
    $A^i_{k,l} \leftarrow A^j_{k,l}$
end
if $A^i_{k,l} \neq \emptyset$ *and* $A^j_{k,l} \neq \emptyset$ **then**
    $A^i_{k,l} \leftarrow min\left(A^i_{k,l}, A^j_{k,l}\right)$
end

Finally, performing a shortest path algorithm such as the Floyd-Warshall or Dijkstra algorithm can reduce the redundancies and update the paths in A^i. The process is illustrated in figure 6.6

The closeness centrality can then be calculated for each node and by each node using the information in their respective adjacency matrix. The stopping criteria for the number of exchanges necessary to sufficiently mix the data can then be estimated from the adjacency matrix and closeness centrality information using empirical data which we will show in section 6.3. It is important to note that if the graph of the nodes is known to everyone, encrypting the communication channel becomes even more vital for the protection of the security and privacy of nodes against malicious nodes.

	n_1	n_2	n_3	n_4
n_1	0	2	1	4
n_2	2	0	3	2
n_3	1	3	0	5
n_4	4	2	5	0

	n_3	n_4	n_5	n_6
n_3	0	3	4	2
n_4	3	0	1	3
n_5	4	1	0	2
n_6	2	3	2	0

	n_1	n_2	n_3	n_4	n_5	n_6
n_1	0	2	1	4	5	3
n_2	2	0	3	2	3	5
n_3	1	3	0	3	4	2
n_4	4	2	3	0	1	3
n_5	5	3	4	1	0	2
n_6	3	5	2	3	2	0

Figure 6.6: Left: two personal adjacency matrices. Right: the combined outcome where overlapping weights are assigned the minimum of the two values. Grey values indicate unmeasured values calculated by the shortest path algorithm.

6.3 Experiment and Results

6.3.1 Experimental setup

To verify our data mixing scheme, we perform simulations using artificial parameters as well as simulations using real mobility data from the MDC dataset. The data mixing occurs in shuffling rounds that consist of either a group of markov chain state transitions (representing data exchanges) based on the transition matrix or a full day (24 hours) of proximity events in the real mobility data simulations.

At each time t, a node i will exchange data with a node j either with a probability based on the $A_{i,j}$ element of the transition matrix A for the artificial parameter simulations or based on the proximity of the two nodes in the real mobility data simulations. At each shuffle we take note of what data each node has.

In order to get representative probability distributions of the data, we run 30000 trials of the simulation with each of the parameter sets (a parameter set consists of the following: the number of nodes, data size per node, and the transition matrix or proximity events). This number was selected because it gives us a confidence level of 99% based on the equation $n \geq \frac{log(a)}{log(1-p)}$ to calculate the number of trials necessary given the probability of the occurrence of an event p and the confidence level $1 - a$ that we require. In our case we seek to be confident of events that occur with a probability of at least $p = 0.00015$, that is to say that our probability distributions in our results will have a granularity of 0.00015. We choose $1 - a = 0.99$ which is equivalent to being 99% confident in our results.

For our experiments, the total number of nodes, N, and the amount of data items, M, that they start with is selected as described in section 14. To keep track on how the data flows throughout the network we make sure that each node's initial data is uniquely identifiable by labeling them with integers. For example, if

we set the number of data to 6, then node 1's data will consist of the numbers from 1 to 6, node 2's data will consist of the numbers from 7 to 12, etc. In this way we can easily check how uniformly the data has been distributed by evaluating the probability distribution of each number being in any specific node at the end of the shuffle.

Artificial Parameter Simulations Setup

Initially, we performed simulated experiments with artificial parameters to illustrate and validate the data shuffling procedure. Node mobility is artificially simulated by using three different Markov models where each one is defined by a transition matrix A. The three models consist of a best case scenario transition matrix (equivalent to a group of co-workers or students enrolled in the same course), an intermediate case scenario (equivalent to shift-based co-workers), and a worst case scenario transition matrix (equivalent to otherwise unrelated commuters crossing paths on their way to their individual workplaces). The specifics of the transition matrices are described in section 14.

Real Mobility Data (MDC) Simulations Setup

We use real user mobility traces from the data of the Mobile Data Challenge (MDC) which include GPS traces of real mobile users [198, 199]. The data we used included 191 users and spanned over a year of data for some of the users. To normalize and be able to compare with the artificial cases, we define one shuffling round as a single day and we analyze up to 100 contiguous days (i.e. 100 shuffling rounds) of GPS traces for each trial.

For some users, we might have less than 100 days of data, when we reach the end of the data without having completed the 100 shuffling rounds we cycle from the beginning until we reach the desired number. For example, if a user set only has 50 days worth of data, we will go though his GPS data twice to complete the 100 day trial.

In most cases we have more than 100 days of data for each user set. In this case, since we limit our simulations to 100 shuffling rounds consisting of 100 contiguous days, we make sure that we select 100 contiguous days when the user set is sufficiently active based on two criteria in order of priority: the median of the number of proximity events between all pairs in the user set, and the total number of proximity events.

We assume an exchange of data between two users can be performed under the following conditions:

- They are within 50 meters or less of each other. We call this a *proximity event* since they are within direct communication range of each other.
- They have not exchanged data between each other in the past 30 minutes.

In these experiments we do not consider the bandwidth or throughput of the data transmission and assume that it can be instantaneously exchanged when two users are within communication range.

For each of the 30000 trials for the MDC data simulations, a random subset of N users was selected from the 191 in such a way that they formed a connected graph with a minimum edge weight of 10 (in this case, edge weight indicates the number of exchanges between two users during the entire duration of the study). With this random selection, when we use $N = 10$ the average number of hops between the two most remote users was 9 and the median number of hops between any two users was 3 (which resembles a line topology). The user set was unique in each trial of our simulations, i.e. no two trials had the same set of 10 random users.

We ran an additional simulation with the MDC dataset in which instead of selecting N random users, we selected N users that formed a *clique* (i.e. a fully connected topology with maximum distance of one hop between any two users). Again, we limited the edge weight to be equal to or above 10. During our experiments we discovered that there were not enough cliques of size $\geq N$ in the dataset to justify doing 30000 trials. The total number of maximal cliques (cliques that are not subsets of larger cliques) in the dataset is 890 and the number of cliques with size of at least N is often much smaller than that. It is redundant to perform more trials than there are number of cases because this means that the same case will need to be repeated several times to reach the desired amount of trials. However, to get meaningful statistics it was necessary to do a much larger number of trials than there were number of cliques. To remedy this, we relaxed the requirement for the cliques and allowed ourselves to combine cliques to form a set of N users. The exact procedure by which we combined the cliques is described in Algorithm 9. This algorithm allowed us to generate much more than 30000 different user sets

Algorithm 9: Procedure to combine cliques

```
   Data: The user cliques
   Result: A well connected user set
1  userSet ← ∅
2  while size(userSet) < N do
3      cliq ← random clique
4      if size(cliq) < 0.5 (N − size (userSet)) then
5          go to line 3
6      end
7      userSet ← userSet ∪ cliq
8      if size(userSet) > N then
9          userSet ← select N users ∈ userSet
10     end
11 end
12 if userSet not connected then
13     go to line 1
14 end
```

as evidenced by the results in our simulations where for $N = 10$ there were no two trials with the same user set in all of the 30000 trials. Furthermore, $N = 10$ resulted in user sets with the average number of hops between the two most remote users at 4 and a median number of hops between any two users of 1.

Sim. type	**Transition Matrix**	{**# of Nodes, # of Data**}
best case	fully connected topology w/ prob of no transaction 0 and prob of transation with each of the other nodes $\frac{1}{N-1}$	{10,6} {30,6} {10,20}
intermediate case	line topology w/ prob of no transaction 0.5 for edge nodes and 0 for all other nodes	{10,6} {30,6} {10,20}
worst case	line topology w/ prob of no transaction 0.8 for edge nodes and 0.6 for all other nodes	{10,6} {30,6} {10,20}
MDC data random	GPS traces of a random selection of users	{10,6}
MDC data cliques	GPS traces of cliques of users	{10,6}

Table 6.1: Simulations performed showing total nodes and total amount of data per node for each simulation. There are 11 simulations in total.

Parameter Sets

The number of nodes and number of data items for the MDC data simulations was selected after the artificially simulated cases where we verified that the number of data items did not significantly affect the number of shuffles needed since we always exchange half of a node's total data (as per the protocol discussed in section 6.2.1). We chose to simulate only 2 representative cases with the MDC dataset: choosing a connected set of random users, or choosing users that form cliques in the adjacency matrix. Other cases would be redundant since we already show the effects of changing the number of nodes and data items with the artificial parameter simulations. All simulations are summarized in Table 6.1.

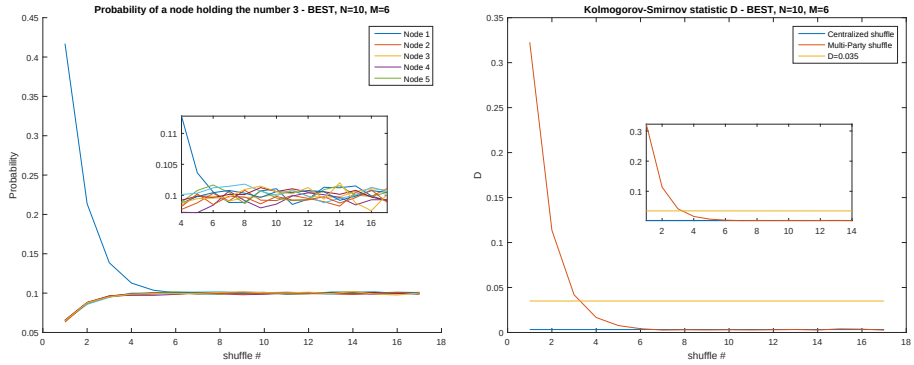
6.3.2 Performance Criteria

The performance criteria that we mainly use is the Kolmogorov-Smirnov test with a uniform distribution of $\frac{1}{N}$ as the reference distribution. With this test, we measure the absolute error between the distribution of the data in our experiment and the ideal uniform distribution. As a result of our experimental setup we are able to perform this test after every shuffling round in our experiment allowing us to see the exact number of shuffles needed to achieve a near uniform mix. For illustrative purposes we first take a look at the probability of holding a specific data item for each node at each shuffling round.

6.3.3 Results

Results Using Artificial Parameters

Intuitively, the more shuffles we do then the more uniform the distribution of the data should be. This intuition is verified in figure 6.7a where we clearly see that the probability of holding a specific data item (as an example we use the data item with number 3) approaches an ideal probability with amplitude $\frac{1}{N}$ as the number of shuffles increases, where N is the number of nodes. Since node 1 is the initial holder of the data item with number 3, it starts with the highest probability in the initial stages and as it shares data with all the other nodes the probability evens out.

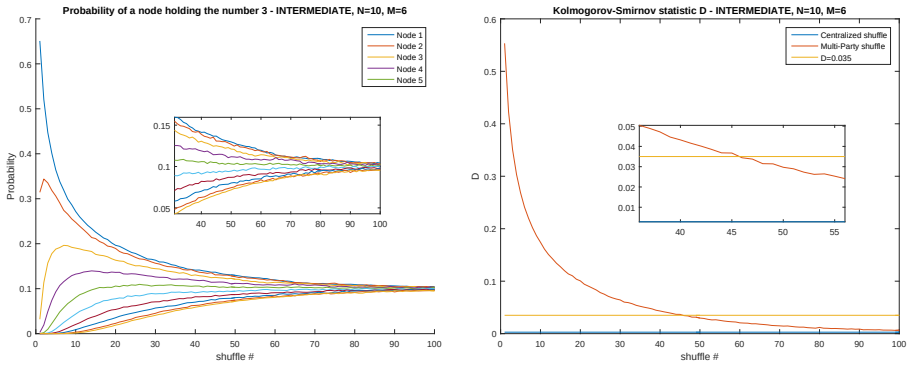


(a) The probability of the number 3 being at different nodes at different number of shuffles (subplot is of a magnified region)

(b) Kolmogorov-Smirnov test at different number of shuffles(subplot is of a magnified region)

Figure 6.7: Results for the best case scenario for $N = 10$ and $M = 6$

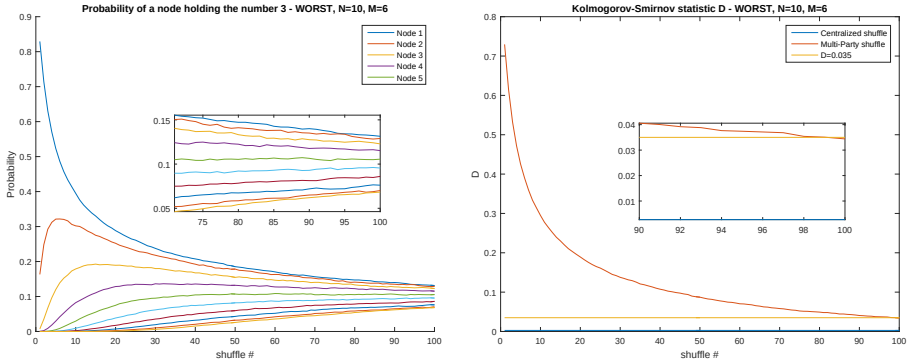
The same is true for the intermediate and worst case of the line topology as we can see in figures 6.8a and 6.9a, although, in this case it takes more than 40 shuffling rounds to reach the same ideal probability for each of the cases. Similarly to the best case, we notice that node 1 starts out with higher probability of holding the data with number 3, and then we notice a sharp increase on the probability of node 2 holding it since it is the only node that is connected to node 1 (recall that intermediate and worst case scenarios have the line topology of nodes).



(a) The probability of the number 3 being at different nodes at different number of shuffles (subplot is of a magnified region)

(b) Kolmogorov-Smirnov test at different number of shuffles(subplot is of a magnified region)

Figure 6.8: Results for the intermediate case scenario for $N = 10$ and $M = 6$



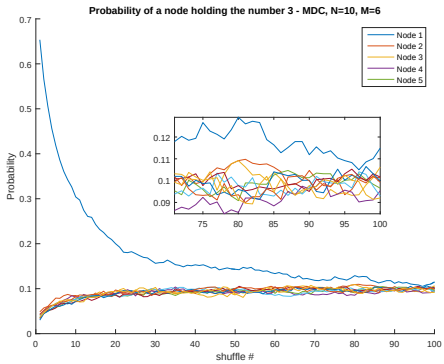
(a) The probability of the number 3 being at different nodes at different number of shuffles (subplot is of a magnified region)

(b) Kolmogorov-Smirnov test at different number of shuffles(subplot is of a magnified region)

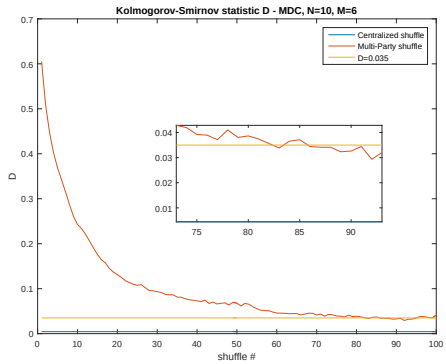
Figure 6.9: Results for the worst case scenario for $N = 10$ and $M = 6$

Results using MDC Dataset

In figures 6.10a and 6.11a we see the results of the MDC dataset simulations. For the MDC simulation with the random selection of users, although the connectivity resembles that of line topology, we cannot see it in figure 6.10a (like in figures 6.8a and 6.9a) because the users are not ideally ordered at each trial to reveal the same pattern as the artificial parameter simulation with line topology (recall that they were randomly selected).

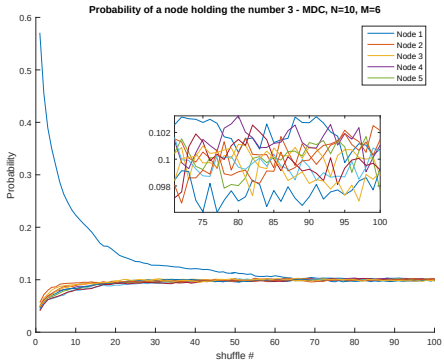


(a) The probability of the number 3 being at different nodes at different number of shuffles (subplot is of a magnified region)

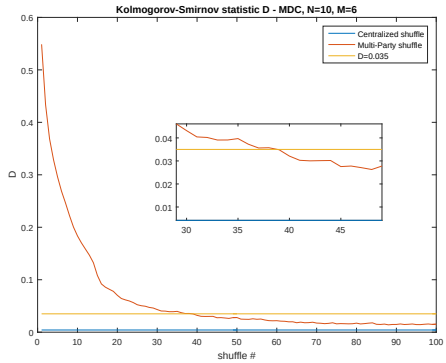


(b) Kolmogorov-Smirnov test at different number of shuffles(subplot is of a magnified region)

Figure 6.10: Results for the MDC data with random user selection for $N = 10$ and $M = 6$



(a) The probability of the number 3 being at different nodes at different number of shuffles (subplot is of a magnified region)



(b) Kolmogorov-Smirnov test at different number of shuffles(subplot is of a magnified region)

Figure 6.11: Results for the MDC data with clique user selection for $N = 10$ and $M = 6$

Kolmogorov-Smirnov test of the experiment

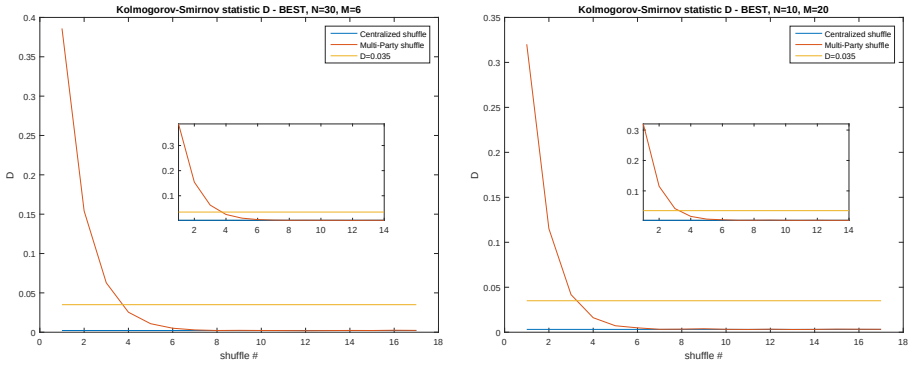
As we can see in figures 6.7b, 6.8b, 6.9b, 6.10b, and 6.11b, the error decreases as the number of shuffles increases. For our experimental setup, 4 shuffles is sufficient to adequately shuffle the data in the best case scenario while more than 60 shuffling rounds are needed in order to reach a similar distribution of the data for the worst case scenario. The MDC dataset simulation with random user selection is comparable to the worst case scenario while with clique based user selection it is better than the intermediate case but worse than the best case scenario.

Simulation type	**# of shuffles for** $D < 0.035$
Best case	4
Intermediate case	46
Worst case	100
MDC data random	85
MDC data cliques	40

Table 6.2: Summary of Kolmogorov-Smirnov test results for $N = 10$ $M = 6$.

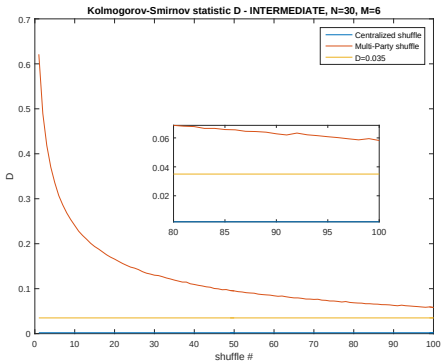
Effects of varying number of nodes and number of data items

For the fully connected topology (best case), varying the number of nodes or number of data items does not seem to have an effect on performance. For the line topology (intermediate and worst case), increasing the number of nodes also increases the number of shuffles necessary. However increasing the number of data items does not have a noticeable effect for those cases. These conclusions can be verified in the figures 6.12,6.13, and 6.14 which show the Kolmogorov-Smirnov statistic as function of the shuffles for different selection of total nodes N and total data items M.

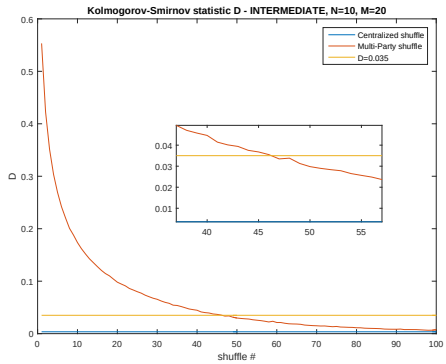


(a) $N = 30$, $M = 6$ (b) $N = 10$, $M = 20$

Figure 6.12: Kolmogorov-Smirnov test for different selection of N and M of the best case scenario

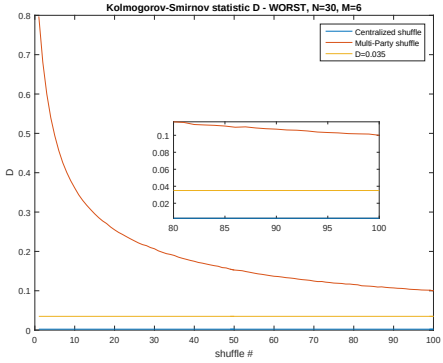


(a) $N = 30$, $M = 6$

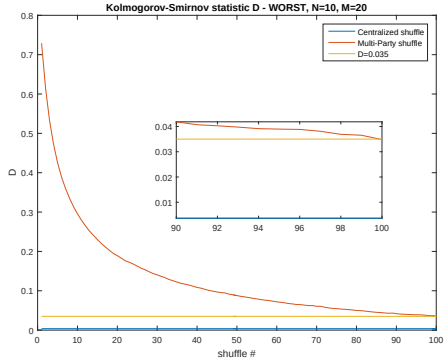


(b) $N = 10$, $M = 20$

Figure 6.13: Kolmogorov-Smirnov test for different selection of N and M of the intermediate case scenario



(a) $N = 30$, $M = 6$



(b) $N = 10$, $M = 20$

Figure 6.14: Kolmogorov-Smirnov test for different selection of N and M of the worst case scenario

6.4 Discussion

In this chapter, we evaluated a basic peer to peer opportunistic mix network in order to generate a baseline of results which can later be used to compare different strategies. We did this by opportunistically shuffling the data among the participants and showed that the number of shuffles is dependent on the properties of the graph that represents the participant interconnections. A fully connected topology requires only 4 shuffling rounds. On the other hand, a line topology required significantly more shuffling rounds; 46 rounds for the intermediate case and 100 shuffling rounds for the worst case. Using real user GPS traces from the MDC dataset we saw that the number of shuffling rounds did not exceed the worst case

when selecting random nodes from the population but at 85 rounds it was significantly higher than the intermediate case. Carefully selecting nodes from the population in the MDC dataset to form a more connected topology made a significant difference in the efficiency of the shuffling (only 40 shuffling rounds) and was significantly better than the intermediate case.

These results can be used to define the stopping criteria for a near uniform shuffle based on the topology of the nodes. A set of 10 nodes in a fully connected topology (having an average closeness centrality of 1) would require at least 4 shuffles. Whereas a set of 10 nodes in a line topology (having an average closeness centrality of 0.3430), would require 100 shuffles. For the MDC dataset the closeness centrality ranged from 0.3430 to 1 and from 0.6 to 1 for the random user selection and the clique-based user selection respectively.

Opportunistic peer to peer mixing, as part of a slicing and mixing strategy, can therefore reasonably mix the data so as to protect the identity of the source in the context of the data routing. However, the data content itself should be further obfuscated in order to protect the identity of the source which might be revealed from analyzing the data content. Such techniques require the manipulation of data entries and can reduce the quality of the data but it is often necessary to do so for the protection of the participants.

There are several technologies that can facilitate the exchange of data in a peer to peer manner. The most basic technology would be Bluetooth, however bandwidth can be a limiting factor with a maximum of $2Mbit/s$. If peers are typically within communication range around 60 seconds, this would mean only $60Mbits$ or a little over $5MBytes$ can be exchanged at a time. This number might be suitable for small data collection campaigns but might not be adequate for long duration campaigns that require data from multiple sensors. Ideally, the Wi-Fi transceiver of the mobile device can be used to directly link two nodes without the need for an access point or internet connection. The *Wi-Fi Direct* protocol enables just that and Apple's *Airdrop* performs a similar function. The bandwidth of Wi-Fi direct is $250Mbit/s$; that's more than 100 times faster than Bluetooth. Given a 60 second timeframe, this translates to $15000Mbits$ or almost $2GBytes$ of data that can be exchanged. That is equivalent to taking temperature readings every second for the next 9 years at least. Suffice it to say, it would work adequately for the majority of data collection campaigns.

Having seen a solution for the mechanism to exchange the data, the next technical hurdle is to detect when two nodes are within communication range. A straightforward method would be to use Wi-Fi scanning of nearby devices thereby using Wi-Fi to solve both the data exchange and the proximity detection. The main disadvantage of using Wi-Fi is that it would need to be on all the time and it can use a significant amount of energy compared to other solutions. Bluetooth Low Energy (LE) is ideal for this step due to its extremely low energy consumption. Bluetooth LE can be used to detect nearby devices and perhaps send some basic data to authenticate the users and their involvement in the data collection campaign and then Wi-Fi direct can be used to facilitate the main exchange of the data.

Peer trust is a topic that was not fully explored in this chapter. Since the nodes directly exchange their data between each other they need to trust that the other

nodes do not misuse, eavesdrop on, or otherwise manipulate the data. Fortunately there are many encryption strategies that can be employed to facilitate the exchange of data without the explicit requirement of trust. Furthermore, since the software on the sensor devices would likely automatically handle the data collection and peer to peer data exchange we can confidently assume that the data is not easily accessible to the nodes. Even so, given a sufficiently large number of nodes in the scenario, if a node is able to access the data that is exchanged, they have no way to confidently determine which node collected it in the first place. For these reasons, the simulated experiments in this chapter ignored trust. Given a less than ideal number of nodes trust could be a factor in the early rounds of shuffling (when nodes hold a higher percentage of their own data). Lets recall that the amount of data that each node exchanges upon each interaction can depend on some privacy or trust metrics. Intuitively, we can guess that as the nodes are mixing the data, thus each one holding data from an increasing number of different nodes, trust will start to increase among the nodes since each data exchange would mean sharing less of their own data. Having mentioned this, it would be interesting to see how much an untrustworthy environment would affect the number of data exchanged between the nodes and how much longer it would take to fully mix a set of data.

Chapter 7

Discussion and Future Work

There are many aspects of data privacy that need to be addressed and in this thesis we have addressed only a subset of them from a technical perspective. We investigated the uses of all the data that can be collected from the physical sensors of a smartphone and evaluated their potential to be a privacy threat. We proposed self-sufficient methods to detect the location context of a user and other location tracking information which do not need to share any data from the device. Finally, we evaluated the real-world efficacy of implementing an opportunistic mix network for the data reporting phase of a generic crowd-sensing scenario. Other aspects include the user behaviour towards data privacy, privacy policies and their ability to inform the average user, the implications of the GDPR, the commercial impact of privacy, the many other services that could potentially be implemented in a self-sufficient manner, blockchain solutions, specifics on secure computation, and many more.

If we can fully address privacy from a technical level (for example, with secure or private algorithms), we can enable developers and creators to make secure services without worrying too much about the continuously evolving policy landscape and at the same time increase consumer confidence. By seeking the answers to our research questions we have come a step closer to this goal. In the following sections we will summarize the results from this thesis and provide answers to the research questions that we posed. We will also propose some future work that can be done in the context of this thesis to either expand our answers to the research questions or find answers to an expanded version of our research questions.

7.1 Results Overview

Chapter 3 result summary. In chapter 3 we saw that most hardware sensor data on a mobile device have one or multiple privacy threats. Location can be determined using most of the sensors available (GPS, Cell ID, WiFi, Bluetooth, Accelerometer, Magnetometer(indoor only), Barometer(floor level only)), while other sensors can be used to authenticate/identify a person based on their behaviour (accelerometer, gyroscope, camera, microphone). MEMS based sensors are all

vulnerable to fingerprinting, that is, each individual sensor has a unique bias in their output which can be used to identify a specific device.

Chapter 4 result summary. In chapter 4 we describe an algorithm for relative localization. Our algorithm is able to use multiple range measurements from different locations to stationary beacons whose locations are not known and then determine the locations of those beacons and the locations where the measurements were taken from and generate a relative map. Our algorithm is shown to have improved accuracy under the same conditions compared to MDS. If accurate range measurements of Cell IDs can be procured by a smartphone (based on RSS for example), our algorithm can be used for location self-provisioning.

Chapter 5 result summary. In chapter 5 we introduce some algorithms to determine location context based solely on Cell ID data. We show that by using Cell Oscillation events and graph based clustering methods we can detect significant places. We also show that we can obtain distance measurements between Cell ID observations and create a distance matrix to which we can apply density based clustering like DBSCAN to determine significant places. This chapter among other work mentioned in the related work chapter 2.1 conclusively proves that location context can be determined on the device itself.

Chapter 6 result summary. In chapter 6 we evaluate the efficacy of opportunistic data mixing using real mobility data. Our results show that depending on the time constraints of a particular data collection study, the mixing of the data can be affected. According to our analysis, at least 40 days should be devoted in opportunistically mixing the data if the participants are carefully selected.

7.2 Answers to Research Questions

After investigating the various uses of smartphone sensor data, developing algorithms for detecting location context and for estimating other location tracking information, and analyzing the efficacy of an opportunistic mix network for data reporting using real mobility data, we are able to answer our research questions.

7.2.1 Research Question 1

Q: *Considering all the different sensor data that can be collected on a mobile ubiquitous device, both data collectors and participants must be made aware of what privacy threats come with sharing this data. For that, we must answer the question:* **Which sensor data originating from a mobile ubiquitous device has the potential to uniquely identify a person or to otherwise reveal sensitive personal information?**

A: In chapter 3 we collected evidence that the following sensors could compromise a users privacy:

- GPS. Reveals location of personal places and is vulnerable to inference attacks.
- Cell ID. Reveals location of personal places and is vulnerable to inference attacks.
- WiFi. Reveals location of personal places and is vulnerable to inference attacks.
- Bluetooth. Reveals location of personal places and is vulnerable to inference attacks.
- Touchscreen. Reveals keystroke dynamics and can be used to reveal one's PIN.
- Microphone. Can be used for speaker recognition and for AEI. It can also be used to identify keystrokes on mechanical keyboards.
- Camera. Can be used for face recognition, gait recognition, location recognition, and even authorship recognition.
- Accelerometer. MEMS sensor fingerprinting can identify individual devices, location/navigation, activity recognition, gait recognition, speaker recognition.
- Gyroscope. MEMS sensor fingerprinting can identify individual devices, can be used for speaker recognition.
- Magnetometer. MEMS sensor fingerprinting can identify individual devices, can reveal location from fingerprinting.
- Barometer. MEMS sensor fingerprinting can identify individual devices, can be used for building floor detection.
- Ambient light sensor. Can reveal elements of location (indoor/outdoor, room), Can reveal the PIN.

At first glance this may seem disappointing in terms of safeguarding privacy but we must keep in mind that several of the experiments that were performed were done in a highly controlled manner and current privacy preserving techniques can mitigate some of the threats. Cell IDs, WiFi APs, and Bluetooth APs can all be encrypted so as to mask their real identifiers and still be useful in detecting personal places while keeping their actual geographic location hidden. Nevertheless, most services that use this data do not obfuscate it because it is easier to use geographic location compared to using abstract location identifiers. The touchscreen data for the location of touches in most smartphones is only available to the application on the foreground and therefore a malicious application cannot eavesdrop unless it is opened on the foreground. This would pose a significant threat only if an application has its own virtual keyboard. The microphone and camera both require the explicit permission of the user to be accessed by applications and this means that at least the user is aware that a particular application

has such access. This does not imply that an application will only access these resources at the user's request. Once an application is granted access, it has the ability eavesdrop at any moment. The accelerometer, barometer, magnetometer, and ambient light sensors are completely open and free to be accessed without any special permissions but machine learning methods can be used to reveal a variety of information. The average user would not know that an application is accessing these sensors and there is no trivial way to block an application from accessing them which could put them at risk. This list is not complete since we have only looked at the physical sensors of the devices and there are many more software-related sources that can "leak" information.

7.2.2 Research Question 2

Q: *With the current trends on mobile ubiquitous device processing power and storage, the option to migrate tasks from the cloud to the edge to process data and derive useful and actionable information can be seriously considered.* **Is it feasible to do data mining and provide basic services, like localization, to users without transmitting sensitive data to a cloud service from the mobile device or otherwise rely on a third party?**

A: Many demanding machine learning and data mining tasks can be somewhat securely implemented using obfuscation techniques that were discussed in the related work in chapter 2.3. Tasks that do not require a significant number of operations can be securely computed using the secure computation methods that were discussed in the related work in chapter 2.6. Less demanding tasks that do not require data from multiple sources can already be performed on modern devices which have ample processing power and memory. In the future, as devices get more powerful, more demanding tasks can be performed on the devices themselves. Furthermore, developments in secure computation and in server technology can help in the future to migrate more demanding tasks to a secure computation scheme. For location and location context awareness there exist methods that are computationally efficient enough to be performed on the device itself and were discussed in the related work (chapter 2.1). We also presented two of our own localization and location context algorithms in chapters 4 and 5 respectively. On the other hand, privacy for exploratory studies which often require the data to be anonymized but as close to its original form as possible are only feasible if the collected data can satisfy privacy measures (k-anonymity, l-diversity, etc.) without significant obfuscation. In general, this is not a realistic assumption and would require that the participants give their full consent after understanding the implications of the data being collected and trusting the entity which is collecting the data.

In summary, although not all services can be implemented without relying on a third party, some can. Location context can be easily detected using data readily available on the mobile device using relatively lightweight algorithms. Other location features like distance traveled, speed, and even mode of transport can be estimated using the on-board sensors of the device but accuracy and drift errors should be taken into account. As such, they may not be suitable for applications

that require high accuracy. Services that require significant use of resources, processing or storage, like speech recognition are still better served as a cloud service. Data mining for population studies, by nature, cannot be reliably performed without individuals somewhat or wholly relinquishing their privacy. However, once models exist for detecting or deriving some feature, individuals can safely use them on their own devices if the model and method are resource-friendly. As the smartphone gets more powerful in terms of resources, more and more complicated models can be used on the device.

7.2.3 Research Question 3

Q: *Opportunistic crowd-sensing leverages the mobility of participants in order to opportunistically collect data from the environment. Some such crowd-sensing schemes break from the opportunistic paradigm when it comes to privacy measures during data reporting. Other schemes utilize the paradigm throughout the entire crowd-sensing framework, which includes the privacy-related tasks of data reporting. Even so, they fail to prove that such an opportunistic scheme can work in a real environment with real mobility data.* **Can fully opportunistic crowd-sensing still be carried out for scientific research without compromising the privacy of individual participants?**

A: Using a combination of secure offloaded computation techniques, on-device computation (considering the capabilities of modern devices), and efficient data mixing techniques seen in the related work and evaluated with real mobility data in chapter 6, we can conclude that yes, fully opportunistic crowd-sensing can be carried out both for data collection and for data reporting. However, caution should be practiced with some types of data by either obfuscating it by using one of the techniques in the related work (chapter 2.3), or by processing it on the device itself before and sending only the result which again may be a privacy threat itself. Determining which data is safe to disseminate should be dealt with in a case by case basis and chapter 3 can serve as a guideline for some of the data types.

The main issue with the opportunistic mix network strategy is the time limitations of the crowd-sensing campaign. The opportunistic mix network may take several weeks to sufficiently mix the data unless the campaign carefully selects participants which are highly connected with each other. Another time-saving adjustment to the mixing phase is to only mix data within small groups of participants, let's say groups of size k, instead of the entire population. This would introduce a certain k-anonymity about the source of a particular batch of data and significantly shorten the time required to mix the data of the entire population of the campaign.

7.3 Implications for GDPR Compliance

The GDPR applies to any entity that handles, uses, or collects "personal data". In the GDPR, personal data is loosely defined as "*any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference*

to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person". This definition is general enough to cover a wide range of data but for non-experts it is not immediately obvious which data is **actually** covered.

Each EU member state is responsible for enforcing the GDPR rules by appointing a supervisory authority (SA) who works with other member state SAs to keep consistency among them. The European Data Protection Board coordinates the SAs. Individuals can submit GDPR claims to the relevant SA who will evaluate the claim and proceed with the appropriate actions. The SA also provides some basic guidance to businesses and organizations in order to help them comply with the GDPR, for example, in the form of a self-assessment checklist. Still, no specific definition is given for what constitutes personal data.

Regarding this issue, chapter 3 of this thesis can bring some clarity for smartphone sensor data, a rather small subset of all the possible data types out there. In that chapter we saw that most of the sensor data on a smartphone can reveal personal information but we can get a sense of how sensitive they are by using the sampling types that we defined (see table 3.1) and assigned to each data type. Data that require a sampling type *A* being very privacy-sensitive and data that require a sampling type *F* being less privacy sensitive. The main implication of the sampling type categorization is that if an organization collects a certain data type at a lesser sampling type than is required to derive sensitive information (see table 3.2), then they may not have to consider this data as personal.

The GDPR has provisions for certification and certification bodies (articles 42 and 43) and a list of these can be found on the European Data Protection Board website [200]. At the time of writing of this thesis there is no official GDPR certification mechanism but we can expect that there will be in the future. A GDPR certification (along with their seals and marks) can let individuals know about the GDPR compliance of an organization and reassure them that their data is handled accordingly and that they are afforded a certain control over their data. However, such a certification would not inform an individual about the level of potential privacy loss they might incur in the case of a data breach against the organization for example.

We can use the results of chapter 3 to aid in evaluating the potential privacy loss of a set of smartphone sensor data. One way to go about this is to assign numerical values to the threat posed by some type of personal information and then combine it with the sampling type required to derive this personal information. For example, lets suppose that a smartphone application only collects Cell ID data. Location and personal places can be derived from Cell ID data, so we will assume that it is a high privacy threat. The sampling type required to derive this information is *C*. Combining the two measures should result in a relatively high potential privacy loss. On the other hand, lets consider barometer data. This data, sampled at the sampling type *B* level, can be used to reveal the altitude or floor level of an individual which seems like a low privacy threat. This should result in a low overall potential privacy loss. The specifics of this evaluation can be further investigated in the future and a concrete methodology can be established for all data types, not just smartphone sensor data, and organizations can display this

score in an effort to inform the users about the level of privacy loss that a user might incur in a worst case scenario.

7.4 Future Work

Despite our efforts, there is still more that needs to be done to fully realize the implications of opportunistic crowd sensing and service self provisioning on privacy.

In order to have a complete summary of privacy threats from all mobile device data types, more data sources need to be investigated. In this thesis we only looked at sensor data, but there are many software sources that need to be scrutinized. Some of these include application usage, phone interaction (for example, screen on/off), browsing history, instant messaging behaviour, and more.

An important subject that was not covered in this thesis was that of database record linking. Essentially two databases with a few congruent fields can be merged into one database that might reveal sensitive information about an individual. The simplest example is a database with fields such as *name*, *age*, *marital status*, *education level*,*area code* and another with fields *age*, *education level*, *area code*, *telephone number*. One can try to match the congruent fields of these two databases to narrow down the entries that could be the telephone number of someone whose name they already know. The threats of database record linking can be devastating to an individual as it could result in harassment or worse. Many database management systems come with standard features that allow the linking of two or more databases into one master database using rules and there is also research into sophisticated probabilistic and machine learning methods to do the task when the data is not easily matched. At the same time, it is very time consuming to create rules or to verify the accuracy of automated methods for large databases. For now, there is not a perfect solution for record linkage and when it comes to big data the time to verify the linkage can make it unapproachable. However, relatively small databases can be realistically vulnerable to this type of exploitation. Even if precautions are taken by organizations to individually secure their databases they must pay close attention to who has access rights in order to avoid this kind of exploitation when it is not desired. Furthermore, linkable databases managed by different organizations can also be exploited in the same manner. Protecting personal data at this level requires inter-organization cooperation and can be difficult to manage. As interesting as this topic is, it is not in the scope of this thesis and would be a fruitful subject for future work in the field of privacy.

Localization techniques that do not rely on third parties need to be improved before they can be adopted in ubiquitous devices such as smartphones. Our localization algorithm in chapter 4, although minimal in requirements, can be computationally complex and still not accurate enough for wide adoption. Improvements in dead-reckoning-based techniques where on-board sensors are used is a good start but the research is saturated with algorithms that still cannot be reliably used by the average smartphone user. As sensors improve and as smartphone capabilities improve, a robust solution, which could appeal to the average user, is just over the horizon.

Location context on the other hand can be determined fairly reliably with only minimal information on the device itself. Applications or services that require such input still want to access the raw location of a user. We need to push for more granular control over what data these application should get. Why should they know the geographical location of your home and not have the device automatically tell them that the current location is indeed ”home” without revealing any geographical information? Changes need to be made in the core interaction between applications and the operating system which provides them with the necessary data.

Location tracking information such as distance and speed should also be treated similarly. There are two possible paths to a solution. One is that the Google location service, for Android devices, or the equivalent location service for other family of devices, provide an API to access location context and location tracking information separately from GPS coordinates. Internally they can use GPS coordinates to easily detect and calculate this information and expose it as a separate service to developers. This requires that the location service is secure and trustworthy. The other path is to build single purpose services that do not rely on GPS coordinates and therefore would not directly pose a threat to user privacy.

In this thesis we showed that location context can be detected without the need for GPS coordinates. We showed that location tracking information could theoretically be measured without the use of GPS as well. Therefore, we can conclude that it is possible to provide location context and location tracking information separately from GPS location.

It would be interesting to develop a framework that accommodates such granularity in the dissemination of sensor information, not limited to location, to third party applications. We can imagine that there would be closed modules that have direct access to the raw data, and then would output processed information such as location context, activity, etc. or obfuscate the raw data ever so intelligently so as to both protect the user and not significantly reduce the utility of the data. In only extreme cases would an application require the use of the raw data and the user would be warned of such a requirement and the possible threats which result.

On the side of data reporting, more studies need to be made in different types of environments. The data used to do the analysis in this thesis was from only one city. Data should be gathered from environments with more or with less population density to determine the cutoff for the feasibility of opportunistic data mixing. Furthermore, recruiting strategies should be developed such that the participants are both representative of what is being studied but also well connected in terms of their mobility.

Bibliography

[1] J. De Vriendt, P. Laine, C. Lerouge, and Xiaofeng Xu, “Mobile network evolution: a revolution on the move,” *IEEE Communications Magazine*, vol. 40, no. 4, pp. 104–111, apr 2002. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=995858http://ieeexplore.ieee.org/document/995858/

[2] S. Dekleva, J. Shim, U. Varshney, and G. Knoerzer, “Evolution and emerging issues in mobile wireless networks,” *Communications of the ACM*, vol. 50, no. 6, pp. 38–43, jun 2007. [Online]. Available: http://www.cs.colorado.edu/{~}rhan/CSCI{_}7143{_}002{_}Fall{_}2001/Papers/4G{_}p38-dekleva.pdfhttp://portal.acm.org/citation.cfm?doid=1247001.1247003

[3] B. N. Schilit, “Mobile Computing: Looking to the Future,” *Computer*, vol. 44, no. 5, pp. 28–29, may 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5767725/

[4] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. a. Peterson, “People-centric urban sensing,” in *Proceedings of the 2nd annual international workshop on Wireless internet - WICON ’06*. New York, New York, USA: ACM Press, 2006, pp. 18–es. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1234161.1234179

[5] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, “Participatory Sensing,” *World Sensor Web Workshop, ACM Sensys*, 2006. [Online]. Available: https://cloudfront.escholarship.org/dist/prd/content/qt19h777qd/qt19h777qd.pdf

[6] D. Estrin, “Participatory sensing: applications and architecture [Internet Predictions],” *IEEE Internet Computing*, vol. 14, no. 1, pp. 12–42, jan 2010. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.122.3024http://ieeexplore.ieee.org/document/5370818/http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5370818

[7] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn, “The Rise of People-Centric Sensing,” *IEEE Internet Computing*, vol. 12, no. 4, pp. 12–21, jul 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4557974/

[8] Common Criteria, “Common Criteria for Information Technology Security Evaluation Part 2 : Security functional components,” *Security*, vol. V3.1 R5, no. September, pp. 1–323, 2017. [Online]. Available: http://oai.dtic.mil/oai/oai?verb=getRecord{\&}metadataPrefix=html{\&}identifier=ADA406677

[9] A. Pfitzmann and M. Hansen, “A terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management,” *Technical University Dresden*, pp. 1–98, 2010. [Online]. Available: http://dud.inf.tu-dresden.de/Anon{_}Terminology.shtml{\%}5Cnhttp://dud.inf.tu-dresden.de/literatur/Anon{_}Terminology{_}v0.34.pdf

[10] A. Yanes, “Privacy and Anonymity,” pp. 1–7, jul 2014. [Online]. Available: http://arxiv.org/abs/1407.0423

[11] BBC, “Edward Snowden: Leaks that exposed US spy programme,” 2014. [Online]. Available: https://www.bbc.com/news/world-us-canada-23123964 (Accessed: June 13, 2018).

[12] Z. Kleinman, “Cambridge Analytica: The story so far,” 2018. [Online]. Available: https://www.bbc.com/news/technology-43465968 (Accessed: June 6, 2018).

[13] L. Sly, “U.S. soldiers are revealing sensitive and dangerous information by jogging,” 2018. [Online]. Available: http://wapo.st/2nlhXed?tid=ss{\%}5C{_}mail{\%}5C{\&}utm{\%}5C{_}term=.13232a8a8b0e (Accessed: May 23, 2018).

[14] S. B. Wicker, “The loss of location privacy in the cellular age,” *Communications of the ACM*, vol. 55, no. 8, p. 60, aug 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2240236.2240255

[15] L. Kugler, “The war over the value of personal data,” *Communications of the ACM*, vol. 61, no. 2, pp. 17–19, jan 2018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3181977.3171580

[16] E. Steel, C. Locke, E. Cadman, and B. Freese, “How much is your personal data worth?” 2017. [Online]. Available: https://ig.ft.com/how-much-is-your-personal-data-worth/ (Accessed: May 15, 2018).

[17] C. Li, D. Y. Li, G. Miklau, and D. Suciu, “A theory of pricing private data,” in *Proceedings of the 16th International Conference on Database Theory - ICDT ’13*, vol. 39, no. 4. New York, New York, USA: ACM Press, 2013, p. 33. [Online]. Available: http://arxiv.org/abs/1208.5258{\%}0Ahttp://dx.doi.org/10.1145/2448496.2448502http://dl.acm.org/citation.cfm?doid=2448496.2448502

[18] V. Gkatzelis, C. Aperjis, and B. A. Huberman, “Pricing private data,” *Electronic Markets*, vol. 25, no. 2, pp. 109–123, jun 2015. [Online]. Available: http://link.springer.com/10.1007/s12525-015-0188-8

[19] A. Acquisti, “Nudging Privacy: The Behavioral Economics of Personal Information,” *IEEE Security & Privacy Magazine*, vol. 7, no. 6, pp. 82–85, nov 2009. [Online]. Available: http://ieeexplore.ieee.org/document/5370707/

[20] A. R. Beresford, D. Kübler, and S. Preibusch, “Unwillingness to pay for privacy: A field experiment,” *Economics Letters*, vol. 117, no. 1, pp. 25–27, oct 2012. [Online]. Available: http://dx.doi.org/10.1016/j.econlet.2012.04.077http://linkinghub.elsevier.com/retrieve/pii/S0165176512002182

[21] A. Acquisti, L. K. John, and G. Loewenstein, “What Is Privacy Worth?” *The Journal of Legal Studies*, vol. 42, no. 2, pp. 249–274, 2013. [Online]. Available: http://www.journals.uchicago.edu/doi/10.1086/671754

[22] M. Gustarini, K. Wac, and A. K. Dey, “Anonymous smartphone data collection: factors influencing the users’ acceptance in mobile crowd sensing,” *Personal and Ubiquitous Computing*, vol. 20, no. 1, pp. 65–82, 2016.

[23] S. Barth and M. D. de Jong, “The privacy paradox – Investigating discrepancies between expressed privacy concerns and actual online behavior – A systematic literature review,” *Telematics and Informatics*, vol. 34, no. 7, pp. 1038–1058, nov 2017. [Online]. Available: https://doi.org/10.1016/j.tele.2017.04.013http://linkinghub.elsevier.com/retrieve/pii/S0736585317302022

[24] C. Hallam and G. Zanella, “Online self-disclosure: The privacy paradox explained as a temporally discounted balance between concerns and rewards,” *Computers in Human Behavior*, vol. 68, pp. 217–227, mar 2017. [Online]. Available: http://dx.doi.org/10.1016/j.chb.2016.11.033http://linkinghub.elsevier.com/retrieve/pii/S0747563216307749

[25] M. Fanourakis and K. Wac, “Lightweight Clustering of Cell IDs into Meaningful Neighbourhoods,” in *Performance & Security Modelling and Evaluation of Cooperative Heterogeneous Networks - HET-NETs*, D. D. Kouvatsos, S. Balsamo, and Y. Takahashi, Eds. River Publishers, 2013, pp. 698–704. [Online]. Available: http://riverpublishers.com/book{_}details.php?book{_}id=234

[26] ——, “ReNLoc: An anchor-free localization algorithm for indirect ranging,” in *2015 IEEE 16th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, jun 2015, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7158145

[27] K. Wac, M. Gustarini, J. Marchanoff, M. Fanourakis, C. Tsiourti, M. Ciman, J. Hausmann, and G. Pinar, “Mqol: experiences of the ’mobile communications and computing for quality of life’ living lab,” in *2015 17th International Conference on E-health Networking, Application & Services (HealthCom)*, no. i. IEEE, oct 2015, pp. 177–181. [Online]. Available: http://ieeexplore.ieee.org/document/7454494/

[28] M. Gustarini, M. P. Scipioni, M. Fanourakis, and K. Wac, "Differences in smartphone usage: Validating, evaluating, and predicting mobile user intimacy," *Pervasive and Mobile Computing*, vol. 33, pp. 50–72, dec 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1574119216300682{\%}5Cnhttp://linkinghub.elsevier.com/retrieve/pii/S1574119216300682http://linkinghub.elsevier.com/retrieve/pii/S1574119216300682

[29] K. Siła-Nowicka, J. Vandrol, T. Oshan, J. A. Long, U. Demšar, and A. S. Fotheringham, "Analysis of human mobility patterns from GPS trajectories and contextual information," *International Journal of Geographical Information Science*, vol. 30, no. 5, pp. 881–906, 2016. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/13658816.2015.1100731http://www.tandfonline.com/doi/full/10.1080/13658816.2015.1100731

[30] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place Lab: Device Positioning Using Radio Beacons in the Wild," in *Pervasive Computing*. Springer, 2005, pp. 116–133. [Online]. Available: http://link.springer.com/10.1007/11428572{_}8

[31] M. Ibrahim and M. Youssef, "CellSense: A Probabilistic RSSI-Based GSM Positioning System," in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*. IEEE, dec 2010, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5683779http://ieeexplore.ieee.org/document/5683779/

[32] ——, "CellSense: An Accurate Energy-Efficient GSM Positioning System," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 286–296, jan 2012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6062428http://ieeexplore.ieee.org/document/6062428/

[33] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello, "Extracting places from traces of locations," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 9, no. 3, p. 58, jul 2005. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1094549.1094558

[34] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava, "SensLoc: sensing everyday places and paths using less energy," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems - SenSys '10*. New York, New York, USA: ACM Press, 2010, p. 43. [Online]. Available: http://dl.acm.org/citation.cfm?id=1869989http://portal.acm.org/citation.cfm?doid=1869983.1869989

[35] Z. Chen, S. Wang, Y. Chen, Z. Zhao, and M. Lin, "InferLoc: Calibration Free Based Location Inference for Temporal and Spatial Fine-Granularity Magnitude," in *2012 IEEE 15th International Conference on Computational Science and Engineering*. IEEE, dec 2012, pp. 453–460. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6417328

[36] Q. D. Vo and P. De, “A Survey of Fingerprint-Based Outdoor Localization,” *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 491–506, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7131436/

[37] K. Yadav, V. Naik, A. Kumar, and P. Jassal, “PlaceMap: Discovering Human Places of Interest Using Low-Energy Location Interfaces on Mobile Phones,” in *Proceedings of the Fifth ACM Symposium on Computing for Development - ACM DEV-5 ’14*, vol. 5. New York, New York, USA: ACM Press, 2014, pp. 93–102. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2674377.2674386

[38] Yufeng Wang, Yuguan Lian, and A. Nakao, “LHDV-HOP: An energy-effective range-free localization scheme in wireless sensor networks,” in *2010 IEEE 12th International Conference on Communication Technology*. IEEE, nov 2010, pp. 1007–1010. [Online]. Available: http://ieeexplore.ieee.org/document/5688808/

[39] T. He, C. Huang, B. M. Blum, J. a. Stankovic, and T. Abdelzaher, “Range-free localization schemes for large scale sensor networks,” in *Proceedings of the 9th annual international conference on Mobile computing and networking - MobiCom ’03*. New York, New York, USA: ACM Press, 2003, p. 81. [Online]. Available: http://portal.acm.org/citation.cfm?doid=938985.938995

[40] V. Ramadurai and M. Sichitiu, “Localization in Wireless Sensor Networks: A Probabilistic Approach.” in *International conference on wireless networks*, 2003, pp. 275–281.

[41] T. Srinath, “Localization in resource constrained sensor networks using a mobile beacon with in-ranging,” in *2006 IFIP International Conference on Wireless and Optical Communications Networks*. IEEE, 2006, pp. 5 pp.–5. [Online]. Available: http://ieeexplore.ieee.org/document/1666591/

[42] L. Doherty, K. Pister, and L. El Ghaoui, “Convex position estimation in wireless sensor networks,” in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 3. Anchorage, AK: IEEE, 2001, pp. 1655–1663. [Online]. Available: http://ieeexplore.ieee.org/document/916662/

[43] M. W. Carter, H. H. Jin, M. A. Saunders, and Y. Ye, “SpaseLoc: An Adaptive Subproblem Algorithm for Scalable Wireless Sensor Network Localization,” *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 1102–1128, jan 2007. [Online]. Available: http://epubs.siam.org/doi/10.1137/040621600

[44] Y. Shang, W. Ruml, Y. Zhang, and M. P. J. Fromherz, “Localization from mere connectivity,” in *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing - MobiHoc ’03*. New York, New York, USA: ACM Press, 2003, p. 201. [Online]. Available: http://portal.acm.org/citation.cfm?doid=778415.778439

[45] Yi Shang and W. Ruml, “Improved MDS-based localization,” in *IEEE INFOCOM 2004*, vol. 4. IEEE, 2004, pp. 2640–2651. [Online]. Available: http://ieeexplore.ieee.org/document/1354683/

[46] A. Ahmed, Hongchi Shi, and Yi Shang, “SHARP: A New Approach to Relative Localization in Wireless Sensor Networks,” in *25th IEEE International Conference on Distributed Computing Systems Workshops*. IEEE, 2005, pp. 892–898. [Online]. Available: http://ieeexplore.ieee.org/document/1437278/

[47] K.-Y. Cheng, K.-S. Lui, and V. Tam, “Localization in Sensor Networks with Limited Number of Anchors and Clustered Placement,” in *2007 IEEE Wireless Communications and Networking Conference*. Kowloon: IEEE, 2007, pp. 4425–4429. [Online]. Available: http://ieeexplore.ieee.org/document/4225051/

[48] Lei Fang, Wenliang Du, and Peng Ning, “A beacon-less location discovery scheme for wireless sensor networks,” in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, vol. 1. IEEE, 2005, pp. 161–171. [Online]. Available: http://ieeexplore.ieee.org/document/1497888/

[49] S. Velagapalli and H. Fu, “Beacon-less Location Detection in Wireless Sensor Networks for Non-flat Terrain,” in *Future Generation Communication and Networking (FGCN 2007)*, vol. 1. IEEE, 2007, pp. 528–534. [Online]. Available: http://ieeexplore.ieee.org/document/4426177/

[50] M. Jin, S. Xia, H. Wu, and X. Gu, “Scalable and fully distributed localization with mere connectivity,” in *2011 Proceedings IEEE INFOCOM*. IEEE, apr 2011, pp. 3164–3172. [Online]. Available: http://ieeexplore.ieee.org/document/5935163/

[51] C.-Y. Wen and Y.-C. Hsiao, “Decentralized anchor-free localization for wireless ad-hoc sensor networks,” in *2008 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, oct 2008, pp. 2777–2785. [Online]. Available: http://ieeexplore.ieee.org/document/4811717/

[52] L. Sweeney, “k-anonymity: A model for protecting privacy,” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, oct 2002. [Online]. Available: http://www.worldscientific.com/doi/abs/10.1142/S0218488502001648

[53] M. E. Nergiz and C. Clifton, “Thoughts on k-anonymization,” *Data & Knowledge Engineering*, vol. 63, no. 3, pp. 622–645, dec 2007. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0169023X07000468

[54] M. Nergiz, C. Clifton, and A. Nergiz, “Multirelational k-Anonymity,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 8, pp. 1104–1117, aug 2009. [Online]. Available: http://ieeexplore.ieee.org/document/4653492/

[55] J. Domingo-Ferrer and V. Torra, “A Critique of k-Anonymity and Some of Its Enhancements,” in *2008 Third International Conference on Availability, Reliability and Security*. IEEE, mar 2008, pp. 990–993. [Online]. Available: http://ieeexplore.ieee.org/document/4529451/

[56] T. Truta and B. Vinay, “Privacy Protection: p-Sensitive k-Anonymity Property,” in *22nd International Conference on Data Engineering Workshops (ICDEW'06)*. IEEE, 2006, pp. 94–94. [Online]. Available: http://ieeexplore.ieee.org/document/1623889/

[57] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, “L-diversity: privacy beyond k-anonymity,” in *22nd International Conference on Data Engineering (ICDE'06)*, vol. 2006. IEEE, 2006, pp. 24–24. [Online]. Available: http://ieeexplore.ieee.org/document/1617392/

[58] N. Li, T. Li, and S. Venkatasubramanian, “t-Closeness: Privacy Beyond k-Anonymity and l-Diversity,” in *2007 IEEE 23rd International Conference on Data Engineering*, no. 3. IEEE, apr 2007, pp. 106–115. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4221659http://ieeexplore.ieee.org/document/4221659/

[59] Ninghui Li, Tiancheng Li, and S. Venkatasubramanian, “Closeness: A New Privacy Measure for Data Publishing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 7, pp. 943–956, jul 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5072216/

[60] D. Rebollo-Monedero, J. Forne, and J. Domingo-Ferrer, “From t-Closeness-Like Privacy to Postrandomization via Information Theory,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 11, pp. 1623–1636, nov 2010. [Online]. Available: http://ieeexplore.ieee.org/document/5288525/

[61] X. Xiao and Y. Tao, “M-invariance: towards privacy preserving re-publication of dynamic datasets,” in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data - SIGMOD '07*. New York, New York, USA: ACM Press, 2007, p. 689. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1247480.1247556

[62] K. L. Huang, S. S. Kanhere, and W. Hu, “Preserving privacy in participatory sensing systems,” *Computer Communications*, vol. 33, no. 11, pp. 1266–1280, jul 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comcom.2009.08.012http://linkinghub.elsevier.com/retrieve/pii/S0140366409002448

[63] T. Giannetsos, S. Gisdakis, and P. Papadimitratos, “Trustworthy People-Centric Sensing: Privacy, security and user incentives road-map,” in *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*. IEEE, jun 2014, pp. 39–46. [Online]. Available: http://ieeexplore.ieee.org/document/6849103/

[64] L. Becchetti, L. Filipponi, and A. Vitaletti, “Opportunistic privacy preserving monitoring,” *Proceedings of PhoneSense 2010*, pp. 51–55, 2010.

[65] D. He, S. Chan, and M. Guizani, "Privacy and incentive mechanisms in people-centric sensing networks," *IEEE Communications Magazine*, vol. 53, no. 10, pp. 200–206, oct 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7295484/

[66] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1928–1946, nov 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0164121211001701

[67] K. Liu, H. Kargupta, and J. Ryan, "Random projection-based multiplicative data perturbation for privacy preserving distributed data mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 1, pp. 92–106, 2006.

[68] J. Wang, Y. Luo, Y. Zhao, and J. Le, "A Survey on Privacy Preserving Data Mining," in *2009 First International Workshop on Database Technology and Applications*, no. April 2013. IEEE, apr 2009, pp. 111–114. [Online]. Available: http://ieeexplore.ieee.org/document/5207803/

[69] K. Chen and L. Liu, "Geometric data perturbation for privacy preserving outsourced data mining," *Knowledge and Information Systems*, vol. 29, no. 3, pp. 657–695, 2011.

[70] N. Patel and S. Patel, "A Study on Data Perturbation Techniques in Privacy Preserving Data Mining," *International Research Journal of Engineering and Technology*, vol. 2, pp. 2120–2124, 2015.

[71] H. Choi, S. Chakraborty, Z. M. Charbiwala, and M. B. Srivastava, "SensorSafe: A framework for privacy-preserving management of personal sensory information," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6933 LNCS, pp. 85–100, 2011.

[72] H. Choi, S. Chakraborty, and M. B. Srivastava, "Design and Evaluation of SensorSafe: A Framework for Achieving Behavioral Privacy in Sharing Personal Sensory Information," in *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*. IEEE, jun 2012, pp. 1004–1011. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6296083http://ieeexplore.ieee.org/document/6296083/

[73] C. Dwork, "Differential Privacy: A Survey of Results," in *Theory and Applications of Models of Computation*, ser. Lecture Notes in Computer Science, M. Agrawal, D. Du, Z. Duan, and A. Li, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, vol. 4978, pp. 1–19. [Online]. Available: http://www.springerlink.com/index/u963k75981004046.pdf{\%}5Cnhttp://link.springer.com/content/pdf/10.1007/978-3-642-38236-9.pdfhttp://link.springer.com/10.1007/978-3-540-79228-4http://link.springer.com/10.1007/978-3-540-79228-4{_}1

[74] K. M. P. Shrivastva, M. Rizvi, and S. Singh, “Big Data Privacy Based on Differential Privacy a Hope for Big Data,” in *2014 International Conference on Computational Intelligence and Communication Networks*. IEEE, nov 2014, pp. 776–781. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7065587http://ieeexplore.ieee.org/document/7065587/

[75] M. Langheinrich, “A Privacy Awareness System for Ubiquitous Computing Environments,” in *UbiComp ’02: Proceedings of the 4th international conference on Ubiquitous Computing*, 2002, pp. 237–245. [Online]. Available: http://link.springer.com/10.1007/3-540-45809-3{_}19

[76] L. Barkhuus and A. Dey, “Location-Based Services for Mobile Telephony: A Study of Users’ Privacy Concerns,” *Interact 2003*, vol. 3, pp. 702–712, 2003. [Online]. Available: http://www.intel-research.net/Publications/Berkeley/072920031046{_}154.pdf

[77] M. P. Scipioni and M. Langheinrich, “I’ m Here! Privacy Challenges in Mobile Location Sharing,” *IWSSI/SPMU*, 2010.

[78] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos, “AnonySense: A system for anonymous opportunistic sensing,” *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 16–30, feb 2011. [Online]. Available: http://dx.doi.org/10.1016/j.pmcj.2010.04.001http://linkinghub.elsevier.com/retrieve/pii/S1574119210000398

[79] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, “Anonysense,” in *Proceeding of the 6th international conference on Mobile systems, applications, and services - MobiSys ’08*. New York, New York, USA: ACM Press, 2008, p. 211. [Online]. Available: http://dl.acm.org/citation.cfm?id=1378624http://portal.acm.org/citation.cfm?doid=1378600.1378624

[80] M. Covington, R. Krishnan, and M. Sastry, “Methods and apparatuses for privacy in location-aware systems,” *US Patent App. 11/772,196*, no. 19, 2007. [Online]. Available: http://www.google.com/patents/US20100024045

[81] J. Krumm, “A survey of computational location privacy,” *Personal and Ubiquitous Computing*, vol. 13, no. 6, pp. 391–399, aug 2009. [Online]. Available: http://link.springer.com/10.1007/s00779-008-0212-5

[82] T. Hashem and L. Kulik, ““Don’t trust anyone”: Privacy protection for location-based services,” *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 44–59, feb 2011. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S1574119210000520

[83] H.-i. Kim and Y.-k. Kim, “A Grid-based Cloaking Area Creation Scheme for Continuous LBS Queries in Distributed Systems,” *Journal of Convergence*, vol. 4, no. 1, pp. 23–30, 2013.

[84] G. Ghinita, P. Kalnis, and S. Skiadopoulos, “PRIVE: Anonymous Location-Based Queries in Distributed Mobile Systems,” in *Proceedings of the 16th international conference on World Wide Web - WWW ’07*. New York, New York, USA: ACM Press, 2007, p. 371. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1242572.1242623

[85] ——, “MobiHide: A Mobilea Peer-to-Peer System for Anonymous Location-Based Queries,” in *Advances in Spatial and Temporal Databases*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 4605, pp. 221–238. [Online]. Available: http://www.springerlink.com/index/m64m21610j686626.pdfhttp://link.springer.com/10.1007/978-3-540-73540-3{_}13

[86] K. Liu, C. Giannella, and H. Kargupta, “A Survey of Attack Techniques on Privacy-Preserving Data Perturbation Methods,” in *Privacy-Preserving Data Mining*, 2008, pp. 359–381. [Online]. Available: http://link.springer.com/10.1007/978-0-387-70992-5{_}15

[87] B. D. Okkalioglu, M. Okkalioglu, M. Koc, and H. Polat, “A survey: deriving private information from perturbed data,” *Artificial Intelligence Review*, vol. 44, no. 4, pp. 547–569, dec 2015. [Online]. Available: http://link.springer.com/10.1007/s10462-015-9439-5

[88] J. Krumm, “Inference Attacks on Location Tracks,” in *Pervasive Computing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, vol. 10, no. Pervasive, pp. 127–143. [Online]. Available: http://www.springerlink.com/index/TG64551RW2716103.pdf{\%}5Cnhttp://research.microsoft.com/en-us/um/people/jckrumm/publications2007/inferenceattackrefined02distribute.pdfhttp://link.springer.com/10.1007/978-3-540-72037-9{_}8

[89] K. Tan, G. Yan, J. Yeo, and D. Kotz, “A correlation attack against user mobility privacy in a large-scale WLAN network,” in *Proceedings of the 2010 ACM workshop on Wireless of the students, by the students, for the students - S3 ’10*. New York, New York, USA: ACM Press, 2010, p. 33. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1860039.1860050

[90] G. Cormode, “Personal privacy vs population privacy: Learning to Attack Anonymization,” in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD ’11*. New York, New York, USA: ACM Press, 2011, p. 1253. [Online]. Available: http://arxiv.org/abs/1011.2511http://dl.acm.org/citation.cfm?doid=2020408.2020598

[91] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, “PriSense: Privacy-Preserving Data Aggregation in People-Centric Urban Sensing Systems,” in *2010 Proceedings IEEE INFOCOM*. IEEE, mar 2010, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5462147http://ieeexplore.ieee.org/document/5462147/

[92] Z. Wei, B. Zhao, and J. Su, “PDA: A Novel Privacy-Preserving Robust Data Aggregation Scheme in People-Centric Sensing System,” *International*

Journal of Distributed Sensor Networks, vol. 9, no. 11, p. 147839, nov 2013. [Online]. Available: http://journals.sagepub.com/doi/10.1155/2013/147839

[93] Rui Zhang, Jing Shi, Yanchao Zhang, and Chi Zhang, "Verifiable Privacy-Preserving Aggregation in People-Centric Urban Sensing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 268–278, sep 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6559974/

[94] S. M. Erfani, S. Karunasekera, C. Leckie, and U. Parampalli, "Privacy-preserving data aggregation in Participatory Sensing Networks," in *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, vol. 1. IEEE, apr 2013, pp. 165–170. [Online]. Available: http://ieeexplore.ieee.org/document/6529783/

[95] Y. Zhang, Q. Chen, and S. Zhong, "Privacy-Preserving Data Aggregation in Mobile Phone Sensing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 980–992, may 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7373649/

[96] D. H. T. That, I. S. Popa, K. Zeitouni, and C. Borcea, "PAMPAS: Privacy-Aware Mobile Participatory Sensing Using Secure Probes," in *Proceedings of the 28th International Conference on Scientific and Statistical Database Management - SSDBM '16*. New York, New York, USA: ACM Press, 2016, pp. 1–12. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2949689.2949704

[97] M. Shin, C. Cornelius, A. Kapadia, N. Triandopoulos, and D. Kotz, "Location Privacy for Mobile Crowd Sensing through Population Mapping," *Sensors*, vol. 15, no. 7, pp. 15 285–15 310, jun 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/26131676http://www.mdpi.com/1424-8220/15/7/15285

[98] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, feb 1981. [Online]. Available: http://portal.acm.org/citation.cfm?doid=358549.358563

[99] C. A. Neff, "A verifiable secret shuffle and its application to e-voting," in *Proceedings of the 8th ACM conference on Computer and Communications Security - CCS '01*. New York, New York, USA: ACM Press, 2001, p. 116. [Online]. Available: http://portal.acm.org/citation.cfm?doid=501983.502000{\%}5Cnhttp://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.104.3124{\%}5Cnhttp://www.pubzone.org/dblp/conf/ccs/Neff01http://portal.acm.org/citation.cfm?doid=501983.502000

[100] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, "PDA: Privacy-Preserving Data Aggregation in Wireless Sensor Networks," in *IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications*. IEEE, 2007, pp. 2045–2053. [Online]. Available: http://ieeexplore.ieee.org/document/4215819/

[101] T. Li, N. Li, J. Zhang, and I. Molloy, “Slicing: A New Approach for Privacy Preserving Data Publishing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 561–574, mar 2012. [Online]. Available: http://ieeexplore.ieee.org/document/5645625/

[102] F. Qiu, F. Wu, and G. Chen, “SLICER: A Slicing-Based K-Anonymous Privacy Preserving Scheme for Participatory Sensing,” in *2013 IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems*. IEEE, oct 2013, pp. 113–121. [Online]. Available: http://ieeexplore.ieee.org/document/6680230/

[103] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in *Proceedings of the 41st annual ACM symposium on Symposium on theory of computing - STOC ’09*. New York, New York, USA: ACM Press, 2009, p. 169. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1536414.1536440

[104] K. El Makkaoui, A. Ezzati, and A. B. Hssane, “Challenges of using homomorphic encryption to secure cloud computing,” in *2015 International Conference on Cloud Technologies and Applications (CloudTech)*. IEEE, jun 2015, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/document/7337011/

[105] W. Wang, Y. Hu, L. Chen, X. Huang, and B. Sunar, “Exploring the Feasibility of Fully Homomorphic Encryption,” *IEEE Transactions on Computers*, vol. 64, no. 3, pp. 698–706, mar 2015. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs{_}all.jsp?arnumber=6573943{\%}5Cnhttp://ecewp.ece.wpi.edu/wordpress/crypto/files/2012/12/journal0705.pdfhttp://ieeexplore.ieee.org/document/6573943/

[106] L. J. M. Aslett, P. M. Esperança, and C. C. Holmes, “A review of homomorphic encryption and software tools for encrypted statistical machine learning,” pp. 1–21, aug 2015. [Online]. Available: http://arxiv.org/abs/1508.06574

[107] L. Wang, J. Li, and H. Ahmad, “Challenges of fully homomorphic encryptions for the internet of things,” *IEICE Transactions on Information and Systems*, vol. E99D, no. 8, pp. 1982–1990, 2016.

[108] N. Saputro and K. Akkaya, “Performance evaluation of Smart Grid data aggregation via homomorphic encryption,” in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, apr 2012, pp. 2945–2950. [Online]. Available: http://ieeexplore.ieee.org/document/6214307/

[109] R. Lu, H. Zhu, X. Liu, J. K. Liu, and J. Shao, “Toward efficient and privacy-preserving computing in big data era,” *IEEE Network*, vol. 28, no. 4, pp. 46–50, jul 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6863131/

[110] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "CryptoNets: Applying neural networks to Encrypted data with high throughput and accuracy," in *Proceedings of the 33rd International Conference on Machine Learning*, New York, NY, USA, 2016, pp. 1–12. [Online]. Available: http://research.microsoft.com/apps/pubs/?id=260989

[111] T. M. T. Do and D. Gatica-Perez, "The Places of Our Lives: Visiting Patterns and Automatic Labeling from Longitudinal Smartphone Data," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 638–648, mar 2014. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6427751http://ieeexplore.ieee.org/document/6427751/

[112] M. D. Redzic, C. Brennan, and N. E. O'Connor, "SEAMLOC: Seamless Indoor Localization Based on Reduced Number of Calibration Points," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1326–1337, jun 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6583154/

[113] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, nov 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4343996/

[114] S. P. Banerjee and D. Woodard, "Biometric Authentication and Identification Using Keystroke Dynamics: A Survey," *Journal of Pattern Recognition Research*, vol. 7, no. 1, pp. 116–139, 2012. [Online]. Available: http://www.jprr.org/index.php/jprr/article/view/427/167

[115] J. Unar, W. C. Seng, and A. Abbasi, "A review of biometric technology along with trends and prospects," *Pattern Recognition*, vol. 47, no. 8, pp. 2673–2688, aug 2014. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2014.01.016http://linkinghub.elsevier.com/retrieve/pii/S003132031400034X

[116] A. K. Jain, K. Nandakumar, and A. Ross, "50 years of biometric research: Accomplishments, challenges, and opportunities," *Pattern Recognition Letters*, vol. 79, pp. 80–105, aug 2016. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0167865515004365

[117] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, "Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, jan 2013. [Online]. Available: http://ieeexplore.ieee.org/document/6331527/

[118] M. Antal, L. Z. Szabó, and I. László, "Keystroke Dynamics on Android Platform," *Procedia Technology*, vol. 19, pp. 820–826, 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S221201731500119X

[119] J.-h. Roh, S.-H. Lee, and S. Kim, "Keystroke dynamics for authentication in smartphone," in *2016 International Conference on Information and*

Communication Technology Convergence (ICTC). IEEE, oct 2016, pp. 1155–1159. [Online]. Available: http://ieeexplore.ieee.org/document/7763394/

[120] Z. Sitova, J. Sedenka, Q. Yang, G. Peng, G. Zhou, P. Gasti, and K. S. Balagani, "HMOG: New Behavioral Biometric Features for Continuous Authentication of Smartphone Users," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 877–892, may 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7349202/

[121] T. Kinnunen and H. Li, "An overview of text-independent speaker recognition: From features to supervectors," *Speech Communication*, vol. 52, no. 1, pp. 12–40, jan 2010. [Online]. Available: http://dx.doi.org/10.1016/j.specom.2009.08.009http://linkinghub.elsevier.com/retrieve/pii/S0167639309001289

[122] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2014, pp. 1695–1699. [Online]. Available: http://ieeexplore.ieee.org/document/6853887/

[123] F. Richardson, D. Reynolds, and N. Dehak, "Deep Neural Network Approaches to Speaker and Language Recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, oct 2015. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7080838http://ieeexplore.ieee.org/document/7080838/

[124] Jianfeng Chen, Jianmin Zhang, A. Kam, and L. Shue, "An Automatic Acoustic Bathroom Monitoring System," in *2005 IEEE International Symposium on Circuits and Systems*. IEEE, 2005, pp. 1750–1753. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1464946http://ieeexplore.ieee.org/document/1464946/

[125] Y. Zhan, J. Nishimura, and T. Kuroda, "Human Activity Recognition from Environmental Background Sounds for Wireless Sensor Networks," *IEEJ Transactions on Electronics, Information and Systems*, vol. 130, no. 4, pp. 565–572, 2010. [Online]. Available: http://joi.jlc.jst.go.jp/JST.JSTAGE/ieejeiss/130.565?from=CrossRef

[126] J. a. Stork, L. Spinello, J. Silva, and K. O. Arras, "Audio-based human activity recognition using Non-Markovian Ensemble Voting," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*. IEEE, sep 2012, pp. 509–514. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6343802http://ieeexplore.ieee.org/document/6343802/

[127] D. Asonov and R. Agrawal, "Keyboard acoustic emanations," in *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, vol. 2004. IEEE, 2004, pp. 3–11. [Online]. Available: http://ieeexplore.ieee.org/document/1301311/

[128] Y. Berger, A. Wool, and A. Yeredor, "Dictionary attacks using keyboard acoustic emanations," in *Proceedings of the 13th ACM conference on Computer and communications security - CCS '06*. New York, New York, USA: ACM Press, 2006, p. 245. [Online]. Available: http://dl.acm.org/citation.cfm?id=1180436{\&}CFID=104977675{\&}CFTOKEN=15114097http://portal.acm.org/citation.cfm?doid=1180405.1180436

[129] L. Zhuang, F. Zhou, and J. D. Tygar, "Keyboard acoustic emanations revisited," *ACM Transactions on Information and System Security*, vol. 13, no. 1, pp. 1–26, oct 2009. [Online]. Available: http://ieeexplore.ieee.org/document/1301311/http://portal.acm.org/citation.cfm?doid=1609956.1609959

[130] H. Malik, "Acoustic Environment Identification and Its Applications to Audio Forensics," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1827–1837, nov 2013. [Online]. Available: http://dblp.uni-trier.de/db/journals/tifs/tifs8.html{\#}Malik13http://ieeexplore.ieee.org/document/6595031/

[131] H. Zhao and H. Malik, "Audio recording location identification using acoustic environment signature," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 11, pp. 1746–1759, 2013.

[132] R. K. Patole, P. Rege, and P. Suryawanshi, "Acoustic environment identification using blind de-reverberation," in *2016 International Conference on Computing, Analytics and Security Trends (CAST)*. IEEE, dec 2016, pp. 495–500. [Online]. Available: http://ieeexplore.ieee.org/document/7915019/

[133] G. Delgado-Gutiérrez, F. Rodríguez-Santos, O. Jiménez-Ramírez, and R. Vázquez-Medina, "Acoustic environment identification by Kullback–Leibler divergence," *Forensic Science International*, vol. 281, pp. 134–140, dec 2017. [Online]. Available: http://dx.doi.org/10.1016/j.forsciint.2017.10.031http://linkinghub.elsevier.com/retrieve/pii/S0379073817304334

[134] S. Tervo and T. Tossavainen, "3D room geometry estimation from measured impulse responses," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, mar 2012, pp. 513–516. [Online]. Available: http://ieeexplore.ieee.org/document/6287929/

[135] D. Markovica, F. Antonacci, A. Sarti, and S. Tubaro, "Estimation of room dimensions from a single impulse response," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, vol. 1, no. 1. IEEE, oct 2013, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/document/6701867/

[136] T. Rajapaksha, X. Qiu, E. Cheng, and I. Burnett, "Geometrical room geometry estimation from room impulse responses," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2016-May. IEEE, mar 2016, pp. 331–335. [Online]. Available: http://ieeexplore.ieee.org/document/7471691/

[137] A. R. Zamir and M. Shah, “Accurate Image Localization Based on Google Maps Street View,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, Berlin, Heidelberg, 2010, vol. 6314 LNCS, no. PART 4, pp. 255–268. [Online]. Available: http://link.springer.com/10.1007/978-3-642-15561-1{_}19

[138] G. Baatz, K. Köser, D. Chen, R. Grzeszczuk, and M. Pollefeys, “Handling Urban Location Recognition as a 2D Homothetic Problem,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, Berlin, Heidelberg, 2010, vol. 6316 LNCS, no. PART 6, pp. 266–279. [Online]. Available: http://link.springer.com/10.1007/978-3-642-15567-3{_}20

[139] T. Weyand, I. Kostrikov, and J. Philbin, “PlaNet - Photo Geolocation with Convolutional Neural Networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9912 LNCS, pp. 37–55. [Online]. Available: http://link.springer.com/10.1007/978-3-319-46484-8{_}3

[140] S. Karayev, A. Hertzmann, M. Trentacoste, H. Han, H. Winnemoeller, A. Agarwala, and T. Darrell, “Recognizing Image Style,” in *Proceedings of the British Machine Vision Conference 2014*. British Machine Vision Association, 2014, pp. 122.1–122.11. [Online]. Available: http://arxiv.org/abs/1311.3715http://www.bmva.org/bmvc/2014/papers/paper121/index.html

[141] C. Thomas and A. Kovashka, “Seeing Behind the Camera: Identifying the Authorship of a Photograph,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, pp. 3494–3502. [Online]. Available: http://ieeexplore.ieee.org/document/7780749/

[142] Y. Hoshen and S. Peleg, “An Egocentric Look at Video Photographer Identity,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2016-Decem. IEEE, jun 2016, pp. 4284–4292. [Online]. Available: http://ieeexplore.ieee.org/document/7780833/

[143] H. Bojinov, Y. Michalevsky, G. Nakibly, and D. Boneh, “Mobile Device Identification via Sensor Fingerprinting,” aug 2014. [Online]. Available: http://arxiv.org/abs/1408.1416

[144] G. Baldini, G. Steri, F. Dimc, R. Giuliani, and R. Kamnik, “Experimental Identification of Smartphones Using Fingerprints of Built-In Micro-Electro Mechanical Systems (MEMS),” *Sensors*, vol. 16, no. 6, p. 818, jun 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/6/818

[145] T. Van Goethem, W. Scheepers, D. Preuveneers, and W. Joosen, “Accelerometer-Based Device Fingerprinting for Multi-factor Mobile Authentication,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9639, pp. 106–121. [Online]. Available: http://link.springer.com/10.1007/978-3-319-30806-7{_}7

[146] G. Baldini, F. Dimc, R. Kamnik, G. Steri, R. Giuliani, and C. Gentile, "Identification of Mobile Phones Using the Built-In Magnetometers Stimulated by Motion Patterns," *Sensors*, vol. 17, no. 4, p. 783, apr 2017. [Online]. Available: http://arxiv.org/abs/1701.07676http://www.mdpi.com/1424-8220/17/4/783

[147] G. Baldini, G. Steri, R. Giuliani, and V. Kyovtorov, "Mobile phone identification through the built-in magnetometers," pp. 1–10, jan 2017. [Online]. Available: http://arxiv.org/abs/1701.07676

[148] G. Lammel, J. Gutmann, L. Marti, and M. Dobler, "Indoor Navigation with MEMS sensors," *Procedia Chemistry*, vol. 1, no. 1, pp. 532–535, sep 2009. [Online]. Available: http://dx.doi.org/10.1016/j.proche.2009.07.133http://linkinghub.elsevier.com/retrieve/pii/S187661960900134X

[149] K. Subbu, C. Zhang, J. Luo, and A. Vasilakos, "Analysis and status quo of smartphone-based indoor localization systems," *IEEE Wireless Communications*, vol. 21, no. 4, pp. 106–112, aug 2014. [Online]. Available: http://ieeexplore.ieee.org/document/6882302/

[150] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp '13*. New York, New York, USA: ACM Press, 2013, p. 225. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2493432.2493449

[151] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, p. 74, mar 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1964918http://portal.acm.org/citation.cfm?doid=1964897.1964918

[152] O. D. Incel, M. Kose, and C. Ersoy, "A Review and Taxonomy of Activity Recognition on Mobile Phones," *BioNanoScience*, vol. 3, no. 2, pp. 145–171, jun 2013. [Online]. Available: http://link.springer.com/10.1007/s12668-013-0088-3

[153] A. Bayat, M. Pomplun, and D. A. Tran, "A Study on Human Activity Recognition Using Accelerometer Data from Smartphones," *Procedia Computer Science*, vol. 34, no. C, pp. 450–457, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2014.07.009http://linkinghub.elsevier.com/retrieve/pii/S1877050914008643

[154] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Cell phone-based biometric identification," in *2010 Fourth IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. IEEE, sep 2010, pp. 1–7. [Online]. Available: http://ieeexplore.ieee.org/document/5634532/

[155] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. R. Choudhury, "Tapprints: Your Finger Taps Have Fingerprints," in *Proceedings of the 10th international conference on Mobile systems, applications, and services*

- *MobiSys '12*. New York, New York, USA: ACM Press, 2012, p. 323. [Online]. Available: http://dl.acm.org/citation.cfm?id=2307666{\%}5Cnhttp://dl.acm.org/citation.cfm?doid=2307636.2307666http://dl.acm.org/citation.cfm?doid=2307636.2307666

[156] P. Marquardt, A. Verma, H. Carter, and P. Traynor, "(sp)iPhone: Decoding Vibrations From Nearby Keyboards Using Mobile Phone Accelerometers," in *Proceedings of the 18th ACM conference on Computer and communications security - CCS '11*. New York, New York, USA: ACM Press, 2011, p. 551. [Online]. Available: http://dl.acm.org/citation.cfm?id=2046771http://dl.acm.org/citation.cfm?doid=2046707.2046771

[157] L. Zhang, P. H. Pathak, M. Wu, Y. Zhao, and P. Mohapatra, "AccelWord: Energy Efficient Hotword Detection through Accelerometer," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys '15*. New York, New York, USA: ACM Press, 2015, pp. 301–315. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2742647.2742658

[158] Y. Michalevsky, D. Boneh, and G. Nakibly, "Gyrophone: Recognizing Speech from Gyroscope Signals," *Usenix Security*, pp. 1053–1067, 2014. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/michalevsky

[159] Seong-Eun Kim, Yong Kim, Jihyun Yoon, and Eung Sun Kim, "Indoor positioning system using geomagnetic anomalies for smartphones," in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, no. November. IEEE, nov 2012, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/6418947/

[160] E. Le Grand and S. Thrun, "3-Axis magnetic field mapping and fusion for indoor localization," in *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, no. 2. IEEE, sep 2012, pp. 358–364. [Online]. Available: http://ieeexplore.ieee.org/document/6343024/

[161] K. P. Subbu, B. Gozick, and R. Dantu, "LocateMe: Magnetic-Fields-Based Indoor Localization Using Smartphones," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, pp. 1–27, sep 2013. [Online]. Available: http://doi.acm.org/10.1145/2508037.2508054http://dl.acm.org/citation.cfm?doid=2508037.2508054

[162] H. Xia, X. Wang, Y. Qiao, J. Jian, and Y. Chang, "Using Multiple Barometers to Detect the Floor Location of Smart Phones with Built-in Barometric Sensors for Indoor Positioning," *Sensors*, vol. 15, no. 4, pp. 7857–7877, mar 2015. [Online]. Available: http://www.mdpi.com/1424-8220/15/4/7857

[163] H. Ye, T. Gu, X. Tao, and J. Lu, "Scalable floor localization using barometer on smartphone," *Wireless Communications and Mobile Computing*, vol. 16, no. 16, pp. 2557–2571, nov 2016. [Online]. Available: http://eprints.soton.ac.uk/266684/http://doi.wiley.com/10.1002/wcm.2706

[164] W. Falcon and H. Schulzrinne, “Predicting Floor-Level for 911 Calls with Neural Networks and Smartphone Sensor Data,” no. June 2017, pp. 1–16, oct 2017. [Online]. Available: http://arxiv.org/abs/1710.11122

[165] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen, “IODetector: A Generic Service for Indoor Outdoor Detection,” in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems - SenSys ’12*. New York, New York, USA: ACM Press, 2012, p. 113. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2426656.2426668

[166] H. G. Kayacik, M. Just, L. Baillie, D. Aspinall, and N. Micallef, “Data Driven Authentication: On the Effectiveness of User Behaviour Modelling with Mobile Device Sensors,” *Proceedings of the 3rd Workshop on Mobile Security Technologies (MoST) 2014*, oct 2014. [Online]. Available: http://arxiv.org/abs/1410.7743

[167] N. Micallef, H. G. Kayacik, M. Just, L. Baillie, and D. Aspinall, “Sensor use and usefulness: Trade-offs for data-driven authentication on mobile devices,” in *2015 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, mar 2015, pp. 189–197. [Online]. Available: http://ieeexplore.ieee.org/document/7146528/

[168] R. Spreitzer, “PIN Skimming: Exploiting the Ambient-Light Sensor in Mobile Devices,” in *Proceedings of the 4th ACM Workshop on Security and Privacy in Smartphones & Mobile Devices - SPSM ’14*. New York, New York, USA: ACM Press, 2014, pp. 51–62. [Online]. Available: http://arxiv.org/abs/1405.3760http://dl.acm.org/citation.cfm?doid=2666620.2666622

[169] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao, “Epsilon: A Visible Light Based Positioning System,” *USENIX Symposium on Network Systems Design and Implementation*, no. 1, pp. 1–13, 2014. [Online]. Available: http://panhu.me/pdf/Epsilon.pdf

[170] S. E. Z. Mazilu and G. E. Z. Tröster, “A Study on Using Ambient Sensors from Smartphones for Indoor Location Detection,” *12th Workshop On positioning, navigation and communication (WPNC). IEEE.*, 2015.

[171] D. R. J. Bassett, B. E. Ainsworth, S. R. Leggett, C. A. Mathien, J. A. Main, D. C. Hunter, and G. E. Duncan, “Accuracy of five electronic pedometers for measuring distance walked,” *Medicine & Science in Sports & Exercise*, vol. 28, no. 8, pp. 1071–1077, 1996.

[172] C. Tudor-Locke, J. E. Williams, J. P. Reis, and D. Pluto, “Utility of Pedometers for Assessing Physical Activity,” *Sports Medicine*, vol. 32, no. 12, pp. 795–808, 2002. [Online]. Available: http://link.springer.com/10.2165/00007256-200232120-00004

[173] S. Crouter, P. L. Schneider, M. Karabulut, and D. R. J. Bassett, “Validity of 10 Electronic Pedometers for Measuring Steps, Distance, and Energy Cost,” *Medicine & Science in Sports & Exercise*, vol. 35, no. 8, pp. 1455–1460, aug

2003. [Online]. Available: https://insights.ovid.com/crossref?an=00005768-200308000-00030

[174] J. Chon and Hojung Cha, "LifeMap: A Smartphone-Based Context Provider for Location-Based Services," *IEEE Pervasive Computing*, vol. 10, no. 2, pp. 58–67, apr 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5686873/

[175] C. Welch, "Google took down over 700,000 bad Android apps in 2017," 2018. [Online]. Available: https://www.theverge.com/2018/1/30/16951996/google-android-apps-removed-security-2017 (Accessed: June 14, 2018).

[176] A. Sulleyman, "Millions of people installed infected apps that criminals could use to 'wreak havoc' on smartphones," 2018. [Online]. Available: https://www.independent.co.uk/life-style/gadgets-and-tech/news/android-apps-google-play-five-nights-survival-guide-adultswine-malware-smartphones-checkpoint-a8159916.html (Accessed: June 14, 2018).

[177] Androidrank, "Androidrank," 2018. [Online]. Available: https://www.androidrank.org/ (Accessed: June 14, 2018).

[178] Google, "Google Play Store," 2018. [Online]. Available: https://play.google.com (Accessed: June 14, 2018).

[179] A. Pal, "Localization algorithms in wireless sensor networks: Current approaches and future challenges," *Network Protocols and Algorithms*, vol. 2, no. 1, pp. 45–73, 2010.

[180] M. Buer, C. Abraham, D. Garrett, J. Karaoguz, D. Lundgren, and D. Murray, "Method and system for authorizing transactions based on relative location of devices," US Patent App. 12/748,175, 2011.

[181] O. De Silva, G. K. I. Mann, and R. G. Gosine, "Pairwise observable relative localization in ground aerial multi-robot networks," in *2014 European Control Conference (ECC)*. IEEE, jun 2014, pp. 324–329. [Online]. Available: http://ieeexplore.ieee.org/document/6862597/

[182] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic, "Augmented reality technologies, systems and applications," *Multimedia Tools and Applications*, vol. 51, no. 1, pp. 341–377.

[183] D. Wang, D. Pedreschi, C. Song, F. Giannotti, and A.-L. Barabasi, "Human mobility, social ties, and link prediction," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*. New York, New York, USA: ACM Press, 2011, p. 1100. [Online]. Available: http://dl.acm.org/citation.cfm?id=2020408.2020581http://dl.acm.org/citation.cfm?doid=2020408.2020581

[184] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering personally meaningful places: An interactive clustering

approach," *ACM Transactions on Information Systems*, vol. 25, no. 3, pp. 12–es, jul 2007. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1247715.1247718

[185] G. Zanca, F. Zorzi, A. Zanella, and M. Zorzi, "Experimental Comparison of RSSI-based Localization Algorithms for Indoor Wireless Sensor Networks," in *Proceedings of the Workshop on Real-world Wireless Sensor Networks. REALWSN '08.* ACM New York, NY, USA, 2008, pp. 1–5.

[186] J. Hull, S. Jalilian, and P. Cheuk, "Loudness based method and system for determining relative location of an acoustic event along a channel," US Patent App. 13/958,358, 2014.

[187] D. Walker and S. Fyke, "System and method of relative location detection using image perspective analysis," US Patent 8,612,149, 2013.

[188] G. Xu, S. Gao, M. Daneshmand, C. Wang, and Y. Liu, "A Survey for Mobility Big Data Analytics for Geolocation Prediction," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 111–119, feb 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7731599/

[189] S. Barbera, P. H. Michaelsen, M. Saily, and K. Pedersen, "Mobility performance of LTE co-channel deployment of macro and pico cells," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, no. May 2016. IEEE, apr 2012, pp. 2863–2868. [Online]. Available: http://ieeexplore.ieee.org/document/6214290/

[190] C. Coletti, L. Hu, N. Huan, I. Z. Kovács, B. Vejlgaard, R. Irmer, and N. Scully, "Heterogeneous Deployment to Meet Traffic Demand in a Realistic LTE Urban Scenario," in *2012 IEEE Vehicular Technology Conference (VTC Fall)*. IEEE, sep 2012, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/6399137/

[191] G. Rubino and B. Sericola, "Sojourn times in finite Markov processes," *Journal of Applied Probability*, vol. 26, no. 04, pp. 744–756, dec 1989. [Online]. Available: http://www.jstor.org/stable/3214379https://www.cambridge.org/core/product/identifier/S0021900200027613/type/journal{_}article

[192] M. A. Bayir, M. Demirbas, and N. Eagle, "Discovering spatiotemporal mobility profiles of cellphone users," in *2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops*. IEEE, jun 2009, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/document/5282489/

[193] L. Goldfarb, "A new approach to pattern recognition," in *Progress in Pattern Recognition 2*, vol. 2, 1985, pp. 241–402.

[194] V. Roth, J. Laub, M. Kawanabe, and J. Buhmann, "Optimal cluster preserving embedding of nonmetric proximity data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1540–1551, dec 2003. [Online]. Available: http://ieeexplore.ieee.org/document/1251147/

[195] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, “A Density-Based Algorithm for Discovering clusters in Large Spatial Databases with Noise,” in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press, 1996, pp. 226–231.

[196] S. Stals, M. Smyth, and O. Mival, “Exploring People’s Emotional Bond with Places in the City,” in *Proceedings of the 2016 ACM Conference Companion Publication on Designing Interactive Systems - DIS ’17 Companion*. New York, New York, USA: ACM Press, 2017, pp. 207–212. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3064857.3079147

[197] B. Guo, Z. Yu, X. Zhou, and D. Zhang, “From participatory sensing to Mobile Crowd Sensing,” in *2014 IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*. IEEE, mar 2014, pp. 593–598. [Online]. Available: http://ieeexplore.ieee.org/document/6815273/

[198] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila, “Towards rich mobile phone datasets: Lausanne data collection campaign,” *Proceedings ACM International Conference on Pervasive Services (ICPS)*, vol. Berlin, 2010. [Online]. Available: http://www.idiap.ch/{∼}gatica/publications/KiukkonenBlomDousseGaticaLaurila-icps10.pdf

[199] J. K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen, “The mobile data challenge: Big data for mobile computing research,” *Proceedings of the Workshop on the Nokia Mobile Data Challenge, in Conjunction with the 10th International Conference on Pervasive Computing*, pp. 1–8, 2012.

[200] EDPB, “GDPR Certification Mechanisms, Seals, and Marks,” 2018. [Online]. Available: https://edpb.europa.eu/our-work-tools/accountability-tools/certification-mechanisms-seals-and-marks{_}en (Accessed: October 19, 2018).