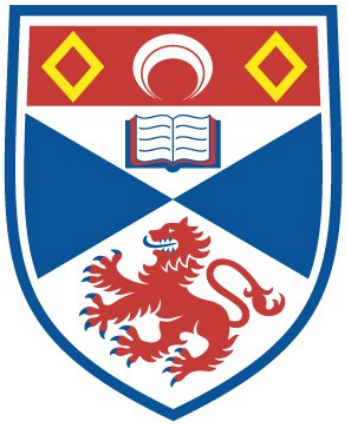
RECOMMENDING PRIVACY PREFERENCES IN LOCATION-SHARING SERVICES

Yuchen Zhao

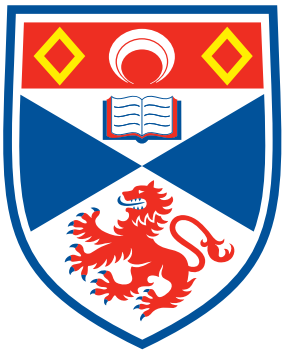
A Thesis Submitted for the Degree of PhD
at the
University of St Andrews



2017

Recommending Privacy Preferences in Location-Sharing Services

Yuchen Zhao



University of
St Andrews

This thesis is submitted in partial fulfilment for the degree of

Doctor of Philosophy

at the University of St Andrews

May 2017

Abstract

Location-sharing services have become increasingly popular with the proliferation of smartphones and online social networks. People share their locations with each other to record their daily lives or satisfy their social needs. At the same time, inappropriate disclosure of location information poses threats to people's privacy.

One of the reasons why people fail to protect their location privacy is the difficulty of using the current mechanisms to manually configure location-privacy settings. Since people's location-privacy preferences are context-aware, manual configuration is cumbersome. People's incapability and unwillingness to do so lead to unexpected location disclosures that violate their location privacy.

In this thesis, we investigate the feasibility of using recommender systems to help people protect their location privacy. We examine the performance of location-privacy recommender systems and compare it with the state-of-the-art. We also conduct online user studies to understand people's acceptance of such recommender systems and their concerns. We revise our design of the systems according to the results of the user studies.

We find that user-based collaborative filtering can accurately recommend location-privacy preferences and outperform the state-of-the-art when training data are insufficient. From users' perspective, their acceptance of location-privacy recommender systems is affected by the openness and the context of recommendations and their privacy concerns about the systems. It is feasible to use data obfuscation or decentralisation to alleviate people's concerns and meanwhile keep the systems robust against malicious data attacks.

Acknowledgements

It has been a long but wonderful journey to finish this thesis. I would like to give my thanks to several people who have helped me.

First of all, I would like to thank my supervisors, Tristan Henderson and Juan Ye, who have offered endless support and guidance during my PhD. Their supervision has not only helped me finish my studies but also taught me the importance of critical thinking. The experience of my PhD under their supervision will inspire me in the future.

I would like to thank the School of Computer Science for funding my PhD over the last 3.5 years. Many thanks to my reviewers: Adam Barker, Mike Weir, Susmit Sarkar, Simon Dobson, and Ishbel Duncan, for giving me feedback about my research to keep me on track. My thanks also go to the administration team and the technician team of the school for supporting my studies.

I am lucky to work with many talented friends and colleagues. Thanks to Ditchaphong Phoomikiattisak, Chonlatee Khorakhun, Luke Hutton, Khawar Shehzad, and Lei Fang, for helping me with the experiments in Chapters 5 and 6.

Finally, thank you to my family for constantly supporting me and giving me confidence.

Declarations

Candidate's Declarations

I, Yuchen Zhao, hereby certify that this thesis, which is approximately 27,000 words in length, has been written by me, and that it is the record of work carried out by me, or principally by myself in collaboration with others as acknowledged, and that it has not been submitted in any previous application for a higher degree.

I was admitted as a research student in September 2013 and as a candidate for the degree of Doctor of Philosophy in September 2013; the higher study for which this is a record was carried out in the University of St Andrews between 2013 and 2017.

date: _____ signature of candidate: _____

Supervisor's Declaration

I hereby certify that the candidate has fulfilled the conditions of the Resolution and Regulations appropriate for the degree of Doctor of Philosophy in the University of St Andrews and that the candidate is qualified to submit this thesis in application for that degree.

date: _____ signature of supervisor: _____

Permission of publication

In submitting this thesis to the University of St Andrews I understand that I am giving permission for it to be made available for use in accordance with the regulations of the University Library for the time being in force, subject to any copyright vested in the work not being affected thereby. I also understand that the title and the abstract will be published, and that a copy of the work may be made and supplied to any bona fide library or research worker, that my thesis will be electronically accessible for personal or research use unless exempt by award of an embargo as requested below, and that the library has the right to migrate my thesis into new electronic forms as required to ensure continued access to the thesis. I have obtained any third-party copyright permissions that may be required in order to allow such access and migration, or have requested the appropriate embargo below.

The following is an agreed request by candidate and supervisor regarding the electronic publication of this thesis:

PRINTED COPY

No embargo on print copy

ELECTRONIC COPY

No embargo on electronic copy

date: _____

signature of candidate: _____

signature of supervisor: _____

"It is true of any subject that the person that succeeds in anything has the realistic viewpoint at the beginning, knowing that the problem is large and that he has to take it a step at a time and he has to enjoy the step-by-step learning procedure."

Bill Evans

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1991, Mark Weiser discussed his perspective [140] about how computers in the 21st century would become. As described in his article, these computers would exist in different types, from pocket-size computers, i.e., tabs, to page-size ones, i.e., pads. All of these computers would be integrated seamlessly into people's daily lives, which is defined as "Ubiquitous Computing". Nowadays, 25 years after the definition of ubiquitous computing, this change is obviously happening around us. In recent years we have experienced the proliferation of "smart mobile devices", including tab devices such as the iPhone and smartphones running Android OS, and pad devices such as the iPad. It is forecast that the global smartphone shipment will be more than 1.7 billion units by 2018 [3]. The ubiquity of these devices boosts the generation of a large amount of personal data about individuals in our daily lives [88]. With the help of the Internet, these data are collected and used by many parties such as commercial companies [155] and research communities [77], and are shared by individual users with each other.

Like all new technologies, ubiquitous computing has not only brought soaring generation of personal data, but also new challenges. One of these is the privacy risks from overexposing personal data. As in ubiquitous computing environments, personal data contain contextual information, such as location, time, and social network, sensitive personal information could be inferred from them. Thus, inappropriately sharing personal data may be an invasion of privacy. To address this issue, **privacy protection** has always been an important topic in ubiquitous computing since its early stage, such as the importance of making ubiquitous computing environments

privacy-aware [81] and the principles to achieve such awareness [80].

One specific privacy issue in ubiquitous computing is location privacy. As most smart devices can locate themselves through embedded receivers of the Global Positioning System (GPS), people can access their location information through their devices. Cooperating with other information, location information can provide the context of users, and such context can help provide personalised services. Such location-based services (LBS) have become increasingly popular. For example, people now can use map applications such as Google Maps[1] on their mobile devices to search and find nearby places or navigate to their destinations, rather than purchasing the devices specifically designed for these purposes. Popular games such as Pokémon Go[2] introduce location-based features that personalise users' game experience based on their geographical positions. People can share their locations with each other for social needs or receiving rewards. For example, online social network (OSN) platforms such as Facebook[3] and Twitter[4] introduce location information in their services, allowing their users to publish location check-ins in their posts or tweets. Location-sharing applications such as Foursquare/Swarm[5] allow users to public location check-ins to claim badges or discounts. This type of location-sharing services (LSS) requires stronger privacy protection than other LBS, since users share their locations with others rather than keeping it by themselves [11]. Thus, users in LSS have concerns about the potential privacy risks, such as revealing home locations or being stalked [134], which impede their adoption of LSS.

Investigating and addressing the privacy issues in LSS may resolve people's concerns and provide them with better experience when using LSS applications. Apart from commercial applications, the location check-in data generated in LSS are also valuable in many research areas. By analysing these location check-in data, we can have a better understanding about people's activities such as their mobility patterns, about the social relationships between them, and the relations between different locations. These results can help us improve the designs of LBS applications and can also contribute to other areas such as tourism [124] and urban

[1] http://maps.google.com
[2] http://www.pokemongo.com
[3] http://www.facebook.com
[4] http:twitter.com
[5] http://www.swarmapp.com

planning [154]. The study of privacy protection in LSS can also contribute to the privacy protection research in other scenarios that have complicated contexts like LSS do, such as online social networks or mobile operating systems.

The most commonly provided privacy protection mechanism by LSS applications is access control [120]. People manually configure the settings that control with whom they want to share their location information. These mechanisms, such as role-based access control (RBAC) models [119], have been being used since long before the birth of ubiquitous computing and LSS. As the contexts in LSS are more complicated than before, such mechanisms have been argued to be not effective enough to protect people's location privacy. People's location-privacy preferences are context-aware, which means they have different decisions about sharing locations in different contexts. Expressing these location-privacy preferences by using access control mechanism is burdensome. For example, as shown in Figure 1.1, people can only manually configure their location-privacy settings on Facebook every time when they want to publish location check-ins. Thus, users have difficulties in controlling access to their location information in LSS.

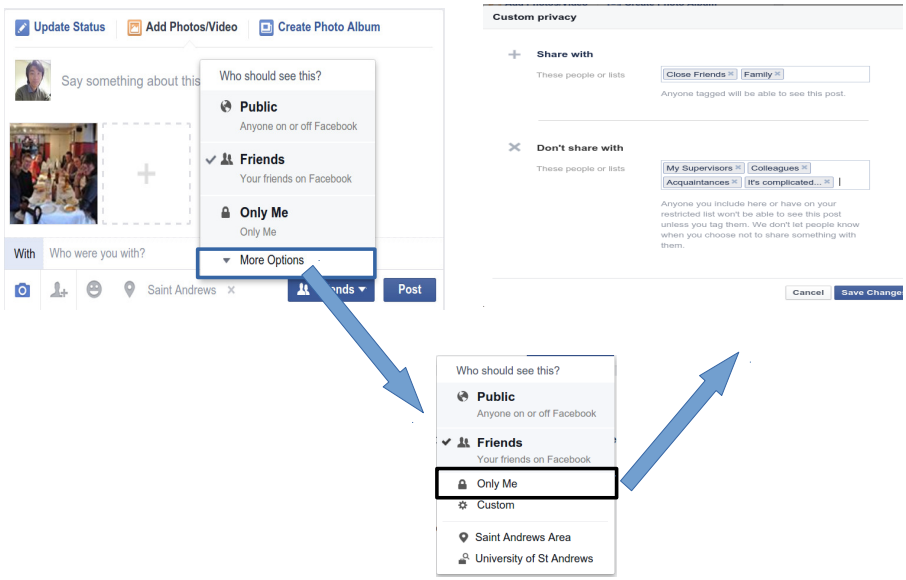


Figure 1.1: Facebook user interfaces to configure location-privacy settings. Users have to manually specify with whom they want to share and not share.

The unwillingness to manually configure location-privacy settings is a typical information overload problem [87]. When being given too many choices, people have difficulty to do it accurately, as their preferences change based on the context of decisions. Such problems have been studied for a long time [40, 52, 87]. One of the commonly used solutions, is using recommender systems to recommend decisions for users. For example, online shopping websites such as Amazon.com recommend products to consumers based on their purchase history, thereby helping them find what they want more quickly. Online video websites such as Netflix[6] and Youtube[7] recommend videos to users who share similar preferences about videos. These recommender systems use their users' preferences in products or videos as crowdsourcing sources, which collect and aggregate the opinions of a large group of people, and make recommendations among the users whose preferences are similar with each other. These applications of recommender systems have achieved success to solve the information overload problems in online shopping and online video browsing. Then, motivated by these successful instances, can we use the same system to solve the same problem in a different area, which is recommending location-privacy settings to people?

Like many other kinds of preferences, location-privacy preferences are complicated. However, like many other kinds of preferences, location-privacy preferences also have similarity. If we can use people's location-privacy preferences as a crowdsourcing source and find the similarity in these preferences, then we can use the data from the users who are similar to each other, in terms of location-privacy preferences, to help each other. If these recommendations are accurate, then we can use these recommendations to configure people's location-privacy settings automatically, thereby alleviating the burden of users to do so.

We therefore propose a location-privacy recommender system using user-based collaborative filtering (CF) to recommend location-privacy decisions and use the recommendations to automatically configure users' location-privacy settings. User-based CF is a technique in recommender systems. It can find people who have similar preferences (e.g., location-privacy preferences in our case) as a crowdsourcing source to make recommendations. If the proposed system works accurately, it may solve the above mentioned problem. Compared with other areas of recommendations, such as movies and musics, location-privacy recommendations are more

[6] http://www.netflix.com
[7] http://www.youtube.com

sensitive. Failed movie recommendations may be ignored by users, whereas failed location-privacy recommendations may cause more consequences. Thus, it is also important to investigate whether people accept location-privacy recommender systems, and what factors can affect their acceptance.

The goal of the proposed location-privacy recommender system is to let people accept the recommendations so that they do not need to manually configure the settings by themselves. Thus, we hope that the recommendations would be as acceptable as possible. If we can find out the factors that affect people's acceptance, then can we revise and improve our design of the system according to our findings, to improve people's acceptance?

This thesis aims to investigate the feasibility of using recommender system to help people with their location privacy and to understand their acceptance of such recommender system. We examine the following questions:

- **Q1** Can recommender systems provide accurate location-privacy recommendations?
- **Q2** What factors affect people's acceptance of location-privacy recommender systems?
- **Q3** How can we modify the design of location-privacy recommender systems to make them more acceptable and robust against malicious data attacks?

1.1 Thesis

We have proposed that recommender systems using crowdsourcing data may be potentially capable to recommend location-privacy settings and be accepted by people. Therefore we offer the following thesis:

User-based CF recommender systems can help people manage location-sharing in an accurate, acceptable, and robust way.

1.2 Goals and approach

The main goal of this thesis is to demonstrate that recommender systems can be used to help people with their location privacy in LSS. To do so, we test our proposed location-privacy recommender on the location-privacy preference data collected from the real world. Although online crowdsourcing platforms such as Amazon Mechanical Turk (MTurk[8]) provide an easier way to collect data with larger sample sizes, we believe that data collected from the real world have better quality than data collected from online platforms.

We conduct a series of experiments, including offline evaluation, online user studies, and offline simulation, to examine our research questions.

To answer Q1, we implement our location-privacy recommender using user-based CF and evaluate its performance on the data through offline experiments. Compared with evaluating the performance in situ, offline evaluation allows us to conduct multiple repeated experiments with different evaluation strategies. We split the data in two different ways to simulate two different application scenarios in LSS, which are the performance with enough data and the performance with insufficient data.

To answer Q2, we conduct on-line user studies to investigate what factors can affect people's acceptance of our location-privacy recommenders when they use them. We evaluate the effects of the factors from both the users' side and the recommenders' side. With the help of a user-centric evaluation framework, we can find out what factors affect users' acceptance of location-privacy recommenders, under the condition of unchanged recommendation accuracy.

To answer Q3, we revise the design of our system by using data obfuscation and decentralisation, and compare its performance with that of the old one through simulation. We also simulate specific scenarios to evaluate the robustness of the revised system.

[8] http://www.mturk.com

1.3 Outline of dissertation

In Chapter 2 and 3, we introduce the background of this work and the state-of-the-art in the area that we focus on.

- In Chapter 2, we introduce the evolution of location-based services (LBS) and location-sharing servers (LSS). We introduce some LSS applications, both commercially and academically. We also examine people's motivations of using LSS and the different types of privacy risks that hinder their adoption.

- In Chapter 3, we discuss the related work in the area of people's location privacy preferences and privacy-enhanced technologies in LSS. We highlight the research questions that this thesis aims to answer and justify our decision.

In Chapters 4, 5, and 6, we discuss our experiments, results, and analysis.

- In Chapter 4, we demonstrate that user-based CF recommenders can provide accurate location-privacy recommendations. In addition, it has better performance with insufficient data, compared with model-based classifiers.

- In Chapter 5, we evaluate people's acceptance of location-privacy recommenders. By conducting online user studies, we demonstrate that people have privacy concerns about providing their data to a centralised location-privacy recommender server and such concerns decrease their acceptance of the system.

- In Chapter 6, we demonstrate that location-privacy recommenders have the potential to be implemented in a privacy-aware fashion, or to be implemented without centralised servers. Such decentralised recommenders have good performance and are robust against the attack from malicious users.

Finally, in Chapter 7, we conclude this thesis by summarising our contributions and discuss possible future research topics.

1.4 Publications

During the course of my PhD, I have published several peer-reviewed publications. In this publications, the major work including experimental design, implementation, and data analysis, are my own. I acknowledge the contributions and guidance of my supervisors in these works.

- Yuchen Zhao, Juan Ye, and Tristan Henderson. **Recommending Location Privacy Preferences in Ubiquitous Computing**. In *Proceedings of the 7th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '14)* , Oxford, UK, July 2014. Poster paper. This paper contributes to Chapter 4.

- Yuchen Zhao, Juan Ye, and Tristan Henderson. **Privacy-aware Location Privacy Preference Recommendations**. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous '14)* , page 120-129, London, UK, December 2014. doi: 10.4108/icst.mobiquitous.2014.258017. This paper contributes to Chapter 4 and Chapter 6.

- Yuchen Zhao. **Usable Privacy in Location-Sharing Services**. In *Proceedings of the Companion Publication of the 21st International Conference on Intelligent User Interface (IUI '16 Companion)* , page 110-113, Sonoma, CA, USA, March 2016. doi: 10.1145/2876456.2876458.

- Yuchen Zhao, Juan Ye, and Tristan Henderson. **The Effect of Privacy Concerns on Privacy Recommenders**. In *Proceedings of the 21st International Conference on Intelligent User Interface (IUI '16)*, page 218-227, Sonoma, CA, USA, March 2016. doi: 10.1145/2856767.2856771. This paper contributes to Chapter 5.

- Yuchen Zhao, Juan Ye, and Tristan Henderson. **A Robust Reputation-based Location-privacy Recommender System using Opportunistic Networks**. In *Proceedings of the 8th EAI International Conference on Mobile Computing, Applications and Services (MobiCASE '16)*, Cambridge, UK, November 2016. doi:10.4108/eai.30-11-2016.2267031. This paper contributes to Chapter 6.

Chapter 2

Location-sharing services and location privacy

In this chapter, we discuss how LSS enter into people's daily lives, how people benefit from using LSS, and what kind of negative implications LSS have brought. First, we introduce the history of LSS in the context of LBS, which is the superset that contains not only LSS, but also other types of services. We then look at why people use LSS and how they can benefit from using them. We finally discuss the negative implications, i.e., the privacy risks, of using LSS.

2.1 Location-sharing services

In this section, we discuss the origin and development of LBS and how LSS have originated with such development. The commercial applications and the research value of LSS are also discussed.

2.1.1 Location-based services

In 1996, the Federal Communications Commission passed the E911 mandate. According to the mandate, mobile-network providers are required to be able to locate 911 emergency callers. The mandate drove the early usage of location information in the real world. In fact, before the mandate, research communities had already considered to introduce location information into

ubiquitous computing environments [126, 140], such as the Active Badge [138] system.

As the technologies that support LBS became mature, the definitions of LBS gradually became clear. As described by Brimicombe [21], from the perspective of systems, LBS are the intersection of several technologies, including geographic information systems (GIS), Internet, and new information and communication technologies (NICTs). The NICTs are specifically defined as mobile devices and wireless devices by Zipf and Jöst [160]. From the perspective of users [161], LBS are "services for mobile users that take the current position of the user into account when performing their task".

From the perspective of systems, as shown in Figure 2.1, LBS are supported by three components, i.e., mobile devices hold by users, Internet that allows data to be transmitted, and GIS that locate users. The mobile devices can be mobile phones, tablets, personal navigational assistant (PNA), and so forth. These devices are can be located through various techniques in GIS, such as GPS. Users' location information is transmitted through communication networks such as cellular networks and wireless local area networks to enable LBS.

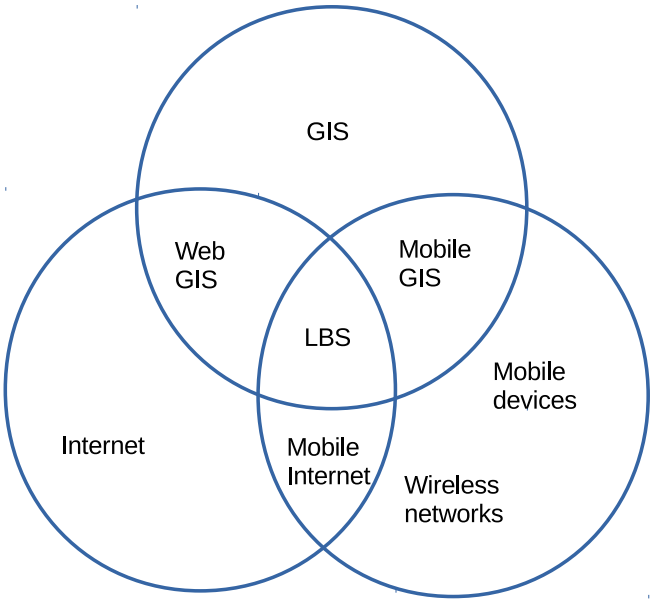


Figure 2.1: LBS from a system-oriented view (from [160]).

As discussed by Bellavista *et al.* [11], the evolution of LBS proceeds with constantly emerging

new technologies. In the early stage of LBS, people's location information was mainly used as a condition to refine or personalise search results. As smart phones become more and more popular, more sophisticated services, such as navigation, are supported. With the proliferation of OSN, people are able to share their location information through OSN platforms with each other. In the following subsections, we discuss the applications in the areas of location-based search, navigation, and location-sharing services, respectively.

2.1.2 Location-based search

Location-based search is one of the major applications in LBS. By using location information, people can find the points of interest (POI) around their geographical locations. For example, the CRUMPET system [124] uses tourists' location information to recommend tourist attractions to them. Their tourism experience can be context-aware with the help of location information. For service providers, location information can also be beneficial. One example is location-based advertising, which personalises advertisements according to target users' locations [34, 73, 82].

In location-based search, location information is used in the same way in which other types of information are used, providing restrictions for search results. The geographical relations between different locations are not exploited. As more and more location data are generated, navigation services based on geographical relations become available.

2.1.3 Navigation

In LBS, navigation applications can help users find the routes from their current locations to their destinations. In early LBS, navigation was mainly provided by specialised devices such as portable navigation devices (PND). With the proliferation of smartphones and the maturation of their functions, commercial navigation services, such as Google Maps[1], have become more and more widely used. This trend leads to the generation of abundant trajectory data, which has inspired many research topics such as mobility prediction [35, 121], trajectory data mining [153], and urban computing [154].

[1] http://maps.google.com

In navigation services, the relations between different locations are considered. However, the location information in navigation is still self-referencing [11], which means that people request their location information and then use it by themselves. With the help of Web 2.0, which enabled user-generated content, and the help of OSN platforms, such as Facebook[2], Twitter[3], and Foursquare[4], which allowed users to share these content with each other, LSS entered people's lives.

2.1.4 Location-sharing services

With the increasing adoption of smartphones [3] and the proliferation of OSN [102], people become able to publish location check-ins and share them with each other. This is known as location-based social networks (LBSN) [128] wherein people can share their locations through LSS.

In 2003, the Dodgeball company was founded by Dennis Crowley and Alex Rainert. The Dodgeball application [56, 96] allowed people to send text messages about their locations to Dodgeball servers and the servers then sent these text messages to the users who were in the sender's Dodgeball network. The location information was reported by users themselves rather than using GPS. In addition, the communication of the service was through short message service (SMS) rather than mobile data networks.

In 2007, the location-based social network site Gowalla[5] was launched. It allowed users to check in at locations and share these check-ins through their Facebook or Twitter accounts. In addition, it provided bonus to users for their location check-ins as incentives.

In 2009, Dennis Crowley and Naveen Selvadurai launched Foursquare. Its early version was a mobile application that provided both location-based searching and LSS. People could use Foursquare to receive recommendations based on their current locations and could publish location check-ins to others. Since 2014, the location-sharing features have been moved to

[2]http://www.facebook.com
[3]http://twitter.com
[4]http://foursqaure.com
[5]http://gowalla.com

Swarm[6] (Figure 2.2), which is a companion application of Foursquare. Google also launched Google Latitude[7] in 2009, after they closed Dodgeball. Google Latitude users could share their current locations to certain groups of people and they could also control the granularity of the shared locations.

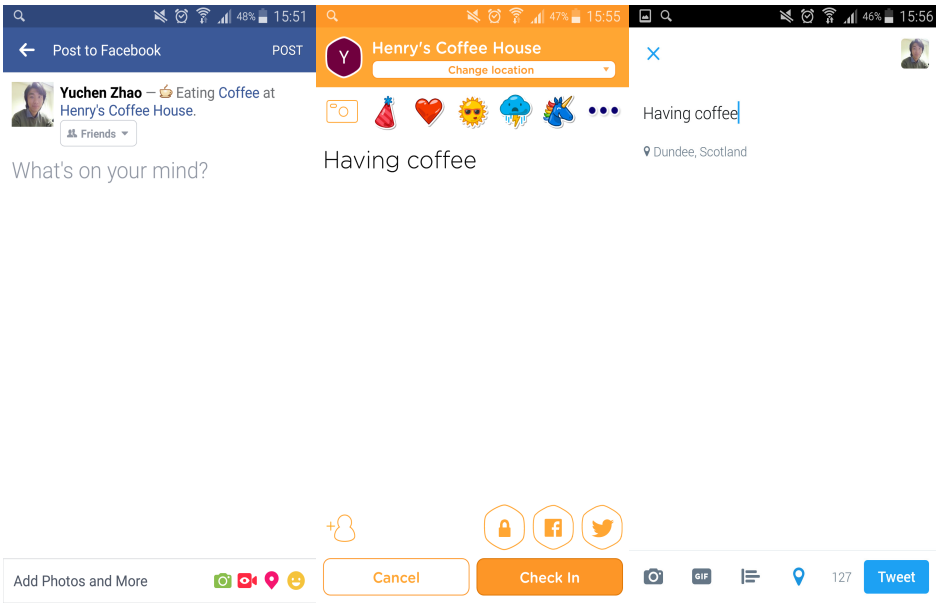


Figure 2.2: LSS applications on Android platform. From left to right: Facebook, Foursquare Swarm, and Twitter. Users can share their locations with their social networks and attach their location information to photos, videos, and activities in their posts.

Apart from the above mentioned applications that dedicate to LSS, many online social network websites also allow their users to associate their posts with location information. For example, as shown in Figure 2.2, Facebook users can associate their posts with their locations. Twitter and Instagram also allow geotagged tweets and photos. People can use these functions to share interesting locations with their friends in their social network, or to find nearby friends.

The widespread use of commercial LSS applications boosts the amount of location check-in data generated in the real world. According to Foursquare, they have more than 10 billion overall check-ins and the average daily check-ins on Swarm are 9 million [1]. This tremendous amount

[6]http://www.swarmapp.com
[7]http://latitude.google.com

of location check-in data have inspired many research areas that can help us understand people's activity patterns [98].

Mining LSS users' location check-in histories can help us understand their activities. Ye *et al*. [147] propose a framework to mine individual life patterns from people's GPS data. Their results show that the predictability of life patterns is relatively high, although some of these patterns are trivial as it is the nature of frequent life styles. In addition, their results show many private patterns from the GPS data, which suggests that there are potential privacy risks from sharing locations. Cho *et al*. [28] investigate people's activity in terms of the predictability of future locations. By analysing the data from location-based social networks, they find that people's mobility shows periodicity and is within a bounded region. These features make their mobility predictable. Meanwhile, people's social relationships, i.e., friendships, are correlated with their mobility, which means considering their friends' mobility can improve the accuracy of prediction. Lian and Xie [83] investigate naming patterns in people's location check-in histories. They find that, in LSS, the naming of locations is mainly decided by four features in the check-in data. By using these features, their model can automatically name locations for users with good accuracy.

Apart from people's activity, research has also been conducted to mine social relationships from people's location check-ins. For example, Hung *et al*. [57] propose a framework to detect communities among users based on the similarity of their trajectories. Their results show that such framework can successfully detect communities among users. Similarly, Chang and Sun [26] also find that check-in counts and co-check-ins are strong indicators of friendships among users. People's location check-in histories can also indicate their similarities. Using category-based location histories [143] or semantic trajectories [144, 149], we can find the users who are similar with each other and can be potential friends. Thus, taking one step further, geo-friends recommendation [150, 156] or prediction [122] based on potential social ties is possible.

Another important research area is to mine location features based on people's location check-ins. Zheng *et al*. [158] propose a model to infer the interests of locations by using the location check-ins from people who have visited those locations[157]. Based on the mining results, the

model also provides classical travel routes among those locations. Their results show that the mining results from people's location trajectories are better than the results from rank-by-count or rank-by-frequency methods. Data mining can also be applied to constructing popular trip routes from uncertain location check-in histories [139]. In addition, by analysing geo-associated documents, such as tweets with location information, we can relate locations with different topics, which helps us find the region of interests [148]. It can also benefit many other applications [151] such as urban planning and location-based social recommendation.

2.2 Motivations of using location-sharing services

We have already talked about how LSS entered into people's lives with the popularity of OSN platforms that allow people to share their location information with each other. To understand why people use LSS and how they can benefit from using them, it is necessary to investigate their motivations.

As LBSN are a subset of OSN, some motivations of using LSS are social-driven [129]. Lindqvist *et al*. [84] conduct user studies to investigate why and how people use LSS. The reasons of using LSS in their results include personal tracking, intimate sharing at a distance, discovery of new people, running into friends, gaming aspect, seeing where friends have been, and recording non-routine places. In addition, some people use LSS as a tool to tell their friends that they have arrived their destinations safely. They find that apart from social-driven motivations, people also check in for fun, such as claiming badges or mayorship of their houses. Similar findings are proved by Cramer *et al*. [30]. Patil *et al*. [105] summarise the main motivations for sharing locations as connecting with people's social circles, to projecting an interesting personal image, and receiving rewards, which also suggests that some motivations of using LSS are social-driven.

From individual's perspective, people use LSS to fulfil their social needs or to receive rewards. From the perspective of scientific research, location check-in data produced in LSS help our understanding of people's activity and social relationships, and the relations between different locations. Thus, we expect more and more location check-in data generated from LSS to contribute to our knowledge in these research areas. People's motivation of using LSS, however,

is hindered by their concerns about privacy [105]. Therefore, we need to understand the privacy risks of using LSS.

2.3 Privacy in location-sharing services

We have discussed the motivations and benefits of using LSS. When using LSS, people's location disclosure is not always within their expectation. There are many reported cases about people accidentally revealing their whereabouts through online social media [2, 5] or geotagged photos [4, 6]. This kind of unexpected location disclosure is one of the negative implications of LSS, which affects people's adoption of LSS. Thus, it is important to understand what concerns people have and how these issues discourage people from using LSS.

Location privacy, as described by Beresford and Stajano [13], is "the ability to prevent other parties from learning one's current or past location". As LSS are cross-referencing LBS [11], which means that people share their location information to others, the "other parties" in LSS refer to people who can see the location information of the publishers.

The privacy issues in LSS have been one important factor that affects people's adoption since the early stage of LSS [138]. People need the rights to control who can see their location information at what places during what time. Otherwise, inappropriately shared location information may reveal other sensitive information such as health conditions and political inclinations [46]. In addition, there are computational threats [78] such as early analysis of movement patterns and context inference. These threats increase people's privacy concerns [10] about LSS. Note that these privacy risks are theoretically similar to the above mentioned data mining on people's location check-in data. Although they both are conducted on people's public data, scientific research are supposed to be under ethical restrictions [58] and have people's informed consents [59].

Tsai *et al*. [134] evaluate people's perceived privacy risks when using LSS. The privacy risks revealed in their results can be categorised into two types. One class is the invasion of boundary, such as being stalked, exposing home locations to people who are not supposed to know them, being found by others you do not want to see or when you want to be alone. Another one is being judged because of revealed locations. The first type of risks, which is the boundary

preservation concern (BPC), is also proved by Page *et al*. [100], which indicates the importance of boundary preservation in LSS. Another type of risks is from the sensitivity of locations. Toch *et al*. [131] show that people perceive location sensitivities based on the diversity of people who visit locations, and their desire to share vary based on the sensitivities. Staiano *et al*. [127] also show that people consider their location information as the most sensitive and valued information among all their mobile data. There are also other types of location privacy risks [117] such as *absence privacy*, which means that people's check-ins at some locations indicate their absence at other locations, and *co-location privacy*, which means that from location check-ins, people's co-presence at a location may be inferred and such event may be sensitive. The violation of absence privacy may lead to the danger of home invasion. For example, as shown in Figure 2.3, the Please Rob Me website[8] used to allow people check whether a Twitter or Foursquare user is at home by analysing the user's public location check-ins. The violation of co-location privacy may cause over-exposure of social relationships [76].

Figure 2.3: "Please Rob Me": A website that provided information about whether people were at home, based on their location check-ins on Twitter and Foursquare.

[8] http://pleaserobme.com

Given the privacy risks and concerns that people have in LSS and the sensitivity of location information, their adoption of LSS is impeded. One way to protect location privacy is through *anonymity*, such as cloaking location information spatially and temporally [14, 29, 46, 90]. By this means, people's locations are hidden in an area, which can be used to access LBS or to be shared with others. Location anonymity protects people's location privacy by controlling to what extent that people share their locations. Its premise is that people can appropriately decide with whom, when, and where they want to share their locations. In LSS, the commonly used mechanism is to let users control their location-privacy policies [132, 134] by themselves. They can decide at what place (or types of place), at what time, with whom, they want to share their locations. Whether the control mechanisms are easy to use, however, is to be tested.

Existing research has shown the limit in the usability aspects of current privacy protection mechanisms [63]. On the one hand, privacy settings need to be expressive so that they can describe people's privacy preferences accurately. On the other hand, the needed expressiveness increases the number of privacy settings and the burden to configure them [43], which causes the usability issues in privacy protection. Such usability issues, which broadly exist in different areas, including OSN [93, 112] and LSS [118], essentially is a type of information overload, which has been widely studied and mitigated with the help of recommender systems [114].

2.4 Recommender systems

Recommender systems are "software tools and techniques that provide suggestions for items that are mostly of interest to a particular user" [114]. The recommended items in recommender systems are decided by the scenario wherein the recommender systems are applied. For example, Amazon.com uses recommender systems to recommend different kinds of products sold on it. Last.fm[9] recommends music, videos, and photos to its users. Facebook and Twitter recommend friends, interesting topics, and news to their users. The primary purpose of using these recommender systems is to free people from the overwhelming amount of online information that they face and to help them find the items that they may be interested in.

The techniques in recommender systems can be categorised as content-based recommender

[9] http://www.last.fm

systems [36], CF recommender systems [97], and hybrid recommender systems [22]. The CF recommender systems, as the most popular and widely used technique [114], recommend items based on the similarities between the users of the recommender system or the similarities between items. Many commercial companies use them to recommend products to their users. For example, as shown in Figure 2.4, Amazon.com provides recommendations from the "Customers Who Bought This Item Also Bought", which is a typical use of CF recommender systems. Netflix also recommends videos to its users based on what they have watched before. The wide adoption of CF recommender systems has shown their ability to solve the information overload problem in many different areas. Thus, it is worth to study whether they can solve the similar problem in a new area, which is privacy protection.

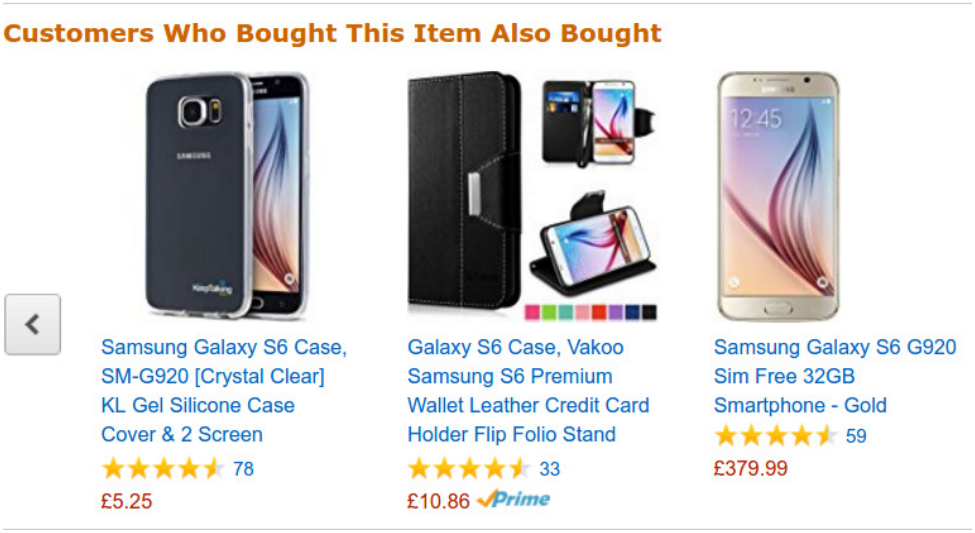


Figure 2.4: Products recommended by Amazon.com, when customers buy smartphones.

2.5 Discussion

LSS are gaining more and more popularity, with the proliferation of smart phones and online social networks. People share their location information for personal tracking, discovery of new people, receiving rewards, and so on. The location check-ins generated from LSS give us better understanding about the activity of people, the social relationships of them, and the relations between locations, thereby helping us improve the design of LBS applications. Both individual users and researchers can benefit from LSS. Therefore, conducting research on LSS

can give us deeper insights about how people use LSS and what issues they have when using them. Answering these questions can help us find the direction to increase the adoption of LSS.

The privacy risks in LSS are known as one of the biggest obstacles for people to accept LSS since their origin. Inappropriate disclosure of locations violates people's social boundaries and overexposes private locations such as home addresses. It may also reveal sensitive information about users, and cause absence privacy or co-location privacy issues, which embarrasses users or reveal the relationships that they do not want others to know. Physical risks such as stalking are also possible from the overexposure of locations. Existing evidence shows that these privacy issues jeopardise people's adoption of LSS. Therefore, we choose the privacy aspects rather than others of LSS as our research area, since we believe it is a vital issue.

As mentioned above, one way to address the privacy risks in LSS is to let users control their location disclosure. Many LSS applications and OSN platforms provide access control mechanisms. Users can manually configure these location-privacy settings. Such mechanisms, however, have been used since long before LSS was born and cannot provide effective protection in LSS. Thus, in this thesis, we investigate new solutions that protect location privacy in LSS more effectively.

2.6 Summary

In this chapter, I have noted the following points:

- LSS have grown in use with the increasing popularity of smartphones and OSN.
- LSS have brought convenience and fun in our social lives.
- Location check-ins generated in LSS have boosted several research areas.
- There are privacy issues such as invasion of social boundary and exposure of sensitive information in LSS.
- People have concerns about these privacy risks when using LSS.
- Existing access control mechanisms have usability issues when protecting location privacy due to the information overload in privacy configuration.

- Recommender systems have been used in many different areas to solve information overload issues.

In the next chapter, we discuss how effectively can people use current location-privacy protection mechanisms and the work that aim to help people protect their location privacy more effectively.

Chapter 3

Privacy-enhancing technologies in location-sharing services

In Chapter 2, we discussed the development of LSS and their applications in our lives. As LSS are becoming increasingly popular and meanwhile they have brought privacy issues to us, it is necessary to study the reasons that cause those privacy issues. By doing this, we can improve our design of LSS applications to protect people's location privacy, thereby making LSS more acceptable. One of the reasons behind the privacy issues in LSS is the usability problem, which means people find the existing mechanisms difficult to use. Thus, we focus on the area of privacy-enhancing technologies (PET) that help people with their privacy policies in LSS.

In this chapter, we discuss related work in the areas of the usability issues in privacy protection and the PET that address those issues. Then we introduce our research focus in the context of the related work and explain why we choose it.

3.1 Related work

3.1.1 Privacy preferences

To understand people's privacy preferences in different areas such as OSN and mobile applications, many user studies have been conducted. Liu *et al*. [85] use an online survey to investigate the difference between the desired privacy settings and the actual privacy settings of Facebook

users. In their results from 200 Facebook users, actual settings match users' desired settings only 37% of the time. This result means that current privacy mechanisms, which ask users to manually configure access control rules, are far from satisfying people's privacy preferences. Similarly, Madejski *et al*. [86] conduct user studies to find the difference between Facebook users' sharing intentions and their actual privacy settings. Their results also show that, for each of the 65 participants in their studies, there is at least one violation in their privacy settings. This result also suggests that people have difficulties to achieve their ideal privacy policies by using existing privacy protection mechanisms.

The reasons behind the failure of existing privacy protection mechanisms are various. Furnell [43] suggests that one important reason is the increasing number of privacy rules, which makes manual privacy configuration extremely challenging for normal users. As also shown in the results of the study conducted by Knijnenburg *et al*. [69], people's information disclosure behaviours are affected by several dimensions. For instance, they have different disclosure intentions for different types of information. Meanwhile, the disclosure intentions of different people may also vary. This result means that we should provide dynamic privacy policies to satisfy different types of information and users, rather than using predefined uniform settings. More privacy policies, however, lead to more decisions to make and more burden for users. In the user study conducted by Zheng *et al*. [152], many participants express that they have lost control to configure their privacy policies and find the privacy control settings on Facebook difficult to use. Similarly, Korff and Bohme [75] show that when being provided with more privacy choices, people have more negative feelings. Such dilemma between people's dynamic privacy need and their inability to configure privacy policies is also mentioned by Vihavainen *et al*. [136]. In addition, they suggest that it is important to provide feedback to users in privacy protection. This suggestion is similar to the concept of "informed choice" mentioned by Furnell [43], who suggests to let users know the extent of their data sharing and the implications of sharing. The user study conducted by Shih *et al*. [125] also shows that people are more cautious about their information disclosure when they are informed.

Location-privacy preferences, as a subset of people's privacy preferences, have the same features mentioned above. People's location-privacy preferences are also affected by different dimensions.

Anthony *et al*. [9] show that people's location-privacy preferences are dynamic and can be categorised into different groups. Their location-privacy preferences are different across different context (e.g., location). Benisch *et al*. [12] also show location's effects on people's location-privacy preferences. They also show that time dimensions, including time-of-day and day-of-week, also affect people's location-privacy preferences. Thus, people need complex privacy settings [64] to express their location-privacy preferences. This need, however, also brings usability issues. Like other types of privacy preferences, people also find it difficult to manually configure effective settings that match their location-privacy preferences [118]. The reasons include the difficulty to pre-define all the settings in one go, the burden of too much configuration, and the lack of incentives to do so [104].

Research has revealed the gap between the privacy protection that current mechanisms can provide and the actual privacy demand of users. As discussed above, users are not informed about the disclosure of their information. Thus, they cannot react correspondingly to change their privacy settings. Even if they are willing to do so, the burden of configuring complex privacy policies makes them unable to achieve their ideal privacy preferences. Therefore, to address these issues, we can use feedback mechanisms to inform people about their information disclosure, or (and) help them configure privacy policies by reducing the burden of doing so.

3.1.2 Feedback in privacy protection

To make OSN users more aware of the disclosure of their information, feedback mechanisms have been introduced in privacy protection. Tsai *et al*. [133] investigate the effect of using information request histories as feedback in LSS scenarios. They find that, when users being provided the histories of requests for their locations, their privacy concerns become lower and their comfort levels become higher, compared with those without feedback. Schlegel *et al*. [123] also use the histories of location requests as feedback. In addition, they quantify the exposure of people's location information by using visual "eyes" on mobile devices' screens, which makes the feedback more intuitive. Their results suggest that such visual metaphor interfaces are more effective than detailed access histories. Similarly, Hoyle *et al*. [54] also propose a visual feedback interface to inform people about their information disclosure. Almuhimedi *et al*. [8] propose

a permission manager that informs people about the frequency of their location information disclosure. Their results show that with such permission managers, people spend more time to reassess the permissions that allow different mobile applications to access their location information.

Apart from information disclosure, other information can also be as useful as feedback. Venkatanathan *et al*. [135] investigate the effect of location histories and find that people can configure consistent and reliable location-privacy preferences if they are informed about their location histories. Knijnenburg and Kobsa [68] show that, justifications, which are the benefits of information disclosure, can affect people's actual information disclosure behaviours and can be used as feedback to increase people's satisfaction. Harbach *et al*. [49] use personalised examples to show the privacy risks of information disclosure as feedback, to help people make their decisions. Their results show that people's decisions are more privacy-conscious when being provided such feedback. Their results correspond to the work of Fu *et al*. [42], wherein people are more likely to take actions to protect their location privacy if they are provided run-time feedback.

Although feedback mechanisms can make people aware of their information disclosure and can make them control their privacy policies [106], it is still cumbersome for normal users to manually do so. Thus, privacy recommenders have been proposed to help people configure their privacy settings.

3.1.3 Privacy recommender

As people's privacy preferences are influenced by different dimensions, the combinations of these dimensions can be used as contexts wherein we recommend different privacy preferences accordingly. For example, Danezis [33] uses people's social networks as contexts and infers privacy policies in different contexts. Jin *et al*. [61] use people's activity at different locations as predictors for their location-sharing decisions. Similarly, Pallapa *et al*. [101] use people's behaviour in smart environments as predictors to automate their information disclosure, thereby reducing their burden. Miettinen *et al*. [91] use sensor readings on people's smartphones, including GPS, WiFi, and Bluetooth, to construct their location context and social context, and to make access to people's information context-aware. Dong *et al*. [37] extract several context

elements from people's OSN data, and use them as predictors for people's privacy preferences.

Although people's privacy preferences are related with contexts, it is not easy to find accurate relations to map context elements to privacy policies. As more and more sophisticated machine-learning techniques have emerged in recent years, many researchers have proposed to use machine-learning classifiers to build models from individual's privacy preference histories, and use such models to recommend privacy decisions to users. For example, Ravichandran *et al*. [112] use decision tree classifiers and unsupervised cluster algorithms to learn a set of default location-privacy settings. Such "smart default" can alleviate people's burden of manual configuration. Similarly, "smart default" based on many other algorithms [17, 94, 118] have also been proposed, and have shown higher accuracy than static policies to match people's actual privacy preferences.

Naini *et al*. [95] compare the performance of different machine-learning classifiers when recommending privacy preferences. Their results show that classifiers based-on decision tree have higher recommendation accuracy than classifiers based-on Naïve Bayes. Similar results have also been demonstrated by Bigwood *et al*. [16]. Their results suggest that the Rotation Forest, which is a type of ensemble learning method, has higher recommendation accuracy than both decision tree and Naïve Bayes do. In addition, they introduce a new metric in the evaluation of privacy recommenders, which is the privacy leak caused by failed recommendations.

To improve the performance of privacy recommenders, user-controllable learning schemes have been proposed to enable users to control and interact with their privacy models. For example, Kelley *et al*. [65] propose a user-controllable policy learning model. In their scheme, systems build models from users' privacy policies and users give feedback to systems to incrementally improve the models. Their results show that such models can provide higher accuracy than manual policy configuration does. Fang and LeFevre [41] propose a privacy wizard that learns people's privacy preferences on OSN. Their scheme also allows users to actively give feedback to their models to achieve better recommendation accuracy. Cranshaw *et al*. [31] also propose a user-controllable Gaussian mixture model that learns users' location-privacy preferences incrementally.

Since existing research [9, 69] has shown that some people's privacy preferences are similar

to each other, privacy recommenders that use crowdsourcing data rather than individual's data have also been proposed. Henne *et al*. [51] discuss the possibility of using online communities to let people support each other with their location-privacy decisions. Toch proposes a Super-Ego crowdsourcing framework [130] that provides location-privacy recommendations based on location semantics. The results show that cooperating personal bias with location semantics can improve recommendation accuracy. User-based CF [113] is a widely used recommender technique that cooperates personal bias with crowdsourcing knowledge. Xie *et al*. [145] combine user-based CF and item-based CF to make location-privacy recommendations. The performance of such ensemble recommender is better than those of other CF recommenders. Ismail *et al*. [60] also apply user-based CF to recommending permissions for mobile applications.

3.1.4 Attacks against recommender systems

CF based recommender systems make recommendations from users' input. As the structure of recommender systems is open, everyone can contribute their ratings to the systems. As a consequence, CF based recommender systems are vulnerable to those malicious users who want to bias the results of recommendation for their own benefits. These malicious users create many fake profiles and use these profiles to inject biased ratings to the systems. This type of attacks is known as *shilling attacks* [79].

Shilling attacks can be categorised as *low-knowledge* attacks and *high-knowledge* attacks [47], depending on attacker's (i.e., malicious users) knowledge about target recommender systems. Many schemes [15, 24, 27, 142] that detect low-knowledge attacks have been proposed. One particular high-knowledge attack, the *sampling attack* [23], has received little attention, because attackers are considered incapable of conducting the attack in centralised recommender systems.

3.1.5 Discussion

LSS are becoming increasingly popular. Therefore, addressing the above mentioned privacy issues in LSS is inevitable. As described in the related work, people have difficulties to use existing mechanisms to properly protect their location privacy. Feedback mechanisms and location privacy recommenders can increase people's awareness of their location privacy and

reduce their burden of configuring location-privacy policies, respectively. Location-privacy recommenders based on individual's data have attracted much attention from researchers, but crowdsourcing location-privacy recommenders have not. The evaluation of location-privacy recommenders has only considered recommendation accuracy, whereas people's acceptance of location-privacy recommenders has not been investigated. There have been many schemes to improve the performance of location-privacy recommenders, but no suggestions from users' perspective for improvement.

Hence, we propose several research areas to be examined:

- Performance of location-privacy recommenders using crowdsourcing data
- Evaluation of location-privacy recommenders from users' perspective
- Influence of improved system design on the performance of recommenders

3.2 Summary

In this chapter, we have noted the following points:

- Existing research has shown that people have difficulties to protect their location privacy.
- PET such as feedback mechanisms and privacy recommenders based on individual's data have been proposed to help.
- The comparison between location-privacy recomenders based on individual's data and crowdsourcing data has not been examined.
- People's acceptance of location-privacy recommenders has not been examined.

In the next chapter, we will examine the performance of user-based CF location-privacy recommenders based on crowdsourcing data, and compare it with the state-of-the-art in different circumstances.

Chapter 4

A location-privacy recommender based on collaborative filtering

4.1 Introduction

In Chapters 2 and 3, we have discussed the privacy risks caused by inappropriate location disclosure and the difficulties that people have when manually configuring their location-privacy settings. To alleviate people's burden of doing this, many machine-learning techniques have been applied to predicting people's location-privacy preferences, thereby automatically configuring their location-privacy settings. These techniques build models from existing data and use the models to make recommendations. Compared with these model-based recommenders, another commonly used technique of recommender systems is to make recommendations from neighbourhoods. Therefore, we speculate that neighbourhood-based recommenders can also be used to recommend location-privacy settings.

In this chapter, we demonstrate the recommendation performance of neighbourhood-based location-privacy recommenders. We test these recommenders on location-privacy preference data collected from the real world and compare their performance with those of model-based recommenders.

The purpose of these experiments is to answer the following questions:

- **Q**1 Can neighbourhood-based location-privacy recommenders perform as well as those model-based location-privacy recommenders do?

- **Q**2 Can neighbourhood-based location-privacy recommenders outperform model-based location-privacy recommenders when having insufficient data?

As we have already discussed, there is similarity in people's privacy preferences. Therefore, neighbourhood-based recommendations may be as accurate as model-based recommendations. Furthermore, neighbourhood-based recommendations are made from crowdsourcing sources, which means that their performance may be better than that of model-based recommenders when the data of individual users are insufficient.

4.2 Methodology

4.2.1 Model-based recommendations

One type of recommender is learning from people's location-privacy preferences and building models to describe the relations between independent variables (e.g., user, location category, time slot, recipient) and dependent variables, i.e., people's location-privacy decisions. In other words, such models are functions that take independent variables as input and generate dependent variable as output accordingly.

Although model-based location-privacy recommenders have good accuracy, they have several drawbacks. First, once a new user joins the system or new data are produced by existing users, the models of users need to be updated to fit the latest data. To keep the models up-to-date, the system needs to re-train the models periodically, to ensure that recommendations are accurate. This process is computationally costly in large commercial applications [97]. Second, for an individual user, when there are insufficient data to train his or her model, the recommendation results may be inaccurate. This may especially affect the recommendation quality for those newly joined users.

To provide accurate location-privacy recommendations and overcome the drawbacks that model-based recommenders have, we used neighbourhood-based recommenders, which are widely used

in commercial recommender systems and do not need costly training process. Specifically, we chose user-based CF, because it can use the similarity in people's location-privacy preferences.

4.2.2 User-based CF

User-based CF [113] is a type of neighbourhood-based CF technique in recommender systems. The purpose of recommender systems is to recommend *items* to *users*, thereby alleviating information overload issues. In user-based CF, the relations between users and items are represented as a user-item matrix. Each row represents a user and each column represents an item. Accordingly, each cell in the matrix represents a rating given by a user to an item. Table 4.1 shows an example of a user-item matrix that has 5 users and 4 items.

User	$item_1$	$item_2$	$item_3$	$item_4$
$user_1$	3	4	5	1
$user_2$	1	2	1	4
$user_3$	NULL	4	5	2
$user_4$	4	5	5	1
$user_5$	2	2	2	5

Table 4.1: An example of user-item matrix for 5 users and 4 items. Users' ratings to items are from 1 to 5, or unrated (NULL).

From a user's perspective, the row of the user is a rating vector that describes its preference for different items. By comparing different users' rating vectors, we can find out which of them are similar to each other. These similar users are called *neighbours* in user-based CF. The assumption behind user-based CF is that people who have similar preferences on some items may also have similar preferences on other items. Therefore, if a user requires a recommendation for an item, we can find its neighbours by comparing their preferences and then use these neighbours' opinions to make a recommendation.

As described above, we can see that user-based CF has several advantages. First, since it does not attempt to calculate an estimated function of the relations between independent variables and dependent variables, it does not need to train models periodically like model-based systems do. Second, since recommendations in user-based CF are made from similar neighbours, i.e., in a crowdsourcing way, even if the data of a new user at the early stage are not adequate to train an

accurate model, the recommendations from crowdsourcing may still be accurate if we can find enough neighbours.

4.2.3 User-based CF location-privacy recommenders

To apply user-based CF to location-privacy recommendations, first, we need to represent people's location-privacy preferences by using the classical representation of preferences in user-based CF [113]. In the scenario of LSS, people's location-sharing behaviours are always in different contexts. These contexts can be represented by many dimensions, including location categories, time slots, recipient types, and so on. For example, if a user decides to share locations when he or she is in restaurants at noon, and allows his or her families to see them, then the context of such "share " decision is $(restaurant, noon, families)$. By this means, we can use the dimensions that we take into account to represent all the possible contexts for people's location-privacy decisions. All of these contexts are in uniform formats and can be simply extended if new dimensions are introduced. If we treat contexts as items and treat people's location-privacy decisions as ratings in these contexts, then we can have a user-context matrix that represents all users' location-privacy decisions in all possible contexts. Each of the rows is a user's location-privacy preference.

Formally, if we consider a set of dimensions $D = \{d_1, d_2, ..., d_{N_D}\}$, the set of contexts can be represented as:

$$C = d_1 \times d_2 \times ... \times d_{N_D}$$

$\times$ means the Cartesian product of two sets. Each dimension d_i is a set of variables. For example, if we have two dimensions, location category $d_1 = L = \{home, leisure\}$ and time slot $d_2 = T = \{morning, evening\}$, then the set of contexts C is:

$$C = d_1 \times d_2 = L \times T = \{(home, morning), (home, evening), (leisure, morning), (leisure, evening)\}$$

We use U to represent all the users in question. We consider two dimensions, i.e., location

category and time slot. The set of time slots of the LSS in question is T and the set of location category is L. Thus, the set of all the possible contexts in the LSS is $C = T \times L = \{c_1, c_2, ..., c_{|T||L|}\}$. The location-privacy preference of u_i can be represented by a vector $R_i = (r_{i,1}, r_{i,2}, ..., r_{i,|T||L|})$, where $r_{i,x}$ is the binary privacy setting (i.e., "share" or "not share") of u_i in context c_x. Table 4.2 shows all the terms that we use to describe our location-privacy recommender.

u	a user
U	the set of all users
t	a time slot
T	the set of all time slots
l	a location category
L	the set of all location categories
C	the context set, $C = T \times L = \{(t_1, l_1), (t_1, l_2), ..., (t_{\|T\|}, l_{\|L\|})\}$ $= \{c_1, c_2, ..., c_{\|T\|\|L\|}\}$
R	a location-privacy preference, $R_i = (r_{i,1}, r_{i,2}, ..., r_{i,\|T\|\|L\|})$, $r_{i,x}$ is u_i's location-privacy setting ("share" or "not share") in c_x

Table 4.2: Terms and symbols

Table 4.3 is an example of user-context matrix. u_i's location-privacy decision for context c_x is represented as $r_{i,x}$ and it is either positive ("share") or negative ("not share").

User	(home, morning)	(home, evening)	(leisure, morning)	(leisure, evening)
u_1	P	N	NULL	N
u_2	N	P	P	?
u_3	N	P	P	P
u_4	N	P	P	P
u_5	N	P	P	N

Table 4.3: An example of user-context matrix for 5 users and 4 contexts. Location-privacy decisions are positive (P), i.e., "share", negative (N), i.e., "not share", or unrated (NULL). The question mark denotes the context in which a recommendation needs to be made: (leisure, evening) for u_2.

When u_i requests a location-privacy recommendation for context c_x, first we need to find the users whose location-privacy preferences are similar to u_i's. Since the recommendation is made from these users, they must have location-privacy decisions for c_x, which means that in the user-context matrix, their location-privacy decisions in the column c_x should not be $NULL$.

To find the most similar users to u_i, we need to calculate the similarities between R_i and the preferences of the other users. In the user-context matrix, we use two numerical values, $r_{positive}$ and $r_{negative}$, to represent the "share" decision and the "not share" decision, respectively. By this means, we have each user's location-privacy preference as a vector $R_i = (r_{i,1}, r_{i,2}, ...)$, and each value in the vector is either $r_{positive}$ or $r_{negative}$. When we calculate the location-privacy preference similarity between u_i and u_j, we do it by calculating the similarity of the two vectors R_i and R_j.

Different users may have different scales when considering their location-privacy decisions. Even the same rating value (i.e., a location-privacy decision) may represent different preferences by different users. For example, in the scenario of LSS, a "share" decision from a user who always shares locations may be different from a "share" decision from a user who rarely shares locations. To reduce the bias from different personal scales, we normalise each user's rating vector by subtracting the user's mean rating from each element in the vector, i.e., mean-centered normalisation. For u_i, the normalised rating vector is:

$$R_i^* = \{r_{i,1} - \bar{r}_i, r_{i,2} - \bar{r}_i, \cdots\}$$

We use the Cosine similarity of R_i^* and R_j^*, which is often used to measure objects' similarities in information retrieval [97], as the similarity of their location-privacy preference. Thus, we have the similarity between u_i and u_j as:

$$sim_{i,j} = \frac{\sum_{x \in C_{i,j}} (r_{i,x} - \bar{r}_i)(r_{j,x} - \bar{r}_j)}{\sqrt{\sum_{x \in C_i} (r_{i,x} - \bar{r}_i)^2 \sum_{y \in C_j} (r_{j,y} - \bar{r}_j)^2}}$$

where $C_{i,j}$ is the set of contexts for which both u_i and u_j have location-privacy decisions.

The result of $sim_{i,j}$ is bounded in $[-1, 1]$. For the Cosine similarity of two vectors, -1 means two opposite vectors, and 1 means two same vectors. 0 similarity means two orthogonal vectors. As suggested by Ning et al. [97], whether non-positive similarity values can be removed depends on the used data. In our experiment, we find that the recommender's performance is better when we

consider non-positive similarity values. Thus, when calculating the neighbourhood of u_i in terms of c_x, we have:

$$N_x(i) = \{j | r_{j,x} \neq NULL\}$$

In user-based CF, when constructing a neighbourhood, an important parameter needs to be controlled is the neighbourhood size. In our case, it is how many users are allowed to be in $N_x(i)$. All the users in $N_x(i)$ are sorted by their similarities to u_i in descending order. The top-N users (i.e., the N users with the highest similarity to u_i), where N is the neighbourhood size, are chosen to contribute to the recommendation in c_x. On the one hand, if the neighbourhood size is too small, the recommendation may be biased by individual users. On the other hand, if the neighbourhood size is too large, there may be many low similarity users influencing the recommendation. Thus, we control N as a parameter in our experiment and investigate its influence on the performance of recommendations.

Once the top-N neighbours are chosen, we use their location-privacy decisions in context c_x to generate a recommendation for u_i. Among these neighbours, there are two types of decisions, i.e., "share" and "not share". The most straightforward way to generate the recommendation is to use the most popular decision. However, since these neighbours have different similarities to u_i, their decisions should weigh differently when generating the recommendation. Therefore, we use their similarities as their weights in the calculation of recommendation, i.e., $w_{i,j} = sim_{i,j}$. The neighbours who have higher similarities to u_i contribute more than others do to make the recommendation. We have the location-privacy recommendation for user u_i in c_x as:

$$\hat{r}_{i,x} = \bar{r}_i + \frac{\sum_{j \in N_x(i)} w_{i,j}(r_{j,x} - \bar{r}_j)}{\sum_{j \in N_x(i)} |w_{i,j}|}$$

$\hat{r}_{i,x}$ is a value between $r_{positive}$ and $r_{negative}$. To decide whether to recommend "share", we compare $\hat{r}_{i,x}$ against a threshold θ, which is the median value of $r_{positive}$ and $r_{negative}$. Then the final decision made by the recommender for u_i in context c_x is:

$$decision_{i,x} = \begin{cases} not\ share & \text{if } \hat{r}_{i,x} \leq \theta \\ share & \text{if } \hat{r}_{i,x} > \theta \end{cases}$$

4.3 Recommendation accuracy and privacy leak

The most important performance metric of recommender systems is how accurate it is. In our location-privacy recommender, when a setting is recommended in a context for a user, it is either "share" or "not share". The user's actual decision in the context is either "share" or "not share", too. Thus, as shown in Figure 4.1, there are four possible combinations of the user's actual decision and the recommended decision.

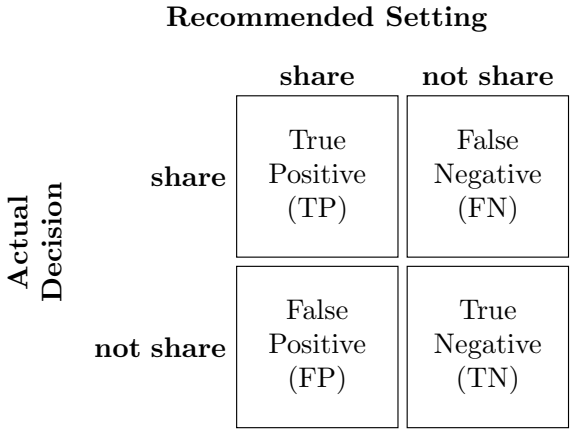


Figure 4.1: Confusion matrix of actual decision and recommended setting.

To evaluate how accurate our recommender is, we use the percentage of correct recommendations among all recommendations as the *accuracy* of our recommender:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Apart from *accuracy*, we are also interested in how much our recommender overexposes users'

location information. Among the two types of incorrect recommendations, i.e., FP and FN, we consider FP is riskier than NP. It is because an FP recommendation leads to overexposure of users' location that they do not want to share. Therefore, we evaluate the percentage of FP recommendations among all recommendations as the privacy *leak* of our recommender:

$$leak = \frac{FP}{TP+TN+FP+FN}$$

We compare the performance (i.e., *accuracy* and *leak*) of our recommender with one baseline recommender, three model-based classifiers, and a more advanced CF recommender.

The baseline recommender uses the most popular location-privacy setting in each context as the recommendation in the context. It is a general crowdsourcing scheme without considering the difference of similarities among users.

For the model-based classifiers, we use J48 decision tree [111], Naïve Bayes [62], and Rotation Forest [115]. We use the WEKA [48] software (version 3.6.10) to implement these classifiers and we use the default parameters to configure the classifiers.

We also compare our user-based CF recommender with a more advanced CF recommender, i.e., Matrix Factorization [74] (MF). User-based CF fits our assumption, which is that there is similarity in people's location-privacy preferences. As a more advanced CF algorithm, MF has shown better performance than user-based CF in some recommendation scenarios [74]. Therefore, we are interested to find out whether MF has better performance than user-based CF when recommending location-privacy settings. We use the Lenskit recommender toolkit [39] to implement both the user-based CF recommender and the MF recommender.

To evaluate the performance of our recommender systems and the other schemes, we tested them on the location-privacy preferences in the st_andrews/locshare dataset [103], which was collected from the real world. In this dataset, each row is a user's location-privacy decision that contains the user's ID, the time when this decision was made, the location category, and the response (i.e., "share" or "not share"). There are also two columns that represent the recipient type and the co-presence. Because of the high percentage of $NULL$ data in these two columns,

we do not use them in our experiments.

The dataset has six location categories:

$$\text{L} = \{\text{Food \& Drink, Leisure, Retail, Residential, Academic, Library}\}$$

The column of the times when decisions were made is in a high-granularity time stamp format. We convert them into five time slots:

$$\text{T} = \{\text{Morning, Noon, Afternoon, Evening, Night}\}$$

The rules of conversion are shown in Table 4.4.

time slot	**time range**
Morning	0700 – 1159
Noon	1200 – 1359
Afternoon	1400 – 1659
Evening	1700 – 2059
Night	2100 – 0659

Table 4.4: Time slot conversion rules

After removing *NULL* data and converting time stamps to time slots, each instance is in the format as $(id, t, l, decision)$, which represents a location-privacy *decision* ("share" or "not share") of a user in the time slot t and location category l. In each round of our experiments, we randomly split these instances into ten equal-sized subsets. Each of these subsets is used as the testing set to evaluate the performance of all the schemes, and the remained nine subsets are merged and used as the training set to build models and recommenders. Therefore, there are ten evaluations in each round. For each round, we use the average result of these ten evaluations as the result of this round. We also repeat 100 rounds of experiments and use the average result of the 100 rounds as the final result. Thus, our experiment is repeated (100 rounds) 10-fold cross validation.

In the dataset, the *decision* of each row is only for the time when that instance was collected. During the data collection, one participant might visit the same location in the same timeslot for many times. Thus, for the same (id, t, l), there might be different *decision*s. In our user-based

CF location-privacy recommender, one user can only have one location-privacy decision in one context. Therefore, when using training sets to build our recommender, for each user, we use the most frequent *decision* in context (t,l) as the user's location-privacy decision in the context.

To find out the influence of the maximum size of neighbourhood, we evaluate both *accuracy* and *leak* of our recommenders with different N. Since there are 40 users in the dataset, we change N from 1 to 39.

As shown in Figure 4.2 and Figure 4.3, with the increase of N, more neighbours are allowed to contribute to recommendations, and both *accuracy* and *leak* are improved. As we consider both positive and non-positive similarity values, as N increases, some neighbours with very low similarity values to others may participate in recommendations as N grows. Thus, there are fluctuation points such as $N = 6$ and $N = 14$ in Figure 4.2, and $N = 15$ in Figure 4.3. The overall performance, however, is better when considering non-positive similarity values.

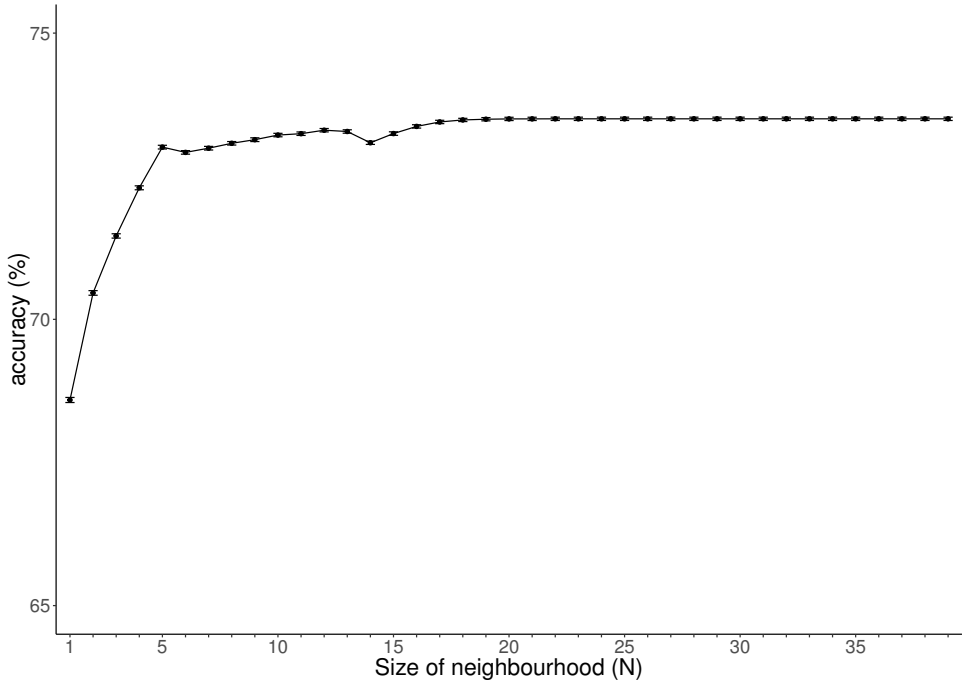


Figure 4.2: The change of *accuracy* with the increase of neighbourhood size (N). *accuracy* significantly increases until $N = 5$ and then slightly fluctuates, reaching the highest when $N = 22$.

Our recommender has the highest *accuracy* when $N = 22$ and has the lowest *leak* when $N = 8$. We name the first one as CF-A and the second one as CF-P and take both of them into account in

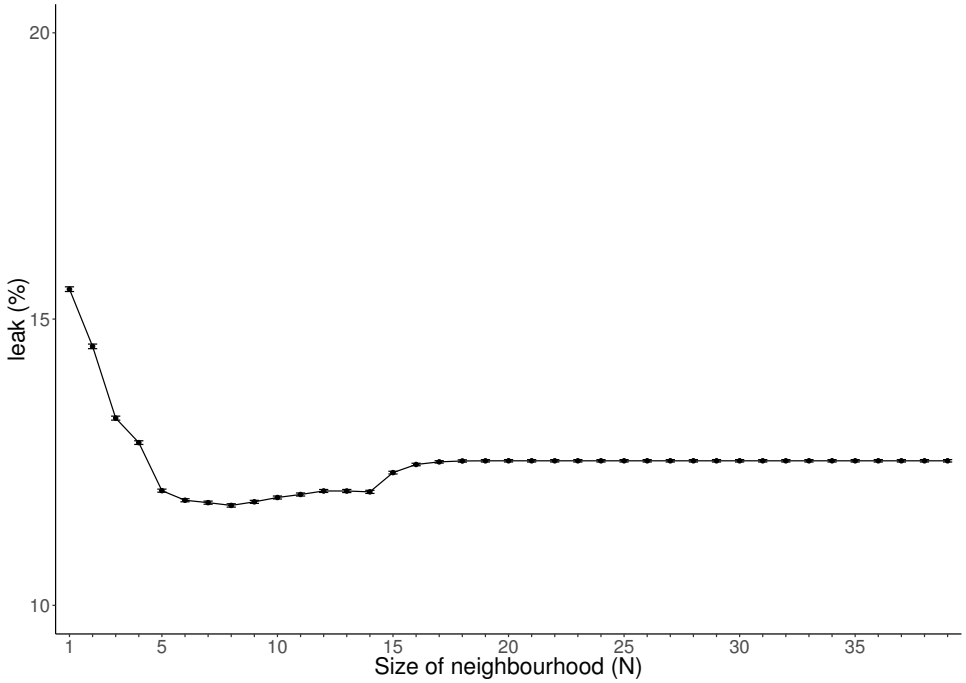


Figure 4.3: The change of *leak* with the increase of neighbourhood size (N). *leak* significantly decreases until $N = 5$ and then slightly fluctuates, reaching the lowest when $N = 8$.

the comparison with the other schemes.

As shown in Figure 4.4, the x-axis represents *leak* and the y-axis represents *accuracy*. Location-privacy recommendations should be as accurate as possible, and should cause overexposure as little as possible. The *Semantic* scheme, in the lower right corner, has the lowest *accuracy* but the highest *leak*. This result shows that general crowdsourcing recommendations based on location category and time are not accurate enough to satisfy people's different types of location-privacy preferences. The performance of *NB* is close to that of *MF*. Both of them are less accurate than our recommenders and have higher *leak* than our recommenders do. The *MF* scheme, as a more advanced CF technique, does not have better performance than our user-based CF scheme does. Although the algorithm of user-based CF is simple, the assumption behind it satisfies the fact that people have similar location-privacy preferences, which leads to higher *accuracy* than that of *MF*. This finding corresponds to the results of Xie *et al*. [145].

CF-A and *CF-P* are in the upper left corner in Figure 4.4, and are close to *RF* and *J48*. They both have lower *accuracy* and lower *leak* than *RF* and *J48*. When overall performances are similar, in

the trade-off between *accuracy* and *leak*, we believe that lower *leak* is more important than higher *accuracy* in the scenario of LSS. The reason is that overexposure is riskier than underexposure. In this case, our schemes outperform *RF* and *J48*. Additionally, model-based schemes such as *RF* and *J48* have computationally expensive processes such as Bootstrap Aggregating (Bagging) and periodical update of tree structures. Our scheme provides a less computationally expensive choice for real-world implementation.

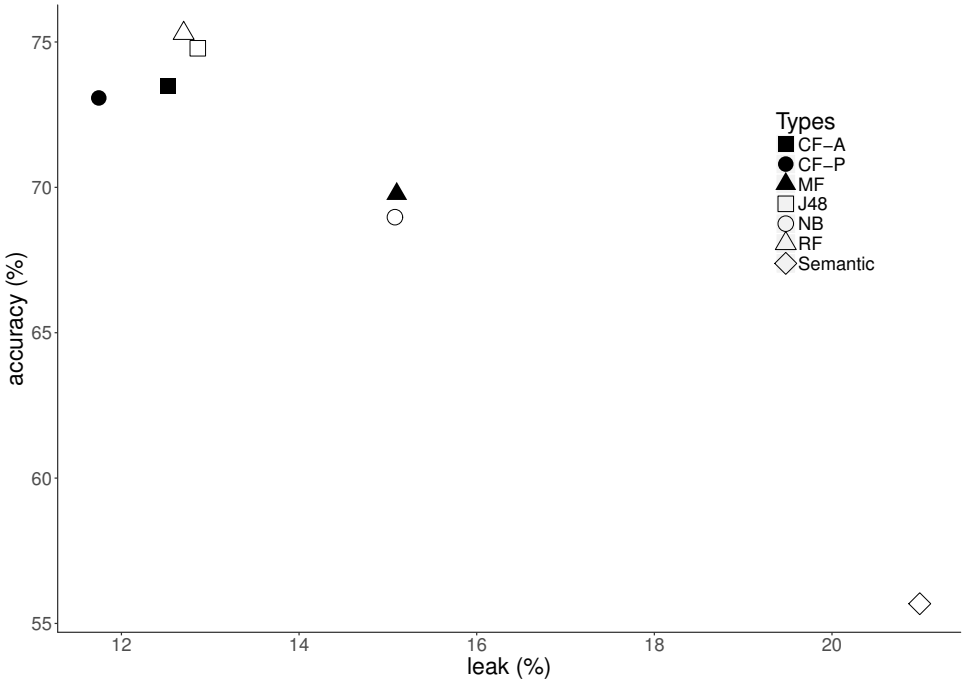


Figure 4.4: *accuracy* and *leak* of CF (CF-A has the highest *accuracy* and CF-P has the lowest *leak*), MF, model-based machine-learning classifiers (J48, Naïve Bayes, Rotation Forest) and crowdsourcing semantic predictions. The CF recommender outperforms crowdsourcing semantic prediction and MF in terms of both *accuracy* and *leak*. The *accuracy* of CF is close to the best performance of model-based machine-learning classifiers and it causes lower *leak*.

4.4 Recommendations using insufficient data

Our first experimental result shows that the overall performance of user-based CF location-privacy recommender is close to the best performance of model-based classifiers. Meanwhile, our recommender causes lower *leak* than model-based classifiers do. The evaluation of overall performance is done by 10-fold cross validation, which means that 90% of the whole dataset is available to be used to train models and recommenders. In real-world applications, it is not

uncommon that people do not have sufficient data to generate recommendations. For example, when a new user starts to use location-privacy recommenders, there are not enough data about the user's location-privacy decisions to build models that describe the user's location-privacy preferences. This is known as the cold-start problem. As a consequence, model-based classifiers may fail to provide accurate recommendations during cold-start periods. In user-based CF, since recommendations are made from other users' preferences, it is still possible to make accurate recommendations, as long as we can find similar neighbours.

To investigate whether our recommender can outperform other schemes during cold-start periods, we test them by using insufficient data. We assume a scenario where our recommender has run for a period of time. All of the existing users have already given enough location-privacy decisions to the system. We consider a new user with insufficient previous information begins to use the recommender. Thus, for this new user, the performance of recommendation may be low, due to the lack of information to build the user's models. In each round of experiment for the cold-start tests, we iterate each user in the dataset as the new user in question. We incrementally add the new user's data into the training sets, starting with 1% and increasing it with 1% until reaching 10%. For each user, we repeat 100 rounds of experiments. In each round, the seeds used to randomly split the new user's data are different. We examine the difference between the performances of different schemes in the cold-start tests. We take *CF-P* as the representative of our recommender and name it as *CF* in the cold-start tests.

As shown in Figure 4.5 and Figure 4.6, the performances of all the schemes are improved with the increase of training data. At the beginning of the cold-start tests, *CF* has higher *accuracy* and lower *leak* than *RF* does. After the percentage of training data reaches 6%, *RF* has enough to provide more accurate recommendations than *CF*. The *leak* of *RF* is higher than that of *CF* during the entire cold-start tests.

Our results show that user-based CF location-privacy recommenders outperform model-based classifiers in the cold-start tests. This is because the recommendations of user-based CF are made from crowdsourcing data instead of personal data. For a new user, there are not enough personal data to build accurate model-based classifiers. Using similar neighbour users, however, can provide more accurate recommendations than model-based classifiers do. Moreover, our

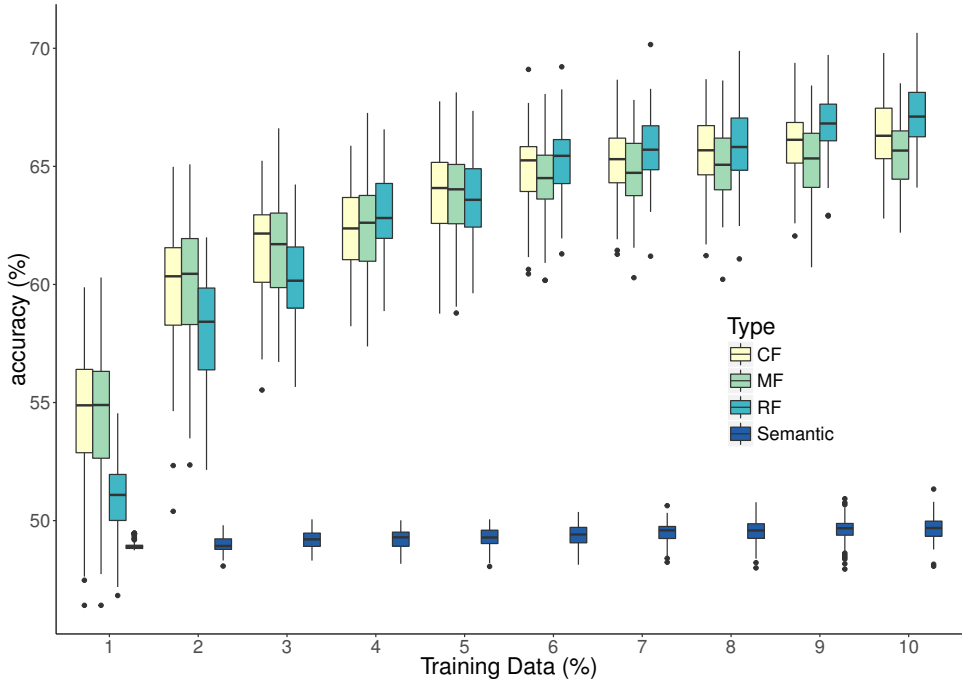


Figure 4.5: *accuracy* of CF, MF, RF and crowdsourcing semantic prediction during the cold-start period. CF recommender can provide higher *accuracy* than RF (except 4%) until using 6% of personal data for training. The *accuracy* of using CF is higher than using crowdsourcing semantic prediction. MF performs slightly worse than CF does. (The box plots in this thesis are made by using ggplot2 [141] – an R package. In each box, the lower and upper hinges are the first and the third quartiles, and the middle hinge is the median. The whiskers are from the hinges to the furthest values within 1.5 * inter-quartile range. Data further than the ends of the hinges are outliers.)

scheme has lower *leak* in the cold-start tests. Since cold-start problems are common issues when deploying recommenders in the real world, our result suggests that using user-based CF can potentially help address these issues.

4.5 Summary

In this chapter, we have demonstrated the following:

- User-based CF can be used to recommend location-privacy settings.
- The best overall *accuracy* and *leak* of user-based CF location-privacy recommender are close to the best performance of the state-of-the-art.

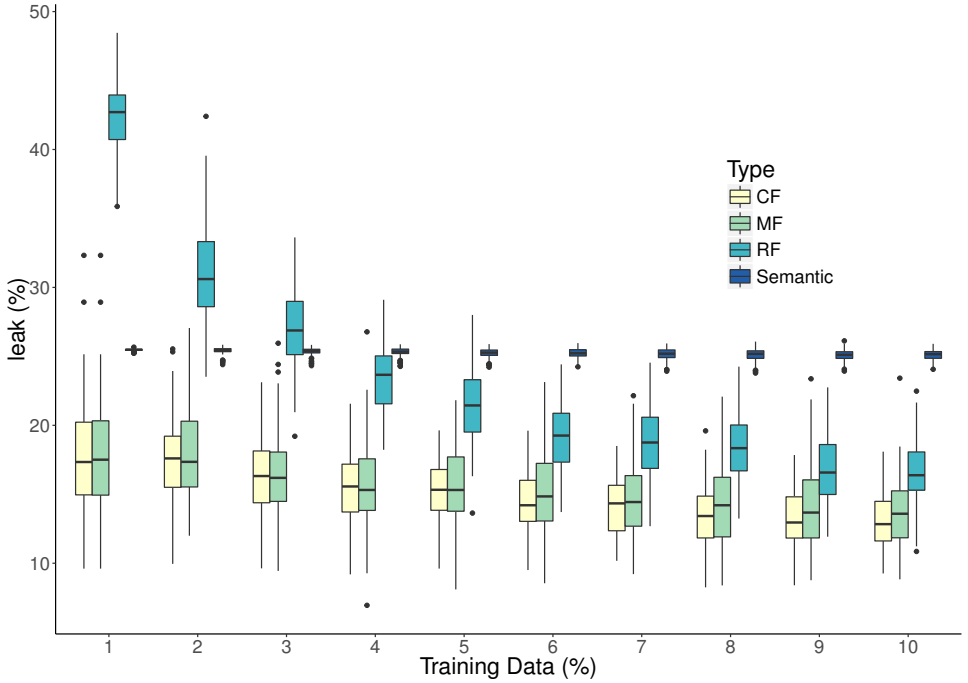


Figure 4.6: *leak* of CF, MF, RF and crowdsourcing semantic prediction during the cold-start period. The CF recommender causes fewer *leak* than RF and crowdsourcing semantic prediction during the cold-start period. MF performs slightly worse than CF does.

- In the cold-start tests, user-based CF recommenders outperform the other recommenders using individual's data.

When deploying recommenders in the real world, their recommendations are used by users. Thus, recommenders and users compose recommender systems, wherein they interact with each other. In recommender systems, users play an important part in the Human-Recommender Interaction (HRI) [89]. Good recommendations are not only accurate, but also acceptable by recommender users. Therefore, in the next chapter, we investigate whether people accept our location-privacy recommenders and what factors may affect their acceptance.

Chapter 5

Acceptance of location-privacy recommenders

5.1 Introduction

Chapter 4 has demonstrated that user-based CF, as a technique of neighbourhood-based CF, can make accurate recommendations for location-privacy settings. In addition, it outperforms model-based recommenders when training data are insufficient. In recommender systems, apart from recommender algorithms, the users who use recommenders are also an important part of the evaluation in the HRI, as they can decide whether to accept the recommendations made by the systems. Therefore, location-privacy recommenders need to be acceptable, and it is worth to investigate what factors can affect people's acceptance.

In this chapter, we evaluate our location-privacy recommenders from users' perspective to find out which factors can affect people's acceptance of location-privacy recommendations. We conduct an online user study that asks people to use our location-privacy recommenders. We collect data from their social network profiles, their interaction with the recommenders, and their answers to questionnaires.

We investigate the influence of two types of factors on users' acceptance. One type is users' *subjective* factors. For example, people with different levels of trust, concern, or satisfaction, may accept our recommenders differently. Another type is recommenders' *objective* factors,

such as the contexts of recommendations or the source of recommendations. These factors may also affect users' acceptance.

Our online user study aims to answer the following questions:

- **Q**1 Which subjective factors, such as privacy concerns, etc., affect people's acceptance of location-privacy recommendations?
- **Q**2 Which objective factors, such as contexts of recommendations, etc., affect people's acceptance of location-privacy recommendations?

Since people's location-privacy preferences are dynamic, we expect that their acceptance of location-privacy recommendations may also be dynamic for different contexts, different levels of recommendation openness, and different crowdsourcing sources. In addition, due to the sensitivity of their location-privacy preferences, people may accept location-privacy recommenders differently according to their trust of technologies and their privacy concerns about our system.

5.2 Methodology

We aim to find out the effects from different factors, both subjective and objective, on people's acceptance of location-privacy recommendations. To do this, we need to design and conduct our experiments under a framework that enables us to measure those subjective factors and objective factors, and evaluate the relations between them.

Many models and frameworks have been proposed to investigate the effects from factors other than accuracy in recommendation systems. The subjective factors of users include personal characteristics, perceived recommendation quality, transparency of systems, and so on. These factors play different roles and can affect each other in HRI. One example is the work of Zins and Bauernfeind [159] that investigates what factors can affect people's satisfaction on recommender systems. Their study mainly focuses on the effects from people's personal characteristics, such as Internet expertise, product involvement, and Internet purchase attitudes. Based on their model, they find that these subjective factors affect people's experience of using online recommender systems. Similarly, Pu *et al*. [110] propose a user-centric framework to investigate the effects of

subjective factors such as perceived system qualities, beliefs, and attitudes. Nevertheless, this framework does not take into account the objective effects from recommender systems either. Since we aim to investigate the effects from both subjective factors and objective factors in HRI, we conduct our user study using the framework proposed by Knijnenburg *et al.* [71], which is a structured model that covers both the types of factors of our interests.

5.2.1 User-centric evaluation of recommender systems

The framework for user-centric evaluation formally describes different parts in HRI as different *aspects*. For example, recommenders are defined as several Objective System Aspects (OSA), such as underlying recommendation algorithms and graphical user interfaces. These OSA are objective factors and they affect users' perception of the recommenders. Users' perception, as subjective factors, is defined as Subjective System Aspects (SSA). SSA affect another two parts in HRI, which are users' Experience (EXP) (e.g., users' satisfaction about their choices) and their Interaction (INT) (e.g., whether accepting recommended items). Thus, SSA stand as moderators between the effects from OSA to EXP and INT. Another two subjective factors, Situational Characteristics (SC), such as privacy concerns, and Personal Characteristics (PC), such as demographics and domain knowledge, are also considered to affect EXP and INT directly.

We apply this framework to our location-privacy recommender system. We are interested in the effects of the following factors that may affect people's acceptance of location-privacy recommendations:

- *trust*: people's general trust in technology.
- *quality*: people's perceived quality of the recommended location-privacy settings.
- *satisfaction*: people's satisfaction about their chosen location-privacy recommendations.
- *concern*: people's privacy concerns about using location-privacy recommender systems.

Positioning these factors in the framework, we have *trust* as PC, *quality* as SSA, *satisfaction* as EXP, and *concern* as SC. For the change in OSA in our study, we use different crowd-

sourcing sources for recommendations as different conditions in OSA. People's acceptance of recommended location-privacy settings is considered as INT:

- *acceptance*: the percentage of location-privacy recommendations that a person agrees to use.

The overall diagram of all the factors and relationships in our study is shown in Figure 5.1.

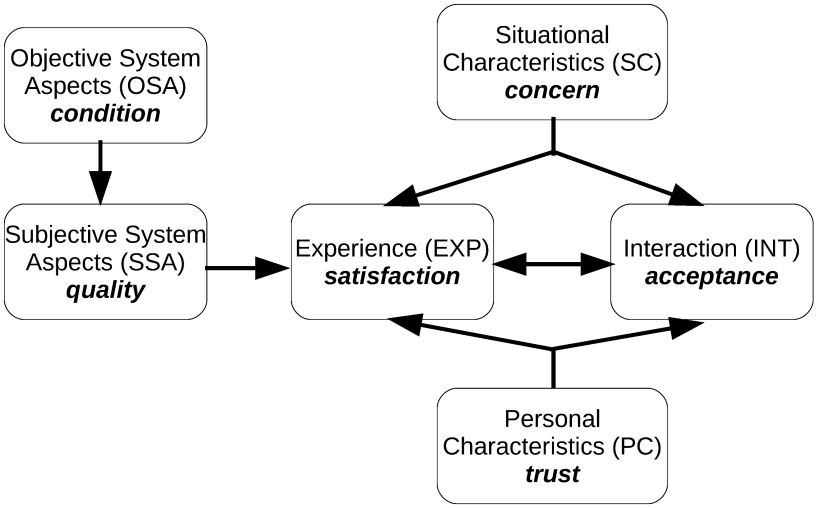


Figure 5.1: Diagram of the framework for the user-centric evaluation of recommenders as used in our experiment.

5.2.2 Questionnaires

In our user study, the objective factors such as *acceptance* can be directly calculated and evaluated. The subjective factors, on the contrary, are constructs that cannot be directly observed. Therefore, we need to design questionnaires, which are commonly used in user studies to capture people's constructs, to evaluate the subjective factors in our study.

For each subjective factor, e.g., people's perceived recommendation quality, the most straightforward way to evaluate it is to ask participants questions such as "How good do you think the recommendation is?", and to ask them to answer with a number from 1 to 7. This method, however, has two drawbacks. The first one is that a single question may not be able to capture

all the characteristics of "good recommendations". People may consider recommendations good because of different reasons (e.g., accuracy or serendipity). Therefore, it is reasonable to adopt multiple questions for each subjective factor. The second one is that different people may interpret the rating scale of answers differently. Some people may be reluctant to use the highest rating in the scale of the answer, while others may be not. Thus, the same rating number from different participants may not mean the same level of perceived quality. To avoid this bias, we present each question as a statement instead of a question. And we ask people to answer by indicating to what extend they agree with the statement (e.g., from "Strongly disagree" to "Strongly agree").

We use the questionnaires from the user-centric framework [71] and adjust the questions according to the scenario of LSS. The questionnaires that we use in our study are: *General trust in technology* for *trust*, *Perceived recommendation quality* for *quality*, *Choice satisfaction* for *satisfaction*, and *System-specific privacy concern* for *concern*. These questionnaires have been used and revised in previous research. Adopting them in our user study increases our chance to collect valid answers. Each questionnaire has multiple questions, each of which is a statement. We refer each question as an *item* and use a five-point Likert scale from "Strongly disagree" to "Strongly agree" as the choices of answers. The detailed questionnaires can be found in Appendix C.

5.2.3 Online user study

We hypothesise that **people's acceptance of location-privacy recommendations change with different crowdsourcing sources, with different levels of recommendation openness, and different contexts**. To test these hypotheses, we designed an online location-privacy recommender system with three crowdsourcing sources. We recruited participants to use our recommenders and then investigated which factors have effects on their acceptance of location-privacy recommendations.

We invited our participants to login with their Facebook accounts when using our system. By this means, our recommenders could use their real-world location check-in information to generate recommendations. Our online user study had three parts. First, to each participant, we showed

a prebriefing page that explained the different recommenders in our system, and how these recommenders could help people with their location-privacy protection. Then in the second part, we showed the participants with a series of location-privacy recommendations for the location check-ins in their Facebook histories (as shown in Figure 5.2). When presenting a recommendation to a participant, we told the participant that the recommendation was made using a particular crowdsourcing source. In the example of Figure 5.2, we told the participant that the recommendation is generated from his or her Facebook friends' data. In fact, we used the same random generator to make all the recommendations in our experiment. Participants were hidden from this fact. The reason was to keep the objective recommendation accuracy the same, so that it did not affect *quality*, *satisfaction*, or *acceptance*. The effect of objective recommendation accuracy was not of our interests in our user study. Finally, in the third part, as shown in Figure 5.3, we provided the participants with our questionnaires asking them about their perceived quality of recommendations (*quality*), their satisfaction about their choices (*satisfaction*), and their system-specific concerns (*concern*).

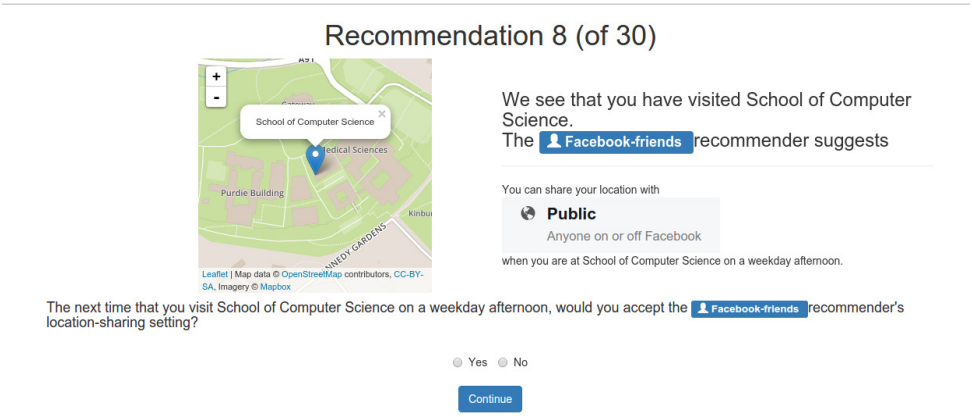


Figure 5.2: Each participant in our experiment was presented with 30 recommendations made by our 3 recommenders (10 recommendations from each recommender), and asked if they would accept the recommendation.

We advertised our experiment in several ways, including university mail lists and university Facebook groups. In our experiment and advertisement, we did not use the term "location-privacy preference". Instead, we used "location-sharing preference" to avoid biasing our sample by increasing the number of privacy fundamentalists in our participants or making the participants become more privacy aware during the study. 164 participants tried to take part in our experiment.

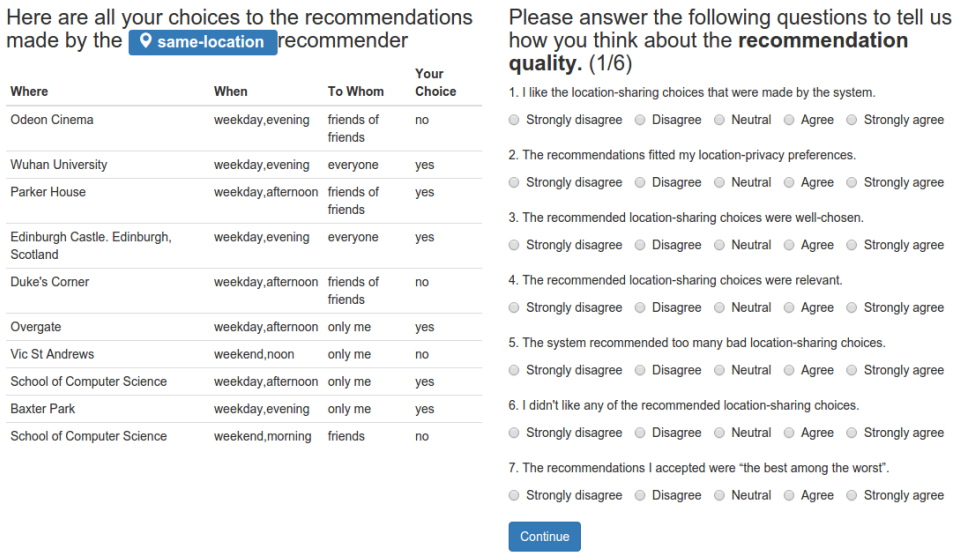


Figure 5.3: Questionnaires were used to collect data on perceived recommendation quality (*quality*), satisfaction of choices (*satisfaction*), and concerns about our system (*concern*). Participants were also presented with their choices as a reminder.

For each participant, to make sure that there were enough data to generate recommendations, we only allowed those who had at least 10 distinct location check-in instances to take part in. There were 99 participants that satisfied this criterion. Each of them received a £5 Amazon voucher for participation.

We submitted the proposal of our experiment to our institutional ethics committee and our application was scrutinised and approved. The form of ethics approval can be found in Appendix B.

5.2.3.1 Prebriefing

In the prebriefing of our user study, we told the participants that our system had three recommenders that used different crwodsourcing sources to make location-privacy recommendations, which were:

- *same-location* recommender: using the data of people who have been to the same location.
- *similar-people* recommender: using the data of people who have similar previous location-sharing preferences.

- *Facebook-friends* recommender: using the data from people's Facebook friends.

To make sure that our participants understood the difference between different recommenders, we also provided three examples to familiarise them with our recommenders. There was also a small quiz after the prebriefing to check if they fully understood. After the participants completing the quiz without mistakes, we asked them to login with their Facebook accounts through the PRISONER platform [58]. The PRISONER platform was designed to keep experiments related to social media privacy-sensitive, which means that we could only access to the minimum amount of data from our participants for our experimental goal. The participants were notified what kind of data we asked from them and they could explicitly give us consents to access these data.

We asked the access to the participants' public information, location check-in information, and Facebook friend lists. Since all the recommendations were generated randomly, we did not actually use their friends' data as data sources. The reason why we asked for Facebook friend lists was to make our recommenders look realistic.

To make sure for each participant there were enough data to generate recommendations, we only allowed the participants who had at least 10 distinct location check-ins to enter the next part of the experiment.

5.2.3.2 Exploring recommendations

In the second part of our experiment, we showed the participants some location-privacy recommendations and evaluated their *acceptance*. Before doing this, we asked our participants about their demographic information including age and gender, and used a questionnaire to ask about their general trust in technology, i.e., the *trust* factor in our experiment. Table 5.1 shows the demographic information of our participants. Due to the means by which we recruited our participants, the samples in our experiment are mainly university students from 18 to 24. The university is located in a small town, which means the daily lives of our participants may happen in several popular places (e.g., libraries, classrooms, and dormitories). Thus, our experimental results may be biased by our samples and cannot represent the other population such as older age groups or people living in cities.

Category	Options	Participants(%)	Facebook(%)
Gender	Female	63	51
	Male	37	49
Age	18-24	74	21
	25-34	24	27
	35-44	2	20
	45-54	0	16
	55+	0	16

Table 5.1: The demographics of our experiment (Participants) compared with the overall UK Facebook over-18 user population (Facebook). The Facebook data were taken from the Facebook Adverts Manager in October 2015.

Since we claimed that there were three recommenders using different crowdsourcing sources, there were three conditions for the OSA. We tested our participants in these three conditions on a within-subject basis, which means we presented each participant with 30 recommendations, 10 from each recommender. For each participant, for the 10 recommendations from each recommender, we selected 10 location check-ins from the participant's location history on Facebook as the contexts for the recommendations.

As shown in Figure 5.2, every time we showed a location-privacy recommendation to a participant, there were also the context of location check-in, a map of the location check-in, and the recommended setting from one of our recommenders. For the context, we had the name of the location and the time slot of the location check-in. The rules of converting timestamps to time slots are the same as in 4.4. For the recommended location-privacy settings, we used the default settings from Facebook, i.e., *Only Me*, *Friends*, *Friends of friends*, and *Public*, and randomly chose one for each recommendation. We asked the participants whether they wanted to use the recommended setting in their future visit to the place during the time slot in the context of the recommendation. A "Yes" answer was recorded as an accepted recommendation, otherwise the recommendation was not accepted.

5.2.3.3 Final questionnaires

After the participants completed exploring all the recommendations, in the third part of our experiment, we evaluated their subjective factors. We collected data on *quality*, *satisfaction*, and *concern*. For the first two factors, we wanted to know whether different recommenders

affected these factors differently. So we evaluated *quality* and *satisfaction* for each of the three recommenders. Thus, there were six questionnaires to be answered first. When the participants answered one of these questionnaires for a particular recommender, as shown in Figure 5.3, we showed the participants' choices to the recommendations from this recommender in the second part of the experiment. The reason of doing this was to remind them with their decisions. For *concern*, since it was about system-specific privacy concerns, we only used one questionnaire to evaluate it after the participants completed the first six questionnaires.

After the questionnaires, we gave our participants opportunities to give us free-text comments at the final step of our experiment. They could comment about their opinions and suggestions about our system.

5.3 Analytical approaches

The data collected in our experiment are in three forms. The first one is the participants' *acceptance* of the recommended location-privacy settings. It is recorded as the percentage of accepted recommendations among all recommendations. The second one is the participants' answers to all the questionnaires. Each factor is measured by using one questionnaire and each questionnaire contains several question *items*. The answer of each *item* is in the form as a 5-point Likert scale, from "Strongly disagree" to "Strongly agree", to represent to what extent the participants agree with a statement. The third one is the optional free-text comments given by the participants at the end of our experiment.

For *acceptance*, we can directly use it in our analysis since it is numerical. For the subjective factors such as *quality* and *satisfaction*, we need to convert the questionnaire answers to measurable values that can be used in our analysis.

The first step is to establish the validity of the measured subjective factors. This is to make sure that the questionnaires have successfully captured the subjective factors through participants' answers. To do this, we use Confirmatory Factor Analysis (CFA), which establishes both convergent and discriminant validity. Convergent validity is to make sure that the question *items* in the same questionnaire measure the same subjective factor. Discriminant validity is to make

sure that different questionnaires measure different subjective factors.

We convert all the questionnaire answers into ordinal values based on the 5-point Likert scale. Thus, through CFA, we can calculate the R^2 value of each question *item*, and use this value as the *loading* of the question *item*. Given a subjective factor, the criterion of being convergent valid is to have an AVE, which is the average value of all the R^2 values under the subjective factor, being larger than 0.5. Therefore, to maintain the convergent validity, we keep removing the question *items* with the lowest *loadings* until the AVE of the subjective factor becomes larger 0.5. We repeat this procedure for all the subjective factors in our experiment. To maintain discriminant validity, if the correlation between two subjective factors is higher than the square root of either the AVE of the two factors, then it means that the two questionnaires actually measure the same factor. In that case, we should remove the subjective factor with the lower AVE.

After establishing the convergent and discriminant validity, we investigate the potential effects between different factors. We propose several hypotheses about the possible effects and test whether they are significant. To do this, we need a model that allows us to position all our hypotheses. Structural Equation Modeling (SEM), as an integrative modeling method, enables us to test all the hypotheses at the same time. We apply SEM in our analysis so that we can use an integrative structure to analyse and link all the significant effects together.

The advantage of using the combination of CFA and SEM rather than other multivariate regression analysis is that CFA provides a way to test the validity of all the variables in questions, which eliminates the coefficients of invalid variables. Meanwhile, compared with other regression analysis, SEM can represent causal relationships [107, 20]. Thus, the coefficients detected in SEM represent the causal effects in our proposed model and the directions of arrows represent the causal directions [70].

In our experiment, we use lavaan [116], which is an R package, to do both the CFA and SEM analysis.

Construct	Question items	R^2	AVE
Quality	I like the location-sharing choices that were made by the system.	0.858	
	The recommendations fitted my location-privacy preferences.	0.789	
	The recommended location-sharing choices were well-chosen.	0.822	
	The recommended location-sharing choices were relevant.	0.440	
	The system recommended too many bad location-sharing choices.	0.469	
	I didn't like any of the recommended location-sharing choices.	0.328	
	The recommendations I accepted were "the best among the worst".	0.321	**0.575**
Satisfaction	I like the recommendations that I've accepted.	0.510	
	Some of my chosen location-sharing choices could become part of my default location-privacy settings.	0.506	
	I would recommend some of the chosen location-sharing choices to others/friends.	0.544	**0.520**
Concern	I'm afraid that the system discloses private information about me.	0.470	
	The system invades my privacy.	0.861	
	I feel confident that the system respects my privacy.	0.586	
	I'm uncomfortable providing private data to the system.	0.524	
	I think the system respects the confidentiality of my data.	0.571	**0.602**

Table 5.2: Results of Confirmatory Factor Analysis (CFA). Question items with low R^2 values are removed in the refined results. The general trust to technology (*trust*) is removed because it only has two question items to keep its AVE greater than 0.5. Both the convergent validity and the discriminant validity of our model hold.

5.4 The effect of privacy concerns

We first run CFA on the questionnaire answers of all the subjective factors. The refined results of the CFA after removing low-loading question *items* are shown in Table 5.2. In the refined results, *trust* only has two question *items* to make its AVE larger than 0.5. Since each factor needs no less than three question *items*, we remove *trust* from the SEM analysis. The convergent validity of our results holds, since the AVEs of *quality*, *satisfaction*, and *concern* are all larger than 0.5. We have not found any correlations between any two factors larger than the square root of the AVEs of both the factors, which means that the discriminant validity of our results holds too.

We apply SEM to the refined factors to investigate potential effects between them. For all the subjective factors, the answers of their questionnaires are represented as ordinal variables. For the objective factor, i.e., *conditions*, we use the *same-location* recommender as a baseline condition. Then we can use two dummy variables, i.e., *friends* and *similar*, to represent the conditions of the *Facebook-friends* recommender and the *similar-people* recommender, respectively.

We aim to find out any significant effects between the factors (both subjective and objective) in our experiment based on the structure of the user-centric evaluation model (Figure 5.1). For *quality*, we hypothesise that it is affected by users' privacy concerns and the crowdsourcing sources. Therefore, we have $quality \sim concern + similar + friends$. Similarly, for *satisfaction*, we examine $satisfaction \sim concern + similar + friends + quality$. For *acceptance*, we examine $acceptance \sim concern + quality + similar + friends + satisfaction$. Since *trust* is removed in the SEM analysis, we do not hypothesise any effects on it.

Before looking at individual hypotheses, we need to check the fit of our SEM to make sure that our proposed model fits our collected data. Our SEM model's fit is adequate and can be measured as a series of metrics:

- $\chi^2_{125} = 483.67, p < 0.001$
- $root\ mean\ squared\ error\ of\ approximation\ (RMSEA) = 0.098$
- $Comparative\ Fit\ Index\ (CFI) = 0.977$
- $Turker - Lewis\ Index\ (TLI) = 0.972$

Hu and Bentler [55] propose that the cut-off values of good fit are: $CFI > 0.96, TLI > 0.95, RMSEA < 0.05$. Kenny *et al.* [66], however, suggest not measuring the RMSEA value for models that have small degree of freedom and small sample sizes.

The results of our SEM analysis are shown in Figure 5.4. We find four significant (p<0.001) effects. First, there are two negative effects from *concern* (SC) to *quality* (SSA) and *satisfaction* (EXP). This result means that the participants who have higher privacy concerns about our system perceive lower quality of recommendations. In addition, these participants are less satisfied with their choices. Another two effects, which are positive, are from *quality* (SSA) to *satisfaction*

(EXP) and *acceptance* (INT). This result means that *quality* acts as a mediator in these two indirect negative effects.

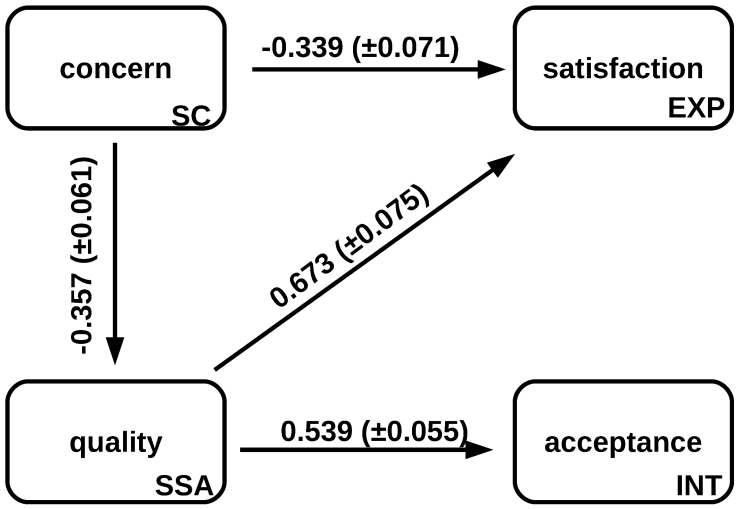


Figure 5.4: The structured equation modeling (SEM) results. $p < 0.001$ for all coefficients. Numbers above arrows mean the $\beta - weight(\pm standard\ error)$ of the effect. $Standard\ deviation = 1$. The *concern* has negative effects on *acceptance* (moderated by *quality*) and *satisfaction* (directly and moderated by *quality*).

The experimental results show that when using location-privacy recommenders, people's privacy concerns about sharing their data with such recommenders play an important role. To better understand the coverage of these concerns in our participants, we regress all the "Neutral" answers in the *concern* questionnaire into a baseline *concern*. We find out that 44% of our participants have higher *concern* than this baseline. These privacy concerns about sharing data with our recommenders not only decrease their acceptance of the recommended location-privacy settings (through their perceived recommendation quality), but also decrease their satisfaction about their choices (directly and through their perceived recommendation quality). When people sharing their location-privacy preferences with recommenders, they have to tell the recommenders both their location information and their privacy settings. Therefore, location-privacy preferences are inherently sensitive, and it is not surprising that *concern* negatively affects *quality*, *satisfaction*, and *acceptance*.

When designing recommender systems, our aim is to make users have both high acceptance of recommendations and high satisfaction about their choices. Both of them are users' subjective factors. As shown in Figure 5.5, in the left box, there are users' subjective factors, including *quality*, *satisfaction*, and *concern*, and their interaction *acceptance*. These factors cannot be directly manipulated. Thus for recommender system designers, they must measure those objective factors in the right box, some of which may be influential. By measuring potential effects and adjusting recommendation strategies, recommender system designers can make recommendations more acceptable. In our results, we find the effect from *concern* on *quality*, *satisfaction*, and *acceptance*. But we do not know what factors (? in Figure 5.5) on the right can affect *concern*. This question would be investigated in our future work.

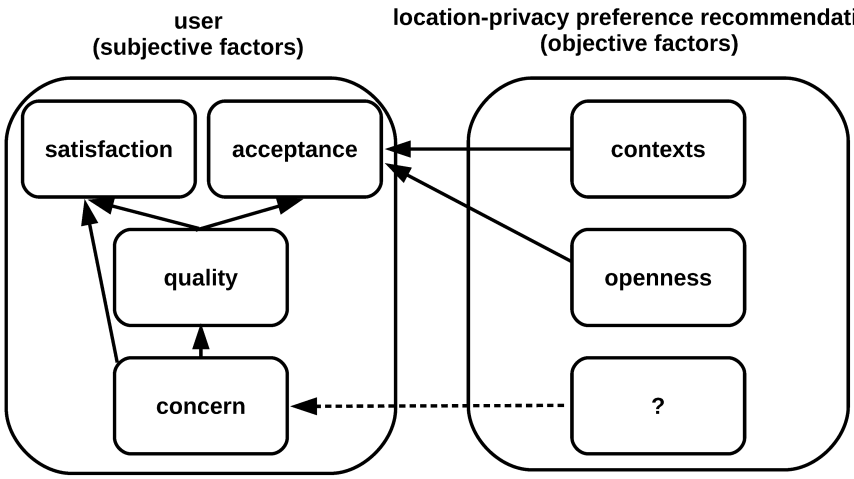


Figure 5.5: The effects of the subjective factors (left side) of users and the objective factors (right side) of location-privacy preference recommendations. The arrows in solid line mean the detected effects in our experimental results. The dash line means a potential effect from an unknown objective factor (marked as ?) to *concern*. Recommender system designers can only control the objective factors on the right side to influence the subjective factors on the left side.

For the *conditions* in our experiment, we find no significant effects from *friends* (OSA) or *similar* (OSA) to any other factors in our model. In other word, even though in the prebriefing we claimed that our recommenders used different crowdsourcing sources, the participants do not perceive any difference in the quality of the recommendations. As a consequence, they affect neither *satisfaction* nor *acceptance*.

5.5 The effect of openness

In the SEM analysis of our experiment, we find no significant effects from the controlled OSA, i.e., crowdsourcing sources. For recommendations , there are other OSA that may affect people's acceptance. For example, the recommended location-privacy settings in our experiment are *Only Me*, *Friends*, *Friends of friends*, and *Public*, which can be seen as from the least open setting (*Only Me*) to the most open setting (*Public*). Since the settings that have higher openness are riskier for sharing locations, people may be less likely to accept them. Therefore, we are interested in potential effects from the level of openness of the recommendations.

We evaluate the participants' acceptance of the recommendations with different levels of openness. Figure 5.6 shows the distribution of *acceptance* with the change of openness. For the recommendation with the highest openness ("Public"), people have the lowest acceptance. This result indicates that people are less likely to accept the recommended location-privacy settings that would potentially overexpose their location information. To our surprise, people's acceptance of the "safest" recommendation, which is "Only Me", is the second lowest in our results. This result shows that in LSS, people not only consider their location privacy, but also care about the benefits of using LSS. To find out more evidence to support this finding, we look into the free-text comments in our participants' feedback. One of them says:

- "... if i (*sic*) would only share something to 'only me', then why would i (*sic*) share at all?"

This finding corresponds to the privacy calculus theory [32]. People's privacy-related decisions are decided by the trade-off between privacy and benefits. This means that they neither totally give up the benefits of LSS for 100% privacy, nor vice versa. When they can guarantee the benefits of using LSS, i.e., sharing with friends or friends of friends, the recommendations with lower level of openness are more likely to be accepted by them.

Our results suggest that location-privacy recommenders should be cautious with extreme recommendations. When the recommenders make extreme recommendations, we suggest that they should provide additional information, such as explanations of the recommendations or request for consent, to help people accept them. Meanwhile, it is also suggested to allow users to control the maximum openness that they want the recommenders to make. We find two free-text

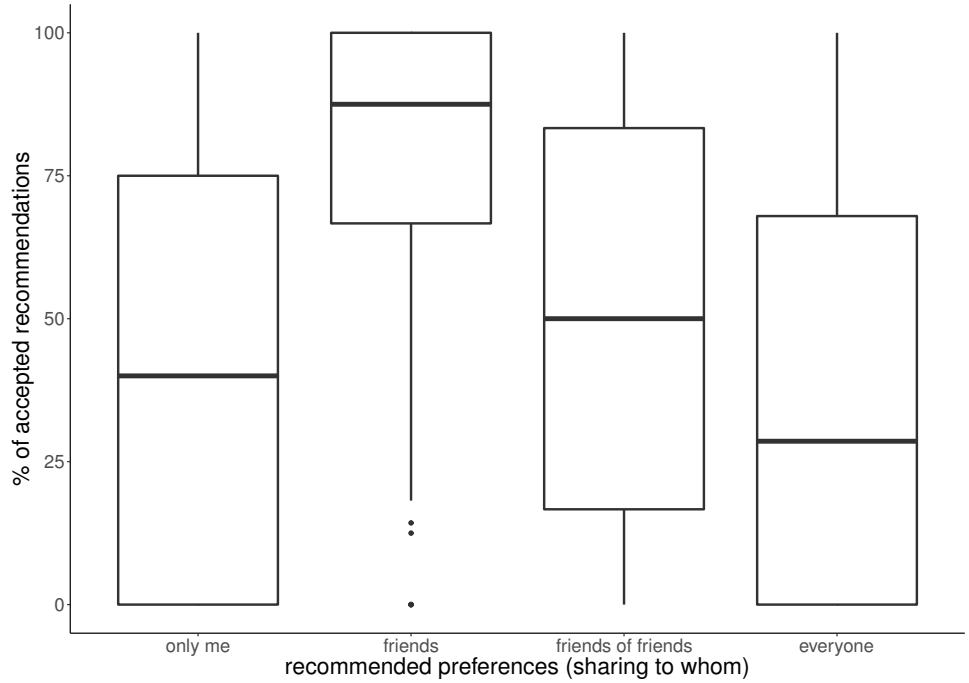


Figure 5.6: Distribution of all participants' acceptance (the percentage of accepted recommendations) of different recommended location-privacy preferences (sharing to whom). The least open preference (*only me*) and the most open preference (*everyone*) are least accepted. For the sharing-preferences (i.e. *friends*, *friends of friends*, and *everyone*), the less open preference is more accepted. (Friedman rank sum test: $chi-squared = 61.527, df = 3, p < 0.001$)

comments in our participants' feedback saying:

- "If the system had a 'never share publicly' option that would work best for my preferences. . . . " (*sic*)
- "There should be a 'maximum exposure' option . . . "

Additionally, some participants find that the openness of the four default settings is not fine-grained enough. For example:

- ". . . , however I would like some more customization. . . . "
- "The recmmenders should take into account the preferences I've set in the past."
- "Would need to learn a bit more about my own preferences as well as aggregating those from other sources to be useful for me."

Therefore, privacy recommender designers should also take into account customised recommendations.

5.6 The effect of contexts

Location-privacy recommendations are naturally with contexts (i.e., location category and time slot). People's acceptance may also be affected by these contexts. Thus, we are interested in finding out the effects from the contexts of location-privacy recommendations on people's acceptance.

The locations collected from participants' Facebook data have two parts of information, i.e., location names and location categories. We manually analyse all the location categories and merge similar ones with each other. We have four location categories, which are: *Entertainment*, *Residential*, *School/University/Library*, and *Transport*. For the time dimensions in contexts, we use the time slots in Table 4.4, which are: *morning* (07:00–11:59), *noon* (12:00–13:59), *afternoon* (14:00–16:59), *evening* (17:00–20:59), and *night* (21:00–06:59).

As shown in Figure 5.7, our participants have different acceptance of the recommended location-privacy settings in different contexts (two-way ANOVA to examine the interaction effect of time slot and location category: $F = 2.039, df = 12, p < 0.05$). More specifically, they are most likely to accept the recommendations in the location category *School/University/Library*. This result may be related to the occupations of the participants. As we advertise our experiment through university mail lists and university Facebook groups, we believe that our participants are mainly university students. The *School/University/Library* category has the locations where they spend most of their time everyday. Therefore, this result implies that our participants mostly accept the recommended location-privacy settings in the contexts where they spend their regular daily lives.

In Figure 5.7, the location category *Transport* experiences the lowest acceptance in the time slots *morning*, *noon*, *afternoon*. To better understand this effect, we look for evidence from the free-text comments. Two comments that may explain this result are:

- "... for example, I was at the airport. This informs all Facebook users that I will be away

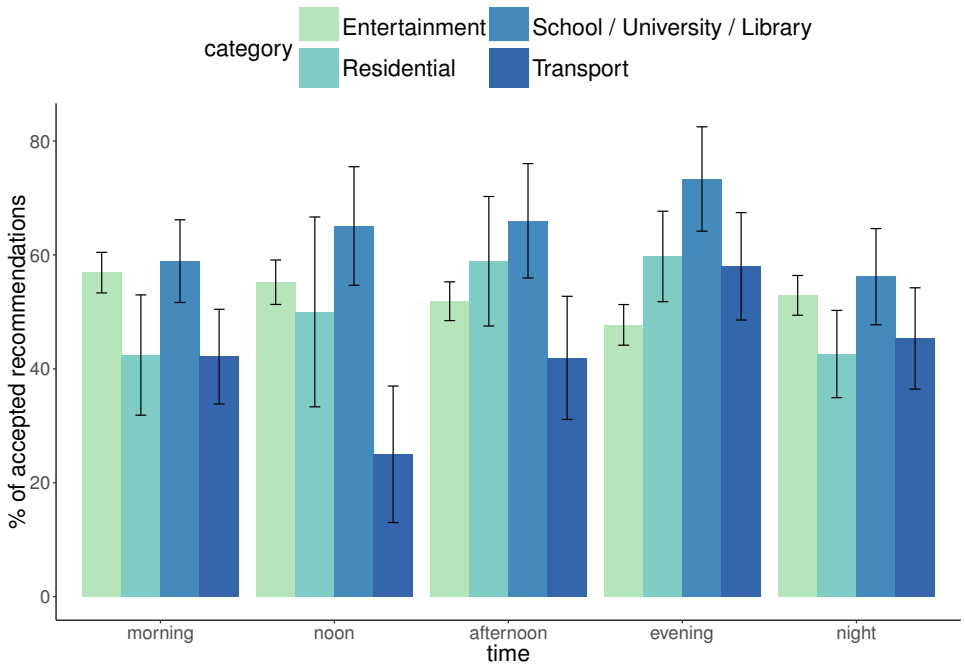


Figure 5.7: Participants' acceptance of the recommendations made for different location categories. For each time slot, our participants have the highest acceptance of recommended location-privacy preferences in the *School/University/Library* category, which is the most regular context for them (two-way ANOVA: $F = 2.039, df = 12, p < 0.05$).

from potentially a longer period of time than usual can could put myself at greater risk of property theft etc. . . . "

- "...I think a better recommender could consider sharing a place by how regular you go there or how far from where you normally are it is ie how exotic it is."

It appears that the first participant worries about the potential risks that a failed recommended location-privacy setting may cause, i.e., our recommender accidentally overexposing his or her location at an airport. For the second participant, the regularity of contexts affects his or her acceptance of the recommendations. This result corresponds to the finding that shows the highest acceptance in *School/University/Library*, which is the most regular location category for our participants. Therefore, we postulate that the **regularity** of contexts and the **potential risks** (e.g., being away from home for a long time and possible property theft) of overexposure may also affect people's acceptance of location-privacy recommendations.

We suggest that privacy recommender designers may need to allow users to choose in what

contexts they want to use location-privacy recommenders, or to tune recommendations according to contexts. Similarly, in areas such as mobile application recommendations, there are evidence [19] indicates that people's usage of mobile applications is also highly dynamic with contexts. Therefore, we also suggest that mobile application systems in such scenarios should be context-aware [18].

5.7 Summary

In this chapter, we have demonstrated the following:

- People have privacy concerns about providing their location-privacy preferences to a centralised recommender.
- People's privacy concerns have negative effects on their perceived recommendation quality, satisfaction about their choices, and acceptance of recommendations.
- The openness of recommendations affect people's acceptance. They are less likely to accept the recommendations with the highest openness or the lowest openness.
- Contexts (time slot and location category) affect people's acceptance.
- Regularity and potential risks caused by failed recommendations may have effects on people's acceptance.

Among these findings, we think that the negative effects from *concern* to the other factors are the most important. They affect both directly and indirectly on *acceptance*. To make location-privacy recommenders acceptable, we must address people's concerns about providing their data to a centralised recommender. Therefore, in the next chapter, we investigate the feasibility of two possible solutions that aim to protect people's data privacy from different angles.

Chapter 6

Alleviating concerns: data obfuscation and decentralisation

6.1 Introduction

Chapter 4 has demonstrated that user-based CF location-privacy recommenders are accurate. In Chapter 5, we have further evaluated the recommenders beyond accuracy, finding that several subjective and objective factors affect people's acceptance of location-privacy recommendations. Among these factors, people's privacy concerns about sharing their data with centralised recommender servers significantly decrease their acceptance of the recommendations, both directly and indirectly. Therefore, it is necessary to investigate how to alleviate such concerns.

One way to protect people's data privacy is through data obfuscation, i.e., adding noise into people's raw data. By adding noise that follows a certain distribution, the quality of recommendations may still be acceptable. Meanwhile, the centralised servers that use obfuscated data can only learn individual's information to a degree of probability, rather than 100%. Wang *et al*. [137] show that obfuscation options make end users more likely to provide their data to service providers. Thus, we examine the feasibility of data obfuscation in our location-privacy recommender systems.

Another choice, more directly, is to eliminate the existence of centralised recommender servers, which are the source of people's privacy concerns. In the scenario of LSS, people carry their

mobile devices with them and these devices physically meet each other. In addition, theses devices are embedded with short range communication interfaces such as Bluetooth, which means that they can exchange data directly with each other when they meet. Thus each device can have a local location-privacy recommender based on the data it received in client-side. Kobsa *et al*. [72] show that personalisations (e.g., recommendations) in client-side cause lower privacy concerns for users. Thus, we also examine the feasibility of deploying location-privacy recommender systems in a decentralised structure.

As in the decentralised structure, everyone can take part in the recommender system to receive and contribute data, the system is inherently vulnerable to malicious users who modify their received data and send them to other users. Therefore, we treat such misbehaviour as a type of attack and examine the its influence on decentralised recommender systems. To address its threats, we propose a reputation scheme based on encounter frequency to discriminate malicious users from *bona fide* users.

In this chapter, we investigate the feasibility of both data obfuscation and decentralisation in our system, and evaluate the performance of these two schemes. The aim of the experiments in this chapter is to answer the following questions:

- **Q**1 How much would data obfuscation influence the performance of recommendations?
- **Q**2 Can decentralised location-privacy recommender systems perform as well as a centralised recommender system does?
- **Q**3 How effective is the attack in decentralised recommender systems?
- **Q**4 How effective is the encounter-frequency-based reputation scheme to alleviate the attack?

For the data obfuscation scheme, as the added noise follows certain distribution, the influence of obfuscated data on the performance of recommendations may be averaged. Thus, we expect that we can protect users' data accuracy at limited cost of recommendation performance.

For the decentralisation scheme, as people in LSS only require location-privacy recommendations when they reach their destinations and want to share their locations, there may be enough time

for their devices to receive data from others before recommendations are required. Therefore, we expect that decentralised location-privacy recommender systems may eventually perform as well as a centralised recommender system does.

6.2 Methodology

We aim to investigate the performance of a centralised location-privacy recommender system with data obfuscation and the performance of a decentralised location-privacy recommender system. First, we describe the designs of these two schemes. Second, to evaluate the robustness of the decentralised system, we formally describe the above mentioned misbehaviour as a type of shilling attacks [79], i.e., the *sampling attack* [23], which can be applied to biasing decentralised recommender systems, and demonstrate its attack effectiveness. Finally, we introduce a reputation scheme and demonstrate its alleviation effect on the sampling attack.

6.2.1 Centralised recommender systems with data obfuscation

When deploying location-privacy recommenders in real-world applications, users have to share their data with the recommenders. A third party that is independent of LSS applications could serve as the recommender, in order to avoid recommendations being biased by the LSS applications for their own benefits. We assume such third-party recommender is semi-honest, which means that the recommender conforms to algorithms to generate recommendations, but try to further analyse users' data to learn additional information from them. Users' location-privacy preferences are recorded as rating vectors. These vectors not only contain their location-privacy settings, but also contain their location information. For a specific context in a rating vector, if it is rated, then that means the user who owns the rating vector has been to the location category during the time slot of the context. Thus, when users require location-privacy recommendations for new contexts, they have to release their location histories that may have some sensitive contexts to the recommender.

One way to make the recommenders provide recommendations without knowing the content in preferences is using homomorphic encryption [25]. Its computational expense, however, is too high for applications using large data sets in the real world. Another way is to obfuscate

the data in preferences. Data obfuscation is normally done by adding noise, i.e., fake ratings, into users' preferences. The added noise follows a certain distribution that makes the loss of performance not significant. By this means, the privacy of individual users is guaranteed at the cost of acceptable recommendation accuracy loss.

We implement the data obfuscation in our centralised recommender system based on the work of Polat and Du [109], which adds fake ratings for unrated items in the preferences. By this means, the recommender cannot tell if a rated item in a preference is real. For each user, there are a number of rated items in his or her preference. We denote this number as m_t. The user can decide the amount of noise to be added in the preference by controlling a noise factor α. The maximum number of fake ratings to be added is denoted by $m_{max} = \alpha m_t$. The actual number of fake ratings, which is denoted by m_f, is randomly selected between 0 and m_{max} and $m_f \sim U[0, m_{max}]$. We randomly select m_f unrated items in the preference and give half of them positive ratings (i.e., "share") and give another half negative ratings (i.e., "not share"). The purpose of increasing the randomness in the process is to decrease the loss of recommendation accuracy.

6.2.2 Decentralised recommender systems using opportunistic networks

We design our decentralised location-privacy recommender system based on two assumptions. First, in the scenario of LSS, users often move around with their mobile devices (e.g., smartphones) and often encounter each other. All of these mobile devices construct an *opportunistic network* [108] and we can build our decentralised recommender system based on it. This opportunistic network enables people to exchange their location-privacy preferences directly when they encounter. Thus, such structure does not need the support from a centralised server. Second, when using LSS applications, people only request location-privacy recommendations when they arrive at a place and decide to publish a location check-in. Thus before one recommendation is requested, there may be long enough time for one's device to receive adequate data from others on the way from one place to another. In Figure 6.1, we demonstrate an example of how a decentralised location-privacy recommender works on Alice's device.

We describe our decentralised recommender system based on the description in Chapter 4. We are interested in investigating how the performance of decentralised recommender changes with

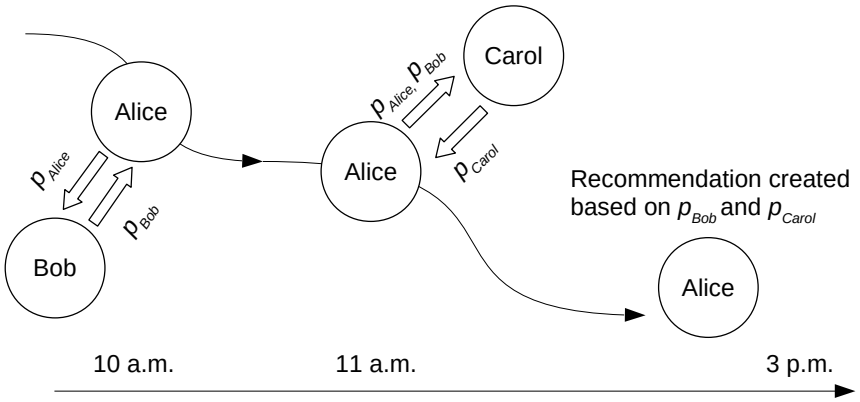


Figure 6.1: At 10 a.m., Alice's device and Bob's device encounter (move within communication range) each other and they exchange their stored location-privacy preferences. Then at 11 a.m., Alice's device encounters and exchanges data with Carol's device. When Alice arrives at her destination at 3 p.m. and wants to share her location, the system uses the data that she received from Bob and Carol to generate a location-privacy recommendation on her device locally.

time. Thus, we put users' location-privacy preferences in their profiles and add timestamps in these profiles. For each user profile p, $p = (id, R, ts)$. id is the profile identity. R is the user's location-privacy preference. ts is the timestamp of the last update time for the profile.

We assume that our decentralised recommender system is in the same LSS scenario where the centralised one mentioned in Chapter 4 is. The context set $C = T \times L = \{c_1, c_2, ..., c_{|T||L|}\}$. Each location-privacy preference is represented as $R = (r_{i,1}, r_{i,2}, ..., r_{i,|T||L|})$. Each $r_{i,x}$ is the privacy setting, i.e., "share" or "not share", of user u_i in context c_x.

The initial value of a profile p's timestamp ts is the time when p is generated. Every time when users update their profiles, such as making new location-privacy decisions, we update the ts to the current time. Table 6.1 shows an extended version of all the terms that we use to describe our decentralised location-privacy recommender system.

In our system, users exchange their profiles, which contain their location-privacy preferences, with each other through opportunistic networks. Each user u_i keeps all the received profiles from others in a set $P_i^{received}$. When two users, say u_i and u_j, encounter each other, they have two data

For users	
u	a user
U	the set of all users
t	a time slot
T	the set of all time slots
l	a location category
L	the set of all location categories
C	the context set, $C = T \times L = \{(t_1, l_1), (t_1, l_2), ..., (t_{\|T\|}, l_{\|L\|})\}$ $= \{c_1, c_2, ..., c_{\|T\|\|L\|}\}$
R	a location-privacy preference, $R_i = (r_{i,1}, r_{i,2}, ..., r_{i,\|T\|\|L\|})$, $r_{i,x}$ is u_i's location-privacy setting ("share" or "not share") in c_x
p	a real profile of a user, $p = (id, R, ts)$, id is the profile identity, R is the user's location-privacy preferences, and ts is the profile's last update time
P^{real}	the set all real profiles, $p_i \in P^{real}$
$P^{received}$	a set of received profiles
For attackers	
C^{target}	a target context set
int	the intent of attack ($push$ or $nuke$)
s	a shill record, $s = (id^{real}, id^{shill}, ts)$, id^{real} is the real id of the affected profile, id^{shill} is the id of the shill profile made from the affected profile, and ts is the last update time of the affected profile
S	a shill record set
P^{shill}	a shill profile set

Table 6.1: Terms and symbols used in our system

exchange schemes:

- *Decentralised Individual Exchange (D-Ind)*: u_i and u_j only exchange p_i and p_j.

- *Decentralised Set Exchange (D-Set)*: u_i and u_j exchange their own profiles and all their received profiles with each other, i.e., u_i sends p_i and $P_i^{Received}$ to u_j, and u_j sends p_j and $P_j^{Received}$ to u_i.

Users update their stored profiles after they receive profiles from others. In the *D-Ind* scheme, after u_i receiving p_j, u_i checks if p_j already exists in $P_i^{Received}$. If u_i does not have p_j in the

received set, or if u_i finds out that the received p_j is newer than a previously existing p_j (u_i can do this by comparing the timestamps of two profiles), then u_i adds p_j into $P_i^{Received}$, or updates the existing p_j.

In our system, people travel from one place to another. Once they arrive at their destinations during a certain time slot, if they want to publish location check-ins, they can use all the received profiles on their devices to recommend location-privacy settings. These recommendations are calculated locally on their devices without the support from a centralised server.

The recommendation algorithm of our decentralised recommender is user-based CF [113], the same as the algorithm of our centralised recommender in Chapter 4. When user u_i requests a recommendation, first, we calculate the similarities of the location-privacy preferences in all the profiles in $P_i^{Received}$, in the same way described in Chapter 4. Next, we use the neighbours with the highest similarities to R_i to recommend a location-privacy setting for u_i. We use the same algorithm and decision threshold as in Chapter 4.

6.2.3 Conducting sampling attack in decentralised recommenders

Our proposed decentralised recommender system is based on opportunistic networks. Everyone who has a mobile device can take part in the system and produce, receive, and relay data in it. This type of open structure is inherently vulnerable to malicious users who misbehave for their own benefits.

One potential attack against our decentralised recommender system is the sampling attack. Sampling attackers use real users' preferences as samples to generate many fake profiles (i.e., shill profiles) with elaborated ratings, in order to bias recommendation results. In centralised recommender systems, the sampling attack is considered impractical, since attackers rarely can access real users' profiles stored on a server. In decentralised recommender systems, however, as users receive and store others' data, the system is more vulnerable to the sampling attack, compared with centralised systems. Malicious users can use their received data as samples to generate shill profiles. These shill profiles are highly similar to the real profiles of victims, which increases the difficulty of detecting them based on similarities. An example of an attacker conducting sampling attack is shown in Figure 6.2.

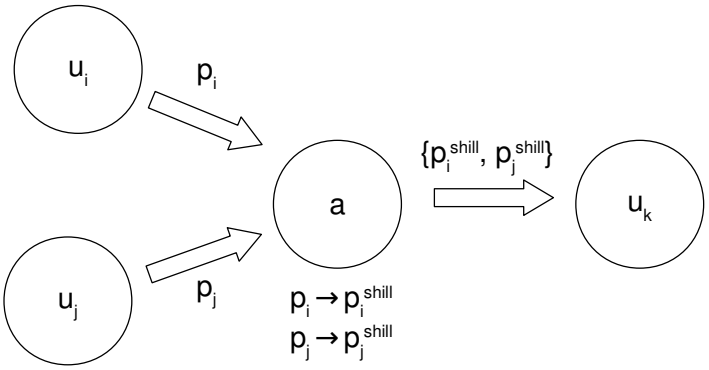


Figure 6.2: A sampling attacker a receives two profiles p_i and p_j from two real users u_i and u_j respectively. The attacker a then uses p_i and p_j as samples to generate two shill profiles p_i^{shill} and p_j^{shill}. When a meets u_k, it sends the two shill profiles $\{p_i^{shill}, p_j^{shill}\}$ rather than the two real profiles $\{p_i, p_j\}$ to u_k.

The incentives behind the sampling attack in our system can be various. As the function of location-privacy recommender systems is to help people with their location-privacy settings by automatic configuration, malicious users may use the sampling attack as a way to bias the recommendations in the systems, thereby overexposing (or underexposing) people's location information. For instance, business owners may want people who visit their shops to share the locations there, in order to increase the number of location check-ins in their shops on social media platforms (e.g., Facebook). Such attack can increase the popularity of their own business. Similarly, for their rivals' places, they may conduct the sampling attack to make people who visit there not share their locations.

Before we formally describe the process of sampling attack, we define the ability of the attackers in our system as follows:

- Attackers can take part in the decentralised recommender system to receive data from other users, to store the data on their devices, and to inspect the data.

- Attackers can generate multiple *id*s for multiple shill profiles. These shill profiles are made based on the received profiles from real users.

Each attacker has some target contexts to attack, i.e., at which location category and during which time slot the attacker wants the sampling attack to work. For each target context, the attacker has an intent of attack, which represents that the attacker wants to encourage people in the target context to publish location check-ins, i.e., *push* attack, or wants to discourage people to check in, i.e., *nuke* attack.

After deciding the set of target contexts, i.e., C^{target}, and the intent of attack, i.e., *int*, the attacker joins the decentralised recommender system and encounters with other users. Every time when the attacker meets a user, they exchange data with each other. Using the received profiles as samples, the attacker generates shill profiles. Based on *int*, these shill profiles give "share" or "not share" ratings to the contexts in C^{target}. We formally describe the process of sampling attack in the *D-Ind* scheme in Algorithm 1. This is only for generating shill profiles for one received profile. When conducting the sampling attack in the *D-Set*, the attacker repeats Algorithm 1 for all the profiles in the received list.

6.2.4 Encounter-frequency-based reputation scheme

To solve the threats from sampling attackers, we propose a reputation scheme that uses the encounter frequency of devices to discriminate shill profiles from real profiles. Traditionally, shill profiles in other types of shilling attacks, such as *random attack*, can be discriminated by analysing the distribution of their similarities. The premise of similarity-based detection is that attackers cannot access to the samples of real users' preferences. Thus, the distribution of the preference similarities in shill profiles is different from the distribution of the preference similarities in real profiles, because attackers generate shill profiles randomly or through some simple rules. This premise is reasonable in centralised recommender systems, but not in decentralised structures.

In the sampling attack, shill profiles are made from the profiles of real users. The only difference between a shill profile and its original profile is the ratings in target contexts. Therefore, detecting them through similarity-based schemes is difficult. In addition, it is not costly for attackers to produce many shill profiles with small differences in similarities to pass any thresholds of similarities. Thus, we seek solutions that use other features rather than similarities.

Algorithm 1: Sampling attack. Generating new shill profiles or replacing old shill profiles.

Data: p is a received profile.
S is a set of victim records.
C^{target} is the set of target contexts.
int is the intent of the attacker on the target contexts.
Result: P^{shill} is the set of shill profiles.
begin
  if $\nexists s \in S, s.id^{real} = p.id$ **then**
    `/* Generating a new shill profile from` p `*/`
    $id^{shill} \leftarrow$ an id for a new shill profile;
    $S \leftarrow S \cup \{(p.id, id^{shill}, p.ts)\}$;
    $R^{shill} \leftarrow p.R$;
    foreach *c in* C^{target} **do**
      if *c is not rated in* R^{shill} **then**
        Change its rating in R^{shill} based on *int*;
    $p^{shill} \leftarrow (id^{shill}, R^{shill}, p.ts)$;
    $P^{shill} \leftarrow P^{shill} \cup \{p^{shill}\}$;
  else
    if $p.ts > s.ts$ **then**
      `/* Replacing the old shill profile of` p `*/`
      $p^{old} \leftarrow$ the old shill profile of p;
      $p^{shill} \leftarrow$ the new shill profile of p, made from $p.R$, C^{target}, *int*, and $p.ts$;
      $Pshill \leftarrow P^{shill} \setminus \{p^{old}\} \cup \{p^{shill}\}$;
      $s.ts \leftarrow p.ts$;

As described above, the mobile devices in our system compose an opportunistic network and they physically encounter each other. The encounter frequency of all the devices are decided by the number of devices and their mobility patterns. If we assume that all the devices in our system have similar mobility patterns, for individual devices, their frequency of encountering others should be similar. Thus, we can use encounter frequency as a type of resource, i.e., *reputation*, and it is bounded with devices rather than profiles. If a sampling attacker generates multiple profiles on his or her device, the encounter-frequency-based reputation has to be divided for these shill profiles. To increase the reputation, the attacker has to deploy more devices carried by multiple persons with different mobility trajectories. Compared with simply generating more shill profiles, this is more expensive and more difficult for the attacker. Therefore, we use encounter frequency to build our reputation scheme.

Figure 6.3 shows an example of our reputation scheme. In Figure 6.3 (a), there are two real users, i.e., u_1 and u_2, and they encounter with each other three times. As a real user, u_2 only keeps one profile, i.e., p_2, on its device. Every time, when encountering with u_1, u_2 identifies itself as p_2. Thus from the perspective of u_1, the profile p_2's reputation is 3. But if u_2 is an attacker a, as shown in Figure 6.3 (b), who has three shill profiles on its own device, it can only identify itself as one of these three shill profiles. Therefore, given similar amount of encounter frequency, from u_1's perspective, the reputation of the shill profiles of attacker a would be lower than the reputation of the profiles of real users, since a has to divide the reputation for different shill profiles.

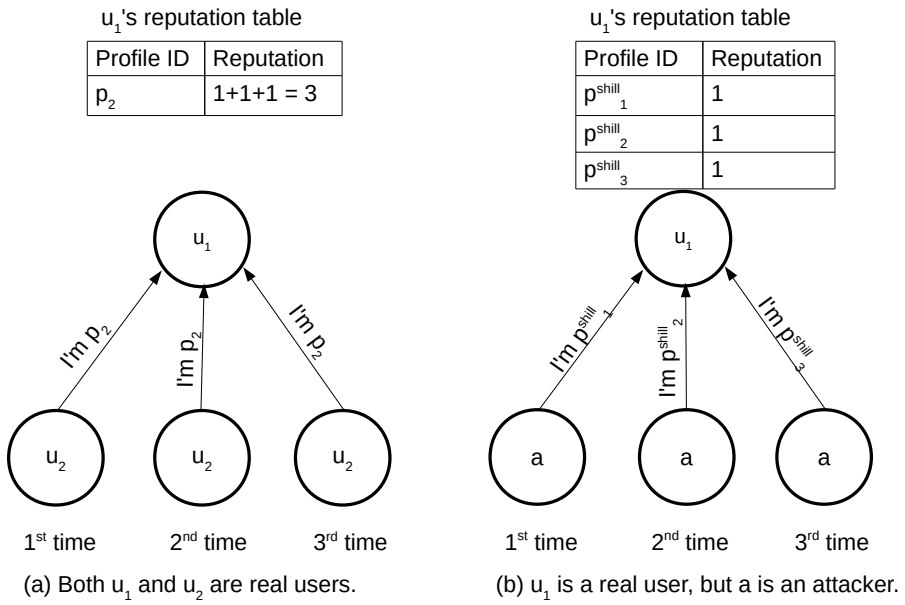


Figure 6.3: Encounter frequency based reputation scheme. In (a), real users u_1 and u_2 encounter three times. Each time, u_2 identifies its profile p_2 with u_1 and the reputation of p_2 is 3. In (b), a is an attacker and has to divide the same amount of reputation to its different shill profiles.

We define the reputation of profile p_j from user u_i's perspective, i.e., $rep_{i,j}$, as the frequency by which u_i encounters p_j. We use such encounter-frequency-based reputation to discriminate shill profiles from real profiles. For u_i, before generating recommendations locally, we need to decide a threshold to filter out profiles with low reputation values. We draw on the threshold value of the trust-based filtering [99], i.e., the mean of trust values, and calculate the average reputation

of all the received profiles, i.e., $\overline{rep_i}$, as the threshold to filter out low reputation profiles in our scheme. Therefore, the set of profiles that takes part in making recommendations is:

$$P_i^{candidate} = \{p_j | p_j \in P_i^{received}, rep_{i,j} \geq \overline{rep_i}\}$$

According to the design of our reputation scheme and our assumption that devices have similar mobility patterns and encounter frequency, as long as an attacker has multiple shill profiles, it has to divide the opportunities of gaining reputation among those shill profiles. Therefore, each shill profile has lower reputation than a real profile does. One exception is that the attacker may generate only one shill profile and this shill profile may have similar amount of reputation as a real profile does. Our reputation scheme cannot discriminate this kind of single shill profiles from real profiles. However, the influence of a single shill profile on biasing recommendation results is low. To increase the attack effectiveness, the single-shill-profile attacker has to deploy multiple devices to increase the number of shill profiles in the entire system, which increases the expense of conducting such attack.

6.3 Results of data obfuscation

To examine the influence of different amount of noise on recommendation performance, we use different noise factor α. For each one, we run 100 rounds of 10-fold cross-validation experiments. The recommendation performance without noise is used as the benchmark. Comparing it with the performance with noise can tell us the loss in *accuracy* and the increase in *leak*. One the one hand, the more noise we add, the less probably a recommender can tell whether a rating in a preference is real. On the other hand, more noise causes more loss of the recommendation performance. We aim to find out the trade-off between performance and privacy.

As shown in Figure 6.4, the x-axis represents the noise factor α. It changes from 1 to 20. The y-axis represents the *accuracy* of our recommender under the influence of the added noise. The dashed line is the *accuracy* of our recommender without noise. The loss of *accuracy* goes up (from 0.76% to 5.35%) when we increase α. This is within our expectation of the influence of added noise. To find out how likely a real rating can be identified, we define *Privacy Level* (PL)

as the percentage of the expected value of fake ratings among all ratings, i.e., $PL = \frac{E(m_f)}{E(m_f)+m_t}$. As we mentioned, $m_f \sim U[0, m_{max}]$, then we have $E(m_f) = \frac{m_{max}}{2} = \frac{\alpha m_t}{2}$. Thus, given a noise factor α, we have a PL as $\frac{\alpha}{\alpha+2}$. For a recommender without any noise, i.e., $\alpha = 0$, the PL is 0, which means the recommender knows every rating in the preference is real and the owner of the preference has been to the context related to the rating. When $\alpha = 1$, the PL is 33.33%. This result is at the cost of only 0.76% loss of *accuracy*. Similarly, as shown in Figure 6.5, adding noise causes the increase of *leak* from 0.86% to 2.21%. The trade-off between *leak* to PL is also minimal.

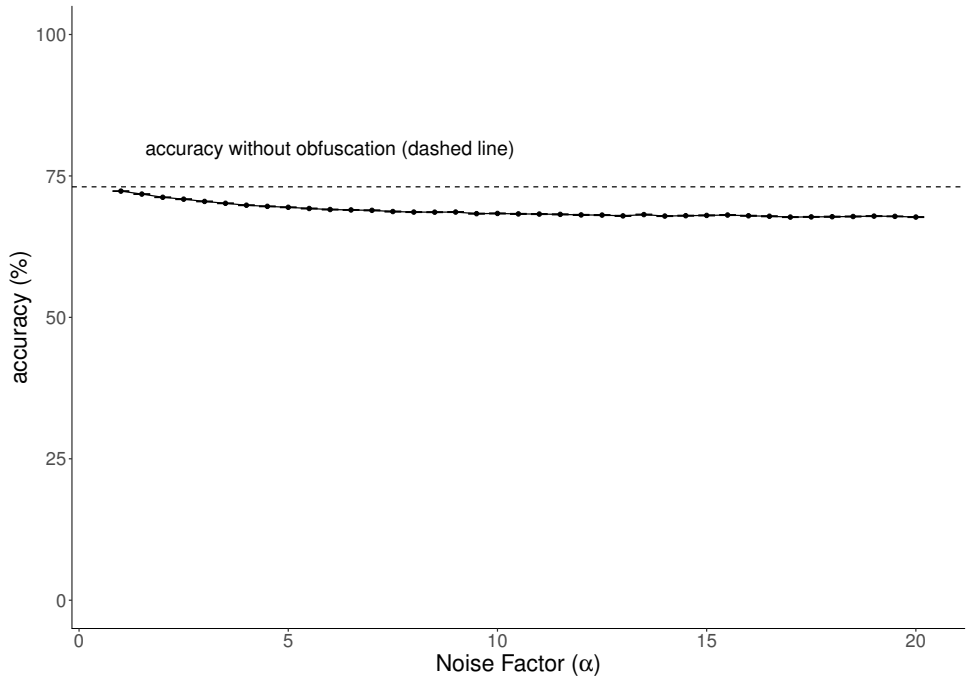


Figure 6.4: *accuracy* of the privacy-aware recommender with different noise factor (α). The dashed line represents the *accuracy* of the recommender without fake ratings. The loss of *accuracy* is minimal (0.76%, $\alpha = 1$) when α is small. It increases with the growth of α and reaches 5.35% when $\alpha = 20$.

It should be noted that, when $\alpha = 20$, the theoretical expected number of fake ratings is ten times as the number of real ratings. However, since there are 5 time slots and 6 location categories in the dataset, the length of each preference is uniform and fixed, i.e., 30. Therefore, when the number of fake ratings and real ratings in a preference reaches the maximum length of the preference, increasing α will not increase the number of fake ratings. And that is the reason of the influence of noise being stable as α increases. This suggests that small α is effective to

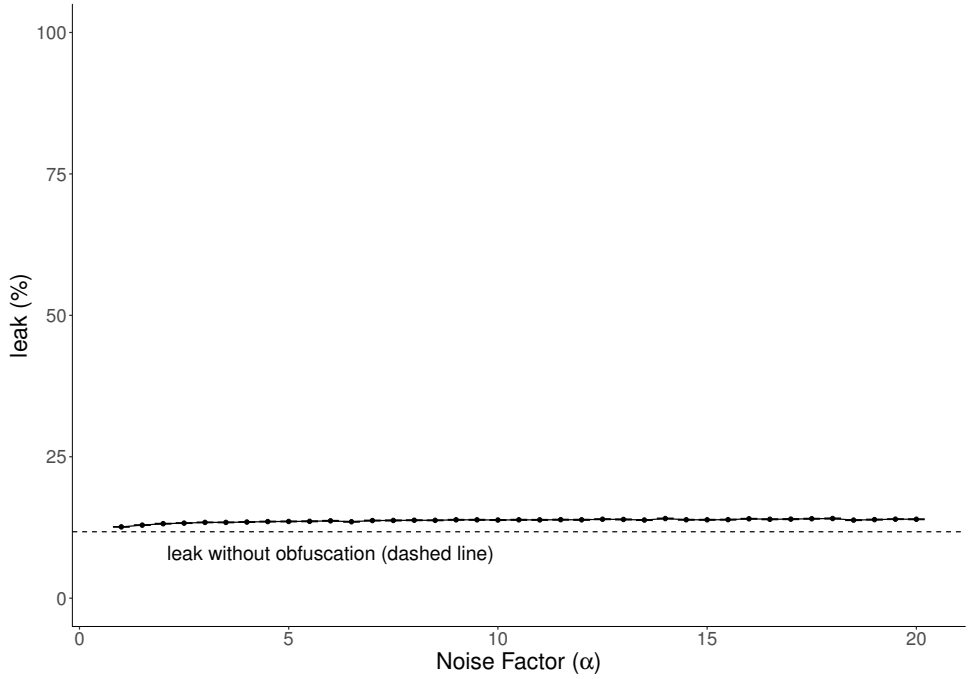


Figure 6.5: *leak* of the privacy-aware recommender with different noise factor (α). The dashed line represents the *leak* of the recommender without fake ratings. The increase of *leak* is minimal (0.86%, $\alpha = 1$) when α is small and reaches 2.21% when $\alpha = 20$.

increase the expense of the recommender to learn whether a rating is real.

This result indicates that our centralised location-privacy recommender system can be implemented in a privacy-aware fashion. By adding noise that follows uniform distribution, we can increase the privacy level for recommender users at the cost of minimal performance loss.

6.4 Performance of decentralised recommender systems

To evaluate the performance of our decentralised location-privacy recommender system, we conduct our experiment through opportunistic network simulations driven by real-world traces. We use the Opportunistic Network Environment (ONE) simulator [67] to realise the simulation. In each round, we simulate 24 hours and divide this time span into five time slots (i.e., morning, noon, afternoon, evening, and night) as shown in Table 6.2. We use the same dataset of location-privacy preferences that we use in Chapter 4, i.e., the st_andrews/locshare from the CRAWDAD data archive [103]. Since there are 40 participants in the dataset, we deploy 40 nodes in our

simulation and each of them has a different location-privacy preference selected from the dataset.

time slot	time range	probability
Morning	0700 – 1159	18%
Noon	1200 – 1359	13%
Afternoon	1400 – 1659	23%
Evening	1700 – 2059	28%
Night	2100 – 0659	17%

Table 6.2: Probability of check-in in different time slots

The mobility patterns of the nodes in our simulation are restricted to a map that represents the road layout of the town of St Andrews. Every time a node moves, it first chooses a place as its destination on the map. We have five points of interests (POI) to represent the places, including university buildings and night clubs, where people are more likely to visit. A node has the probability of 0.8 to choose a POI as its next destination. The node travels through the shortest path on the map to its destination. The node stays at the place between 0 seconds and 120 seconds after arriving. After that, it chooses the next destination based on the same rules. We repeat 100 rounds of simulations and randomly generate the nodes' initial positions and mobility patterns by using different randomness seeds. The detailed simulation configuration of our experiments is shown in Table 6.3.

Parameters	Values
simulation time	86400 seconds (24 hours) / round
time update interval	2 seconds
transmit range	10 metres
number of nodes	41 (40 real users, 1 attacker)
walking speed	0.0 m/s to 1.5 m/s
number of points of interests	5
probability of visiting POIs	80%
world size	4500 metres * 3400 metres
movement map	streets of St Andrews
movement model	shortest-path map-based movement
wait time	0 seconds to 120 seconds
router	direct delivery
number of rounds	100

Table 6.3: Simulation configuration

In our simulation, we do not consider the influence of data transmit speed between two nodes and

the storage size on each node. In our recommender, the main payload of messages is preference sets. These preferences are represented as binary vectors. The expense of transmitting and storing such preferences, we believe, is not influential. Thus, once two nodes encounter with each other, the data exchange between them can be done in one time interval of our simulation (2 seconds).

Compared with the off-line evaluation in Chapter 4, we evaluate our decentralised recommender system on the fly. The reason is that we want to investigate how the performance would change with simulation time. We start each round of simulation from 0700 in the morning. All nodes keep travelling and encountering with others. Figure 6.6 shows the cumulative distribution function (CDF) of the encounter frequency in 100 rounds of simulation. As the nodes' movement patterns are restricted by the road layout of small town and there are several points of interest where the nodes are likely to visit at the same time, the encounter frequency in larger areas with the same number of nodes may be lower than that in our simulation and the nodes may need longer time to receive enough data. However, in a real world scenario, there might be more nodes with more complex movement patterns (e.g., using public transport to move faster), which may increase the density of nodes and the encounter frequency in some areas.

A node may decide whether publish a location check-in once it arrives at a destination. During different time slots of the simulation, the probabilities of publishing location check-ins are different. We calculate the probabilities from the percentage of instances of each time slot in the dataset, as shown in Table 6.2. If the node decides to check-in, we request a location-privacy recommendation. We first convert the current simulation time to the time slot it belongs to, based on Table 6.2. If there are any settings in the node's allocated location-privacy preference for the current time slot, then we randomly choose one from them and use it as the ground truth to be compared with the recommendation to be made. The possible combinations are the same as in the evaluation section of Chapter 4, and we use the same metrics, i.e., *accuracy* and *leak*. Once a setting has been chosen for evaluation, it will not be chosen again in the same round of simulation. After the comparison between the ground truth setting and the recommendation, the node's profile is updated by adding the tested setting in it and updating the timestamp to the current time. The node uses this updated profile for data exchange and making recommendations

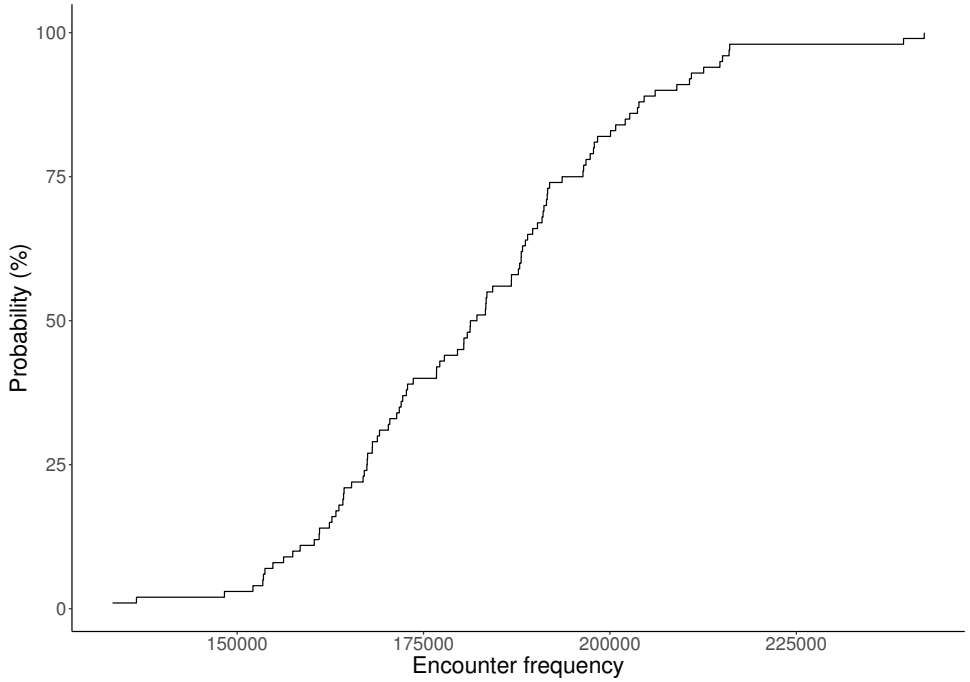


Figure 6.6: Cumulative distribution function of the encounter frequency of 100 rounds of simulation. Each round of simulation has 41 nodes (40 real users and 1 attacker) and 24 hours simulation time.

in the rest of this round of simulation. We use the same recommender engine, i.e., Lenskit [39], as we do in Chapter 4. The maximum neighbourhood size is 8, which has the lowest *leak* in the evaluation in Chapter 4.

We compare the *accuracy* and the *leak* of our decentralised schemes, i.e., *D-Ind* and *D-Set*, with a centralised recommender that can access all the data, i.e., *C-Rec*. As *C-Rec* always uses the whole data of all the nodes in the simulation to make recommendations, it has the ideal performance that a recommender can achieve. Thus we use its *accuracy* and *leak* as benchmarks.

As shown in Figure 6.7 and Figure 6.8, *D-Ind* and *D-Set* have similar performance. The *accuracy*s of *D-Ind* and *D-Set* are 9% and 11% lower than that of *C-Rec* and their *leak*s are 4% and 3% higher than that of *C-Rec*. In our decentralised schemes, once a profile is generated, the holder of the profile needs to move and encounter with other nodes to send this profile to them. Thus the nodes in our decentralised schemes need more time than those in the centralised scheme to receive adequate data to make accurate recommendations. Therefore, the difference between the performance of decentralised and centralised recommenders is within our expectation.

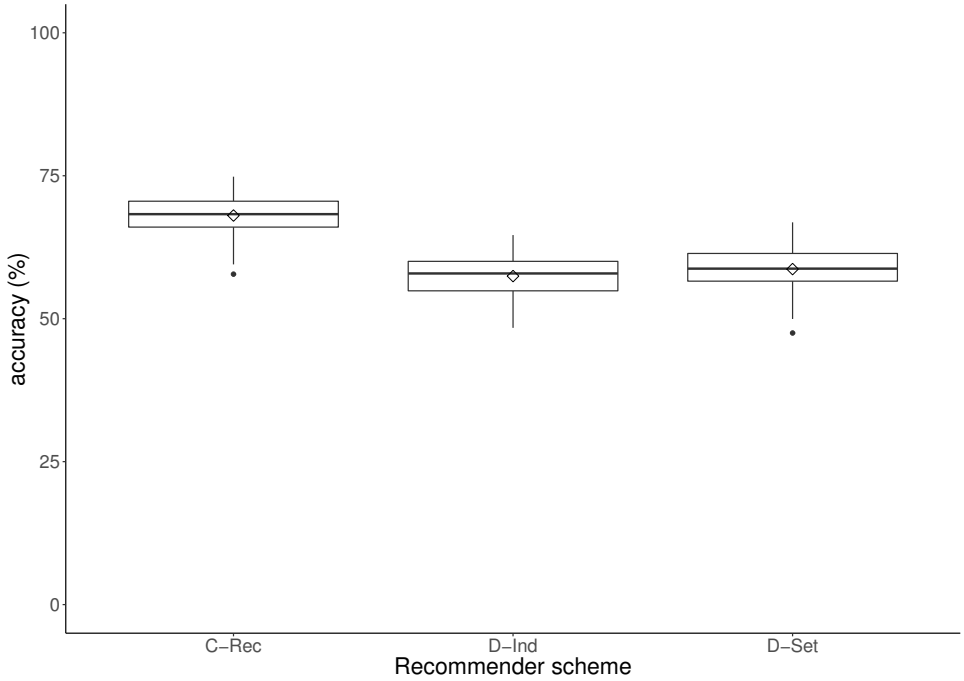


Figure 6.7: Overall *accuracy* of different recommenders in 100 rounds of simulations. The centralised recommender (*C-Rec*) has the highest average recommendation accuracy (68%) compared with the decentralised recommenders' *accuracy* (both $p < 0.01$, t-test). For the decentralised recommenders, the average *accuracy* of *D-Ind* is 57%, 2% lower than the *accuracy* of *D-Set*, 59% ($p < 0.01$, t-test).

Among the decentralised schemes, *D-Set* has higher *accuracy* and lower *leak* than *D-Ind* does. To find out the reason, we measure the *profile coverage* of both schemes. The *profile coverage* is drawn from the *message coverage* metric from existing work [50]. For each profile, we define its coverage as the number of nodes that have received this profile, divided by the number of nodes that should receive this profile. In our simulation, since we deploy 40 nodes, for each new generated profile, there are 39 nodes that are supposed to receive it. In each round of simulation, we measure the average *coverage* of all the profiles and record the average values of all the 100 rounds of simulations. As shown in Figure 6.9, the *profile coverage* in *D-Set* is higher than that in *D-Ind*. In *D-Set*, since the nodes send not only their own profiles, but also their received profiles to other nodes, once a new profile is generated, it can cover more nodes in *D-Set* than in *D-Ind*, due to the relay. Thus, the performance of *D-Set* is better. Therefore, we suggest that the *D-Set* scheme is a better candidate than *D-Ind* to be deployed in real-world deployment. In the rest of this chapter, we only choose *D-Set* to represent our decentralised recommender system and use it

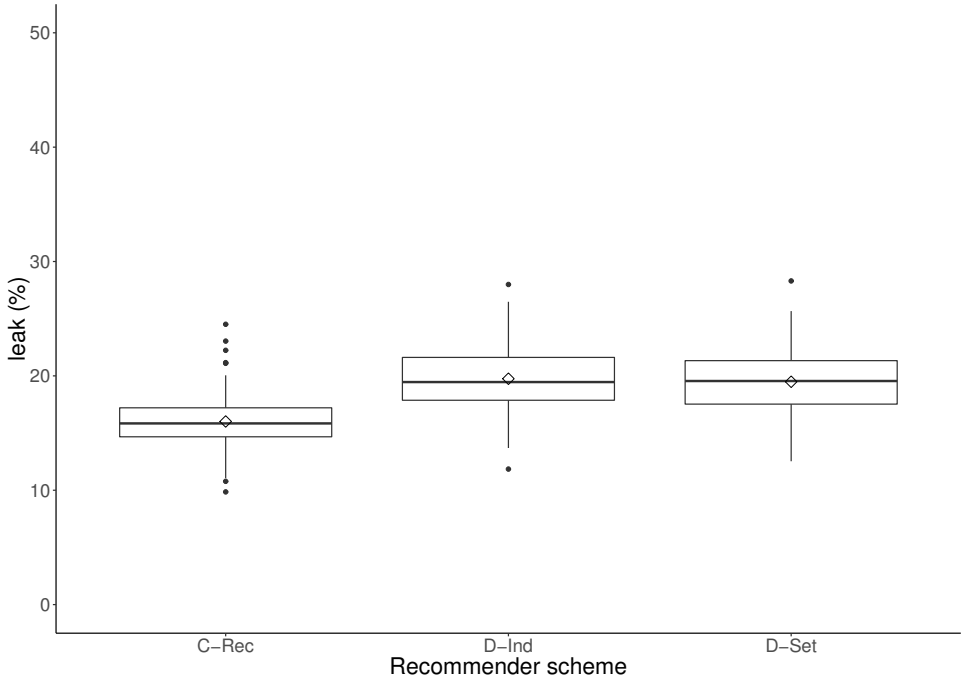


Figure 6.8: Overall *leak* of different recommenders in 100 rounds of simulations. As in *accuracy*, the centralised recommender (*C-Rec*) has the best performance, with an average *leak* of 16% (both $p < 0.01$, t-test). The difference between *D-Ind* (20%) and *D-Set* (19%) is not statistically significant ($p > 0.01$, t-test).

to compare with other schemes.

The overall performance only shows the difference at the end of each round of simulation. But during each simulation, the performance may change with time. As the time of simulation goes, the nodes in our decentralised recommender system collect more and more data, and the performance may get close to that of the centralised one. To find out the change of performance with time, for each round of simulation, we group the evaluation results into 90-minute buckets, which have the smallest bucket size that makes all the buckets have sufficient data. We calculate the *accuracy* and the *leak* of different schemes in each time interval to see whether the difference between them changes with time.

As shown in Figure 6.10 and Figure 6.11, at the beginning of the simulation, which is the cold-start period, the difference between the performance of *D-Set* and *C-Rec* is the greatest. The performance of *D-Set* approaches the performance of *C-Rec* as the simulation time goes on. In fact, the average *accuracy* difference and the average *leak* difference between the two schemes

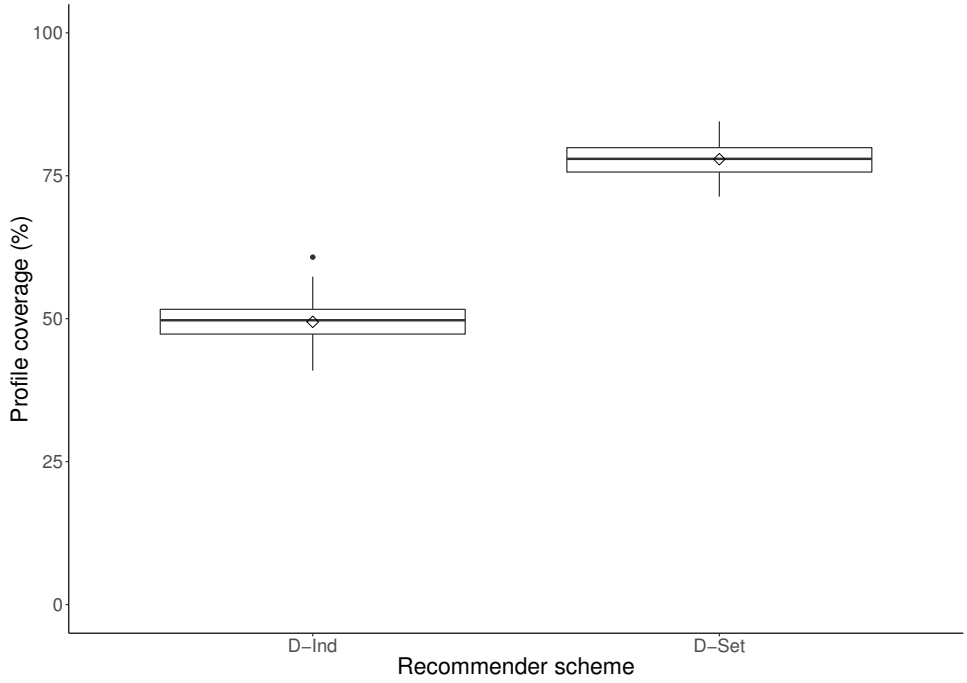


Figure 6.9: Overall profile *coverage* of decentralised recommenders in 100 rounds. Due to the data forwarding, the average *coverage* (78%) of *D-Set* is higher than the average *coverage* (49%) of *D-Ind* ($p < 0.01$, t-test).

are 3% and 1% respectively, after 4.5 hours of simulation time. There are a few statistically significant differences in the comparison of performance after 4.5 hours of simulation time, such as the 9th hour in Figure 6.10 and the 15th hour in Figure 6.11. The reason is that, in different time slots, all the nodes in the simulation only request recommendations of the current time slot. Once the simulation enters the next time slot, as no one has the knowledge about the recommendations in this new time slot, both the centralised and the decentralised systems experience a new cold-start period. During this period, the profile coverage and performance of the centralised system can recover more quickly than the decentralised system.

In real-world LSS scenarios, it is unlikely that people publish all their location check-ins and request location-privacy recommendations within one day. Thus, there would be more time for the nodes in our decentralised recommender system to receive adequate data before recommendations are requested. Therefore, we believe that the performance of our decentralised recommender system would be closer to that of the centralised one in real-world LSS applications than in our simulation.

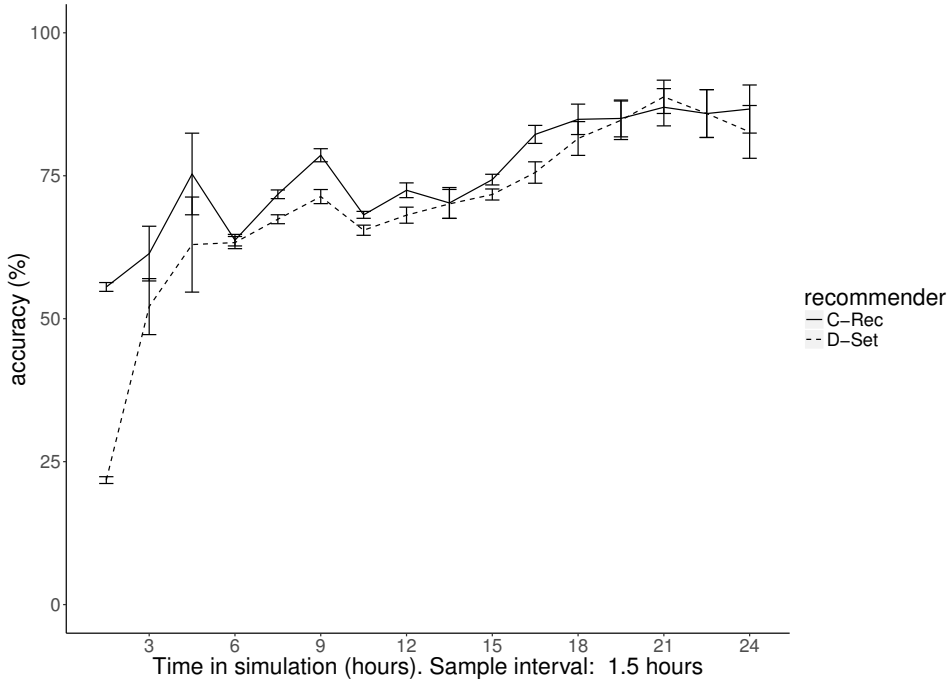


Figure 6.10: *accuracy* of *C-Rec* and *D-Set* over time. The difference between the two schemes' *accuracy* becomes small as the simulation time goes on. After 4.5 hours of simulation time, the *accuracy* of the two schemes are close.

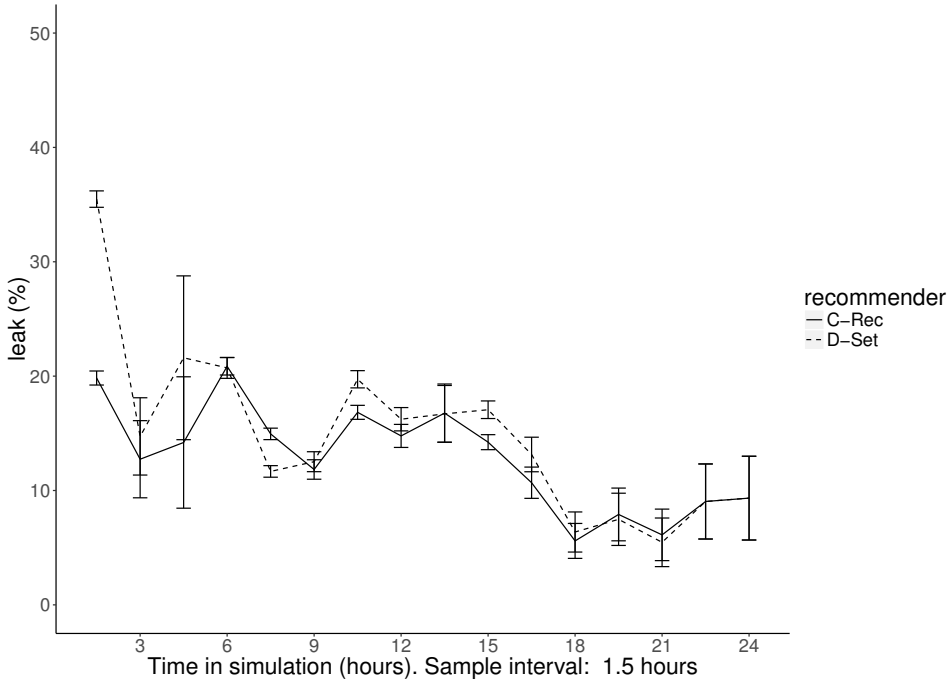


Figure 6.11: *leak* of *C-Rec* and *D-Set* with the change of simulation time. After 4.5 hours of simulation time, the *leak* of the two schemes are close.

6.5 Attack effectiveness

To evaluate the effectiveness of the sampling attack, we deploy one attacker node that has the ability to generate unlimited number of *id*s for its shill profiles. Every time the attacker node receives a profile from a user, it makes a shill profile. Compared with the attack size (15%) commonly used in other types of shilling attacks [23, 92], the size of sampling attack in our simulation is 100%, which is stronger.

In each round of simulation, the attacker first randomly chooses one target location category and one attack intent. Thus, the set of target contexts, i.e., C^{target}, are all the contexts whose location category is the target location category. To examine the effectiveness of the attack, each real user node keeps two decentralised local recommenders and only one of them is affected by the attacker's shill profiles. Every time one real user node requests a recommendation, we compare the outputs from these two recommenders. If the outputs are different, which means that the attacker's shill profiles have changed the original recommendation, we record this recommendation as a changed recommendation. Given the target contexts set C^{target} and the attack intent int, we use $ChangeRec(C^{target}, int)$ to represent all the changed recommendations due to the attacker's input.

For the attacker, once its C^{target} and int are decided, the recommendations which it aims to change are those recommendations whose contexts are in C^{target} and whose original recommendations are different from int. For instance, without the existence of the attacker, if a recommendation result is "not share", then it is a target recommendation after a "push" attacker being introduced. Equally, all the "share" recommendations are target recommendations for a "nuke" attacker. Given C^{target} and int, we represent the set of target recommendations by $TargetRec(C^{target}, int)$. Thus, for one round of simulation, the attack success ratio of the attacker is:

$$Suc(C_{target}, int) = \frac{|TargetRec(C_{target}, int) \cap ChangedRec(C_{target}, int)|}{|TargetRec(C_{target}, int)|}$$

We first examine the attack effectiveness without any mitigation. As shown in Figure 6.12, the average attack success ratio across 100 rounds of simulation is 57%. These successful attacks are conducted among the target recommendations whose contexts are in the set C^{target} and

whose original recommendation results are different from *int*. One attacker node, by simply generating 40 shill profiles from real user samples, can bias more than half of the location-privacy recommendations in the target contexts. This result shows that our decentralised location-privacy recommender system is vulnerable to the sampling attack.

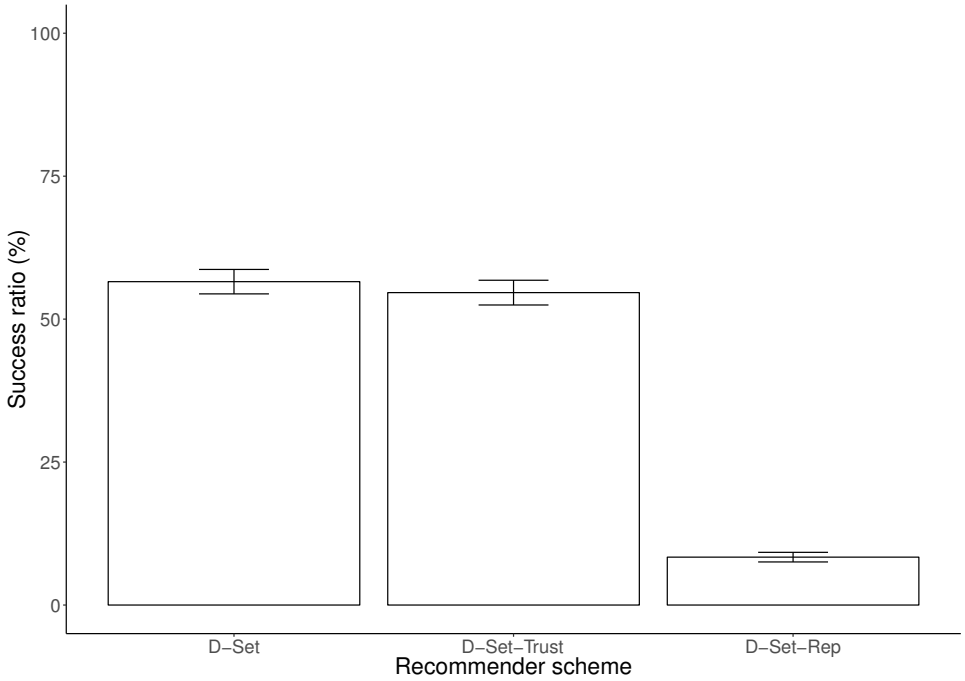


Figure 6.12: The percentage of successful attacks using different recommenders. The average attack success ratios of *D-Set* is 57%. The location-level trust's effect on alleviating the sampling attack is minimal. The attack success ratio of *D-Set-Trust* (55%) is 2% lower compared with *D-Set* ($p < 0.01$, t-test). With the reputation scheme *D-Set-Rep*, the average attack success ratio drops to 8% ($p < 0.01$, t-test).

6.6 Mitigation effectiveness

We evaluate the mitigation effectiveness by comparing the attack success ratio with and without our reputation scheme. For the result with our reputation scheme, in our analysis, we refer to it as *D-Set-Rep*.

Since the sampling attack is difficult to be detected by similarity-based methods, we compare our scheme with an existing trust-based model for recommender systems [99]. In this work, two types of trust are defined, i.e., profile-level trust and item-level trust. A profile-level trust

generally describes how accurate one profile is in terms of contributing to recommendations. An item-level trust, more specifically, describes how accurate one profile is in terms of contributing to the recommendations for a type of items. Since shill profiles in the sampling attack are highly similar with real profiles, even if they lose their profile-level trust in target recommendations, they can still gain their profile-level trust from other recommendations. Thus, we adapt the item-level trust as the location-level trust in our experiment. By this means, we can ensure that the shill profiles' location-level trust values in their target location categories are lower than those of real profiles, since they always try to make incorrect recommendations in the target location categories. We refer to the scheme of using the location-level trust model as *D-Set-Trust*. For each profile p, its location-level trust value in terms of recommending location-privacy settings in location category l^* is:

$$Trust^L(p,l^*) = \frac{|\{(r,l) \in CorrectSet(p) : l = l^*\}|}{|\{(r,l) \in RecSet(p) : l = l^*\}|}$$

$RecSet(p)$ is the set of all the recommendations that p has contributed to. $CorrectSet(p)$ is the set of the correct recommendations that p has contributed to. Thus, $Trust^L(p,l)$ is the percentage of the correct location-privacy recommendations that p has made in the location category l.

To apply the location-level trust model to recommenndations, we combine profiles' similarities with their trust values as their trust-based weightings. When u_i requests a recommendation in location category l, for each profile p_k in u_i's neighbourhood, we use the harmonic mean of its trust and similarity as its trust-based weighting [99]:

$$w(u_i,p_k,l) = \frac{2 \times w_{i,k} \times Trust^L(p_k,l)}{w_{i,k} + Trust^L(p_k,l)}$$

We use this weighting instead of $w_{i,k}$ when making recommendations. As this weighting is the harmonic mean of two values, it will be high only when both trust and similarity values are high. It showed better performance compared with other methods in previous studies [99].

Every profile needs an initial trust value for bootstrap, because it will not have a trust value until it takes part in recommendations. If p_k contributes to a recommendation for u_i in location

category l for the first time, its initial trust value $Trust^L(p_k, l)_0$ is $w_{i,k}$. By this means, p_k's initial trust-based weighting is its Cosine similarity.

We evaluate the mitigation effectiveness of our reputation scheme (*D-Set-Rep*) and compare it with the location-level trust model (*D-Set-Trust*). As shown in Figure 6.12, after deploying the reputation scheme, the attack success ratio in *D-Set-Rep* drops from 57% to 8%. The mitigation effectiveness of the location-level trust model, however, is only 2% (from 57% to 55%). As the location-level trust model is a posterior model, before the trust values of attackers accumulate, there has to be enough recommendations made by them. Thus, until the trust values of attackers drop from a series of incorrect recommendations, they are not significantly lower than those of real users. Therefore, they can still successfully conduct the sampling attack during this period of time.

To analyse the sensitivity of the threshold, we change the threshold factor β from 0 to 1 with 0.05 as the increment and use $\beta * \overline{rep}$ as the threshold to filter out profiles with low reputation. As shown in Figure 6.13, between $0.2\overline{rep}$ and $\overline{rep}$, the reputation scheme can effectively mitigate the sampling attack. When the $\beta < 0.2$, the attack success ratio increases significantly. As the reputations scheme is based on the nodes' encounter frequency, which is based on the nodes' mobility patterns, this result only represents our simulated scenario. In the real world, people's mobility patterns may be more complex and the density of population may be different, thus the sensitivity of the $\overline{rep}$ threshold may be different.

The encounter frequency of nodes in opportunistic networks, as our results suggest, can be used as a proxy of profile reputations to discriminate shill profiles from real profiles. Such reputation scheme can effectively mitigate the effectiveness of the sampling attack against decentralised recommender systems. The design of the encounter-frequency-based reputation scheme is independent of the preferences in profiles. Thus, it is difficult for attackers to elaborate their shill profiles to bypass the reputation filter. Additionally, unlike the posterior trust model, our reputation scheme does not need recommendation results to update their reputation values, which makes our scheme quicker to work.

A potential side effect of our reputation scheme is that it may filter out real profiles that have low encounter frequency and consequently influence the recommendation performance. Thus

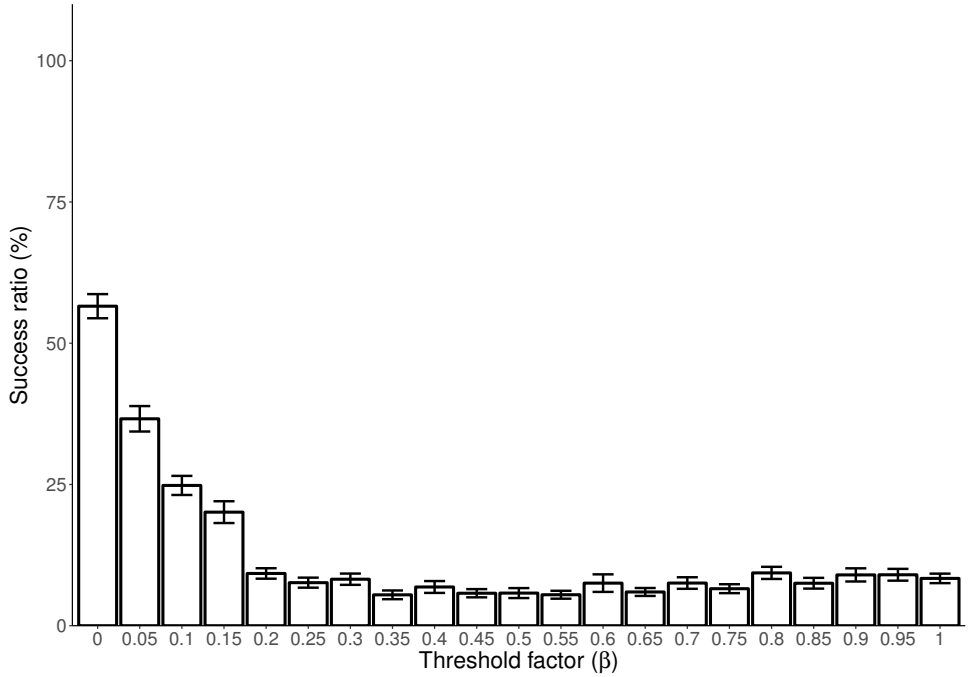


Figure 6.13: Sensitivity analysis of reputation threshold. The x-axis is a factor β and we use $\beta * \overline{rep}$ as the threshold to filter out low-reputation profiles. The y-axis is the success ratio of the sampling attack. When $\beta < 0.2$, the success ratio increases significantly.

we compare the *accuracy* and the *leak* of *D-Set* and *D-Set-Rep* to find out how much this influence can be. As shown in Figure 6.14 and Figure 6.15, the reputation scheme decreases the average *accuracy* by 2% and increases the average *leak* by 1%. Compared with the overall performance and also considering its significant mitigation effectiveness on the attack success ratio, its influence on recommendation performance is minimal.

We have demonstrated that, with the protection of our reputation scheme, the attack effectiveness of a sampling attacker with a single device can be significantly mitigated. To increase the reputation values of its shill profiles, the attacker has to encounter more nodes, thereby increasing its encounter frequency with others. One way to do that is to deploy more attacker nodes. Thus, we enable the attacker to do this to examine how many nodes the attacker has to deploy in *D-Set-Rep* to achieve the same attack success ratio that it can easily achieve in *D-Set*.

We let the attacker in our simulation deploy multiple nodes in *D-Set-Rep*. All of these nodes are controlled by the attacker and they all have the same C^{target} and *int* in each round of simulation. The shill profiles generated from the same real profile by these nodes have the same profile *id*.

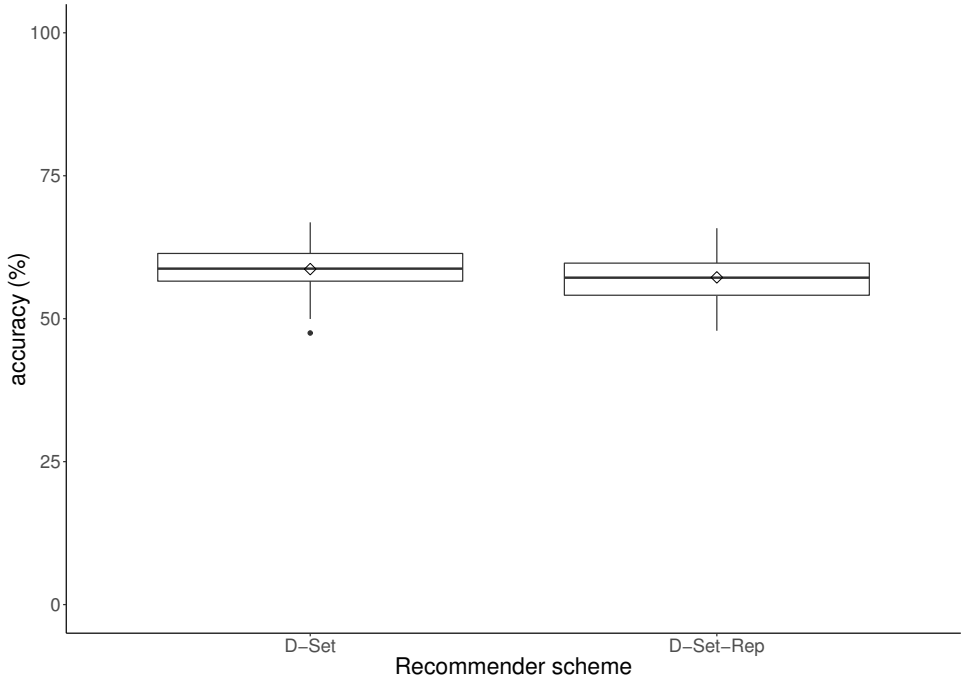


Figure 6.14: Overall *accuracy* of *D-Set* and *D-Set-Rep* in 100 rounds of simulation. The average *accuracy* of *D-Set-Rep*, 57%, is 2% lower than the average *accuracy* of *D-Set*, 59% ($p < 0.01$, t-test).

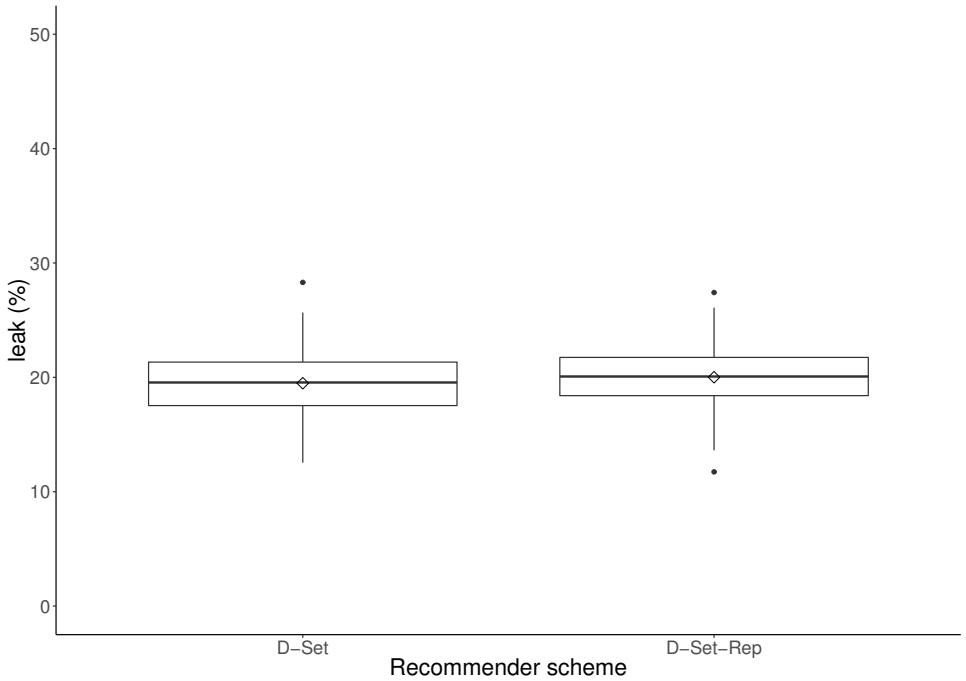


Figure 6.15: Overall *leak* of *D-Set* and *D-Set-Rep* in 100 rounds of simulation. The average *leak* of *D-Set-Rep*, 20%, is 1% higher than the average *leak* of *D-Set*, 19% ($p < 0.01$, t-test),

Thus, when a real user encounters any of these attack nodes, the reputation of the shill profiles on it can increase.

As shown in Figure 6.16, the attack success ratio in *D-Set-Rep* goes up as we deploy more attacker nodes. To achieve the same attack success ratio achieved in *D-Set*, i.e., 57%, the attacker has to deploy at least 30 nodes. This is more expensive than simply generating shill profiles. In addition, to make sure the reputation values of the shill profiles can increase on many real users' devices, these attacker nodes have to be carried by different people who have diverse mobility patterns in order to encounter as many real users as possible.

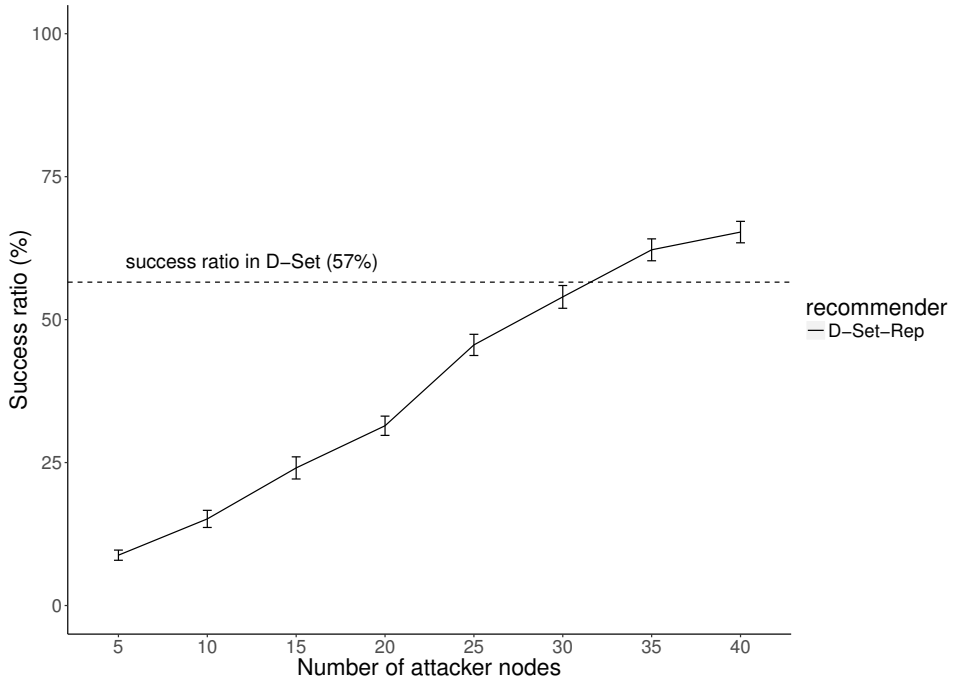


Figure 6.16: The change of success ratio when the attacker deploys multiple devices in *D-Set-Rep*. The dashed line is the success ratio achieved by deploying only one device in *D-Set*, i.e., without the reputation scheme. Our reputation scheme significantly increases the expense (the deployment of at least 30 devices) for an attacker to achieve the same attack success ratio.

6.7 Summary

In this chapter, we have demonstrated the following:

- Adding noise in users' data can protect their data privacy without significantly influencing the performance of centralised location-privacy recommender systems.

- Location-privacy recommender systems can be implemented in a decentralised fashion using opportunistic networks.
- The *accuracy* and the *leak* of decentralised location-privacy recommender systems are close to those of centralised ones, after collecting adequate data.
- The decentralised structure is vulnerable to the sampling attack.
- Encounter frequency of nodes in opportunistic networks can be used as a proxy to build a reputation scheme that can significantly mitigate the effectiveness of the sampling attack.
- The encounter-frequency-based reputation scheme has minimal influence on the performance of recommenders and can increase attackers' expense of conducting successful attack.

We evaluated the performance of the decentralised recommender system through simulation. As the nodes' movement in our simulation was restricted by the road layout of the town of St Andrews, the results from the simulation can only represent scenarios of a small town. Meanwhile, as we only considered walking mobility patterns, our results may be different from the results produced by simulation of nodes driving or taking public transport in cities.

Chapter 7

Conclusions

In LSS, effective mechanisms to control users' location disclosure is important, as inappropriate location disclosure leads to privacy risks. In this thesis, we proposed to use user-based CF recommender systems, which have been widely used in other areas to solve information overload problems, to help users configure their location-privacy settings. We have demonstrated that user-based CF recommender systems can provide accurate, acceptable, and robust location-privacy recommendations, by answering the following research questions:

- **Q1** Can recommender systems provide accurate location-privacy recommendations?
- **Q2** What factor affect people's acceptance of location-privacy recommender systems?
- **Q3** How can we modify the design of location-privacy recommender systems to make it more acceptable?

For Q1, we have been able to conclude that user-based CF recommender systems can provide accurate location-privacy recommendations that are comparable with the recommendations made by model-based classifiers. Meanwhile, user-based CF recommenders are more accurate during cold-start periods. For Q2, we have found negative effects from users' privacy concerns about providing their data to centralised recommender servers on their acceptance of our recommender system. We have also found the effects from the contexts and the openness of recommendations on acceptance. Our hypothesis that users' acceptance is affected by crowdsourcing sources could not be supported by our results. For Q3, we have been able to conclude that it is feasible to

cooperate our centralised location-privacy recommender system with data obfuscation without significantly decreasing its performance. It is also feasible to decentralise our recommender system and make it accurate and robust.

In this chapter, we summarise our contributions, analyse the limitations of our work, and discuss the possible directions for future work.

7.1 Contributions

In Chapter 4, we evaluated the performance of user-based CF location-privacy recommender systems. The contributions of this chapter included:

- We implemented a location-privacy recommender using user-based CF and we demonstrated that the overall performance of the recommender is close to the best performance of the model-based classifiers, which are more computationally expensive to use in the real world.

- We simulated the scenario of cold-start periods and demonstrated that when individual users have insufficient data, the performance of our recommender is better than that of the model-based classifiers.

In Chapter 5, we conducted online user studies to investigate people's acceptance of location-privacy recommenders and the factors that can affect their acceptance. The contributions of this chapter were as the follows:

- Through CFA and SEM, we demonstrated that people have privacy concerns about providing their data to centralised recommender servers. Such concerns have significant negative effects on their perceived recommendation quality, satisfaction about their choices, and acceptance of recommended location-privacy settings.

- Objective factors, including the contexts and the openness of recommended location-privacy settings, affect people's acceptance. Recommendations with the highest openness or the lowest openness are less likely to be accepted.

- By analysing the feedback from our participants, we speculated that the regularity of contexts

and the potential risks caused by failed recommendations may affect people's acceptance of recommended location-privacy settings.

In Chapter 6, we proposed two schemes that could alleviate people's privacy concerns about our system. We evaluated the performance of both of them, which are data obfuscation and decentralisation. For the decentralisation, we also evaluated its robustness against the sampling attack and proposed a encounter-frequency-based reputation scheme to prevent the attack. The contributions of this chapter were:

- We implemented the recommender in a privacy-aware fashion by using data obfuscation to protect users' data privacy. We showed that the loss of performance from the added noise is minimal.

- We compared the performance of the decentralised location-privacy recommender system with that of the centralised location-privacy recommender system. We demonstrated that using opportunistic networks, decentralised location-privacy recommender systems can perform as well as the centralised one does.

- We introduced a sampling attacker device in our decentralised system to bias recommendations. By simply generating shill profiles from received real profiles, more than half of target recommendations were changed by only one attacker device. Our decentralised system is vulnerable to the sampling attack.

- We used the encounter frequency of devices in opportunistic networks as a proxy of reputation. Such reputation scheme can significantly mitigate the effectiveness of the sampling attack. Compared with traditional similarity-based detection, our reputation scheme increases the expense of attackers to conduct successful shilling attack.

7.2 Discussion

In this thesis, we have proposed and evaluated location-privacy recommender systems that can automatically configure people's location-privacy settings. Inspired by the widely successful applications of user-based CF recommenders in the real world and the existing research that

demonstrates the similarity in people's location-privacy preferences, we chose user-based CF to realise our recommender. We have conducted a series of experiments to evaluate its performance and users' acceptance.

In Chapter 4, we conducted a number of evaluations on both the overall performance and the performance during cold-start periods. The data for these offline evaluations were collected previously, from the real world. Compared with data gathered from online crowdsourcing platforms, such as Amazon Mechanical Turk, the sample size of the data we used is smaller and the diversity may be lower. However, the data we used were collected *in situ* when people were actually using LSS applications, which have better quality.

In the user study that we conducted in Chapter 5, we evaluated our recommenders from users' perspective. The samples in our experiments are mainly from university students from 18 to 24 years old and the university is located in a small town. Thus, our results only represent this special group. We have controlled some objective factors of the recommenders unchanged, including the recommendation accuracy. We used a random recommendation generator for all the three recommenders in our study to make sure that the recommendations from different recommenders have the same quality. As a consequence, the overall accuracy of these recommendations is inevitably affected, compared with ideal recommendations. We believe that such side effect affects all participants, which means that it would not contribute specifically to any of the detected effects in our results. Another limitation is that we only used the participants' own location check-in histories to generate recommendations. This was to make sure that they were familiar with the contexts of recommendations, as they had been there. We believe that the choices that our participants made under such circumstances are more accurate than those made by them if we had asked them to hypothetically consider some places they had never been to. This means, however, that we were not able to evaluate their acceptance of the recommendations when they entered some new contexts.

In Chapter 6, we demonstrated that decentralised location-privacy recommender systems can perform as well as centralised systems do. One of the premises behind this result is that people only request recommendations when they arrive at their destinations and want to check in, which makes the recommendation requests sparse. Thus, there are enough time for devices to exchange

data with each other. In other recommendation scenarios, such as movie or music, it is not uncommon that people request "Top-N" recommendations very soon once they begin to use a recommender system. Thus, the decentralised structure may not be suitable to provide accurate recommendations in such scenarios.

7.3 Future work

Our study has demonstrated the feasibility of location-privacy recommender systems and has helped us understand people's acceptance of such systems. Based on our results, we suggest some potentially interesting directions for future research.

7.3.1 Confidence, explanation, and obfuscation

In this thesis, we have evaluated the objective performance, i.e., *accuracy* and *leak*, of our location-privacy recommenders, and people's acceptance of them. Several other factors are also worth to be investigated in the future.

The first one is the confidence of recommendations. As described in Chapter 4, the recommendations in our system were made based on the weights of two groups of neighbours who had different decisions (i.e., "share" or "not share"). If one group's weight is much more than that of another group, e.g., 90% vs. 10%, we can say that the recommendation in this case has a relatively high confidence value. If the two weights, however, are close, e.g., 55% vs. 45%, does it mean that the recommendations are more likely to fail? One possible research question to be examined is whether the confidence of recommendations has relations with the performance of recommenders. We also would like to examine if there are relations between the confidence of recommendations and people's acceptance of them.

The second one is the explanation of recommendations. For the same location-privacy decision, different people may have different reasons to make it. For example, one may decide to share a location for social benefits while another one may make the same decision for discounts. If we can capture these reasons, we may use them as explanations for recommendations. Thus, additional dimensions need to be introduced in contexts to represent different explanations,

which may lead to better recommendation accuracy.

The third one is combining obfuscation with recommended location-privacy settings. As we discussed in Chapter 2, data obfuscation such as anonymity has been used to protect location privacy. Thus, we can introduce data obfuscation in location-privacy recommendations. For example, people may not accept a "share" or "not share" recommendation at a specific location and a specific time point. But adding obfuscation into the temporal dimensions (e.g., marking location check-ins with time ranges rather than time points) or spatial dimensions (e.g., sharing areas rather than specific locations) of recommendations may be acceptable.

The possible outcome from the investigation about recommendation confidence, explanation, and obfuscation may also help with people's acceptance of recommended location-privacy settings. In the user study that we conducted in Chapter 5, people's acceptance varied based on the openness and the contexts of recommendations. For a recommendation, if we provide its confidence and explanation to users and make obfuscation in different dimensions available, may this improve the users' acceptance in the cases that had low acceptance in our study? To examine question, we need to find an effective way to let people express the reasons behind their decisions. We also need to select the most relevant explanations to be shown to users.

7.3.2 Decentralised recommender systems in other areas

Decentralisation has extended the design space of recommender systems, from completely centralised infrastructures to structures based on existing mobile opportunistic networks. It not only provides a way to balance the trade-off between people's data privacy and the performance of systems, but also can potentially reduce the expense of service deployment.

The results of our simulation in Chapter 6 have shown that decentralised location-privacy recommender systems can perform well once have received adequate data. Although, as we discussed in the previous section, the decentralised structure may not be suitable in some specific application scenarios, it may be able to provide accurate recommendations in some applications wherein the requests of recommendations are sparse. For example, in location recommender systems, people normally request recommendations when they want to move to the next place, or during a certain period of time (e.g., recommending nearby restaurants at lunch time). Thus, there

may be enough time for people's devices to receive data before recommendations are requested. We speculate that decentralised recommender systems may be helpful in these scenarios.

Our results have only shown the feasibility of decentralised location-privacy recommender systems. The major motivation of Chapter 6 is to alleviate people's concerns about providing their data to our system. Therefore, it is also needed to investigate the actual mitigation effectiveness of the decentralised structure on people's concerns. In addition, it is worth to examine the reputation scheme against other types of shilling attacks.

7.3.3 Privacy recommenders in other areas

In this thesis, we have applied user-based CF recommenders to helping people make decisions specifically for location privacy. We only considered people explicitly sharing their locations, i.e. through location check-ins. People's location information, however, can be inferred from many other types of medium. For example, from analysing photos and the user tags of photos, the locations when the photos were taken can be inferred [44]. People's online social media contents such as message context, social networks, and user profiles can also be used to infer their locations [7]. Therefore, location-privacy recommenders can also be used to control the disclosure of the above mentioned types of information.

Our work sheds light on the probability of using recommender systems to support complex decisions that have similarity. Privacy decisions are a subset of people's decisions. One explanation of privacy decisions is the *privacy calculus* theory, which says that people's privacy decisions are the result of the trade-off between the risks and the benefits of the decision. If people's privacy decisions in other areas also fit such theory, it is worth to investigate the possibility of a general privacy recommender system that can be used in different areas.

In other areas that have complex contexts, such as online social networks, people also face the difficulty of protecting their privacy effectively. If their privacy preferences also have similarity in these areas, it may be possible to use recommender systems to help them configure their privacy settings as well. Then one of the key research questions is that, in different application scenarios, what features we should consider to compose contexts. Existing research has shown that the **recipient** with whom people share their data and the **reason** why the recipient wants to access

also affect people's privacy decisions. Introducing these features into privacy recommenders may bring better recommendations.

Apart from considering more features, finer-grained privacy preferences may also lead to better recommendations. Due to the limitation of the used data set, we only considered binary privacy decisions (i.e. "share" and "not share") in this thesis. In the real world, people's privacy preferences may not be as simple as this and they may have different thresholds to make decisions. Thus, to test these hypotheses, we need to collect people's privacy preferences with finer granularity. In the process of publishing and using such data, people's data privacy (e.g. differential privacy [38]) should be guaranteed as well.

7.3.4 Deployment and scalability

As the usability issues in privacy protection exist not only in location-sharing services, but also in other applications such as online social media, a reasonable way to deploy privacy recommender systems in the real world is to realise it as a third party open-source framework that can be used by many other applications. By this means, although different applications use heterogeneous data, we can still use the contexts of the information disclosure in these applications to form "user-decision" matrices and build recommenders upon them.

When deploying recommender systems in the real world, one of the major issues is the scalability of the systems. For an online recommender system with n users and a neighbourhood size of k, the major computational expense is to find the k nearest neighbours for each of these n users. Thus, its time complexity is $O(nk)$. Constant time CF algorithms such as Eigentaste [45] has been proposed to reduce the time complexity to $O(k)$.

Appendix A

Glossary

- **Ubiquitous computing**: An environment wherein computing resource is available anytime and everywhere.
- **Global positioning system (GPS)**: A system that provides geolocation information globally to its receivers.
- **Location-based services (LBS)**: Services that use people's location information as a feature.
- **Location-sharing services (LSS)**: Services that enable people to share their location information with each other.
- **Role-based access control (RBAC)**: Mechanisms that control subjects' access to objects, based on the roles of subjects and the privileges of the roles.
- **Online social networks (OSN)**: Online platforms that allow people to build their social networks with others.
- **Points of interest (POI)**: Locations that may be of some people's interests.
- **Web 2.0**: Websites that allow and encourage users to generate their own content.
- **Location-based social network (LBSN)**: Platforms that use people's location information as a feature to customise their social networks.

- **Anonymity**: The ability to be unidentifiable.
- **Collaborative filtering (CF)**: A technique that predicts a user's interests by analysing the preferences from many other users.
- **Privacy-enhancing technologies (PET)**: Techniques that help users to protect the privacy of their personal information.
- **Shilling attacks**: Attacks conducted by malicious users using fake profiles to inject ratings to bias the recommendation results of recommender systems.
- **Confirmatory factor analysis (CFA)**: A form of factor analysis used to examine whether the measures of a factor are valid.
- **Structural equation modeling (SEM)**: A statistical analysis technique used to analyse multiple structural relationships.
- **Opportunistic networks**: A type of network of mobile devices connected wirelessly without an infrastructure.

Appendix B

Ethics approval

The ethics approval letter for the user study discussed in Chapter 5 is included on the next page.

University of St Andrews
Scotland's first university – 1413

University Teaching and Research Ethics Committee Sub-committee

24th June 2015
Yuchen Zhao
School of Computer Science

Ethics Reference No: *Please quote this ref on all correspondence*	CS11570
Project Title:	A User Study About Location-sharing Preference Recommendations
Researchers Name(s):	Yuchen Zhao
Supervisor(s):	Dr Tristan Henderson, Dr Juan Ye

Thank you for submitting your application which was considered at the Computer Science School Ethics Committee meeting on the 19th June 2015. The following documents were reviewed:

1. Ethical Application Form — 27th May 2015
2. Participant Information Sheet — 27th May 2015
3. Consent Form — 27th May 2015
4. Debriefing Form — 27th May 2015
5. Advertisement — 27th May 2015
6. User Instructions — 27th May 2015

The University Teaching and Research Ethics Committee (UTREC) approve this study from an ethical point of view. Please note that where approval is given by a School Ethics Committee that committee is part of UTREC and is delegated to act for UTREC.

Approval is given for three years. Projects, which have not commenced within two years of original approval, must be re-submitted to your School Ethics Committee.

You must inform your School Ethics Committee when the research has been completed. If you are unable to complete your research within the 3 three year validation period, you will be required to write to your School Ethics Committee and to UTREC (where approval was given by UTREC) to request an extension or you will need to re-apply.

Any serious adverse events or significant change which occurs in connection with this study and/or which may alter its ethical consideration must be reported immediately to the School Ethics Committee, and an Ethical Amendment Form submitted where appropriate.

Approval is given on the understanding that the 'Guidelines for Ethical Research Practice' https://www.st-andrews.ac.uk/utrec/guidelines/ are adhered to.

Yours sincerely

P. J. Robertson

Convenor of the School Ethics Committee

Ccs Supervisor
School Ethics Committee

Appendix C

Questionnaires

For the user study discussed in Chapter 5, we use the following questionnaires to measure the *trust*, *quality*, *satisfaction*, and *concerns* of our participants. These questionnaires are a modified version of the questionnaires from the user-centric framework [71], according to our need. The answer for each question is one of "Strongly disagree", "Disagree", "Neutral", "Agree", and "Strongly agree".

- *trust*
    1. Technology never works.
    2. I'm less confident when I use technology.
    3. The usefulness of technology is highly overrated.
    4. Technology may cause harm to people.
    5. I prefer to do things by hand.
    6. I have no problems trust my life to technology.
    7. I always double-check computer results.
- *quality*
    1. I like the location-sharing choices that were made by the system.

2. The recommendations fitted my location-privacy preferences.
3. The recommended location-sharing choices were well-chosen.
4. The recommended location-sharing choices were relevant.
5. The system recommended too many bad location-sharing choices.
6. I didn't like any of the recommended location-sharing choices.
7. The recommendations I accepted were "the best among the worst".

- *satisfaction*
  1. I like the recommendations that I've accepted.
  2. I would like to use my chosen location-sharing choices to protect my location privacy.
  3. The location-sharing choices I chose are incapable to protect my location privacy.
  4. The chosen location-sharing choices fit my location-privacy preferences.
  5. I can configure better-location sharing preferences than the ones that I accepted.
  6. Some of my chosen location-sharing choices could become part of my default location-privacy settings.
  7. I would recommend some of the chosen location-sharing choices to others/friends.
- *concerns*
  1. I'm afraid that the system discloses private information about me.
  2. The system invades my privacy.
  3. I feel confident that the system respects my privacy.
  4. I'm uncomfortable providing private data to the system.
  5. I think the system respects the confidentiality of my data.

References

[1] About Foursquare. https://foursquare.com/about. Accessed: 18-12-2016 16:02:13.

[2] 'El Chapo': Drug Boss's Son Accidentally Reveals Fugitive's Location on Twitter. http://www.independent.co.uk/news/world/americas/el-chapo-drug-bosss-son-accidently-reveals-fugatives-location-on-twitter-10488067.html. Accessed: 03-01-2017 14:50:31.

[3] Global smartphone shipments forecast from 2010 to 2020 (in million units). https://www.statista.com/statistics/271491/worldwide-shipments-of-smartphones-since-2009/. Accessed: 15-12-2016 18:49:20.

[4] John McAfee's Secret Location May Have Been Revealed by Vice Journalist. https://www.theguardian.com/world/2012/dec/03/john-mcafee-location-revealed-vice. Accessed: 03-01-2017 14:55:07.

[5] New Zealander Thought to be Fighting in Syria Accidentally Tweets Locations. https://www.theguardian.com/world/2015/jan/01/new-zealander-syria-isis-accidentally-tweets-locations. Accessed:03-01-2017 14:56:40.

[6] Web Photos That Reveals Secrets, Like Where You Live. http://www.nytimes.com/2010/08/12/technology/personaltech/12basics.html. Accessed: 03-01-2017 14:46:13.

[7] O. Ajao, J. Hong, and W. Liu. A Survey of Location Inference Techniques on Twitter. *Journal of Information Science*, 41(6):855–864, Dec. 2015. https://doi.org/10.1177/0165551515602847.

[8] H. Almuhimedi, F. Schaub, N. Sadeh, I. Adjerid, A. Acquisti, J. Gluck, L. F. Cranor, and Y. Agarwal. Your Location has been Shared 5,398 Times!: A Field Study on Mobile App Privacy Nudging. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 787–796, Seoul, Republic of Korea, Apr. 2015. https://doi.org/10.1145/2702123.2702210.

[9] D. Anthony, T. Henderson, and D. Kotz. Privacy in Location-Aware Computing Environments. *IEEE Pervasive Computing*, 6(4):64–72, Oct. 2007. https://doi.org/10.1109/MPRV.2007.83.

[10] L. Barkuus and A. Dey. Location-Based Services for Mobile Telephony : a Study of Users' Privacy Concerns. In *Proceedings of the 9th IFIP TC13 International Conference on Human-Computer Interaction*, pages 709–712, Zurich, Switzerland, July 2003.

[11] P. Bellavista, A. Küpper, and S. Helal. Location-Based Services: Back to the Future. *IEEE Pervasive Computing*, 7(2):85–89, Apr. 2008. https://doi.org/10.1109/MPRV.2008.34.

[12] M. Benisch, P. G. Kelley, N. Sadeh, and L. F. Cranor. Capturing Location-Privacy Preferences: Quantifying Accuracy and User-Burden Tradeoffs. *Personal and Ubiquitous Computing*, 15(7):679–694, Dec. 2011. https://doi.org/10.1007/s00779-010-0346-0.

[13] A. R. Beresford and F. Stajano. Location Privacy in Pervasive Computing. *IEEE Pervasive Computing*, 2(1):46–55, Jan. 2003. https://doi.org/10.1109/MPRV.2003.1186725.

[14] A. R. Beresford and F. Stajano. Mix Zones: User Privacy in Location-Aware Services. In *Proceedings of the 2nd IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pages 127–131, Orlando, FL, USA, Mar. 2004. https://doi.org/10.1109/PERCOMW.2004.1276918.

[15] R. Bhaumik, C. Williams, B. Mobasher, and R. Burke. Securing Collaborative Filtering Against Malicious Attacks through Anomaly Detection. In *Proceedings of the 4th Workshop on Intelligent Techniques for Web Personalization*, Boston, MA, USA, July 2006.

[16] G. Bigwood, F. Ben Abdesslem, and T. Henderson. Predicting Location-Sharing Privacy Preferences in Social Network Applications. In *Proceedings of the 1st Workshop on Recent Advances in Behavior Prediction and Pro-active Pervasive Computing*, Newcastle, UK, June 2012. Online at http://www.ibr.cs.tu-bs.de/dus/Awarecast/awarecast2012_submission_1.pdf.

[17] I. Bilogrevic, K. Huguenin, B. Agir, M. Jadliwala, and J.-P. Hubaux. Adaptive Information-Sharing for Privacy-Aware Mobile Social Networks. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 657–666, Zurich, Switzerland, Sept. 2013. https://doi.org/10.1145/2493432.2493510.

[18] M. Böhmer, G. Bauer, and A. Krüger. Exploring the Design Space of Context-Aware Recommender Systems that Suggest Mobile Applications. In *Proceedings of the 2nd Workshop on Context-Aware Recommender Systems*, Barcelona, Spain, Sept. 2010.

[19] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer. Falling Asleep with Angry Birds, Facebook and Kindle: A Large Scale Study on Mobile Application Usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 47–56, Stockholm, Sweden, Sept. 2011. https://doi.org/10.1145/2037373.2037383.

[20] K. A. Bollen and J. Pearl. Eight Myths About Causality and Structural Equation Models. In *Handbook of Causal Analysis for Social Research*, pages 301–328. 2013. https://doi.org/10.1007/978-94-007-6094-3_15.

[21] A. Brimicombe. GIS – Where are the frontiers now? In *Proceedings of GIS 2002*, pages 33–45, Manama, Bahrain, Mar. 2002.

[22] R. Burke. Hybrid Web Recommender Systems. In *The Adaptive Web*, pages 377–408. 2007. https://doi.org/10.1007/978-3-540-72079-9_12.

[23] R. Burke, B. Mobasher, and R. Bhaumik. Limited Knowledge Shilling Attacks in Collaborative Filtering Systems. In *Proceedings of 3rd Workshop on Intelligent Techniques for Web Personalization*, pages 17–24, Edinburgh, UK, Aug. 2005.

[24] R. Burke, B. Mobasher, C. Williams, and R. Bhaumik. Classification Features for Attack Detection in Collaborative Recommender Systems. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 542–547, Philadelphia, PA, USA, Aug. 2006. https://doi.org/10.1145/1150402.1150465.

[25] J. Canny. Collaborative Filtering with Privacy. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 45–57, Oakland, CA, USA, May 2002. https://doi.org/10.1109/SECPRI.2002.1004361.

[26] J. Chang and E. Sun. Location 3 : How Users Share and Respond to Location-Based Data on Social Networking Sites. In *Proceedings of the 5th AAAI Conference on Weblogs and Social Media*, pages 74–80, Barcelona, Spain, July 2011.

[27] P.-A. Chirita, W. Nejdl, and C. Zamfir. Preventing Shilling Attacks in Online Recommender Systems. In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, pages 67–74, Bremen, Germany, Nov. 2005. https://doi.org/10.1145/1097047.1097061.

[28] E. Cho, S. A. Myers, and J. Leskovec. Friendship and Mobility, User Movement in Location-Based Social Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1082–1090, San Diego, CA, USA, Aug. 2011. https://doi.org/10.1145/2020408.2020579.

[29] C.-Y. Chow, M. F. Mokbel, and X. Liu. A Peer-to-Peer Spatial Cloaking Algorithm for Anonymous Location-Based Service. In *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems*, pages 171–178, Arlington, VA, USA, Nov. 2006. https://doi.org/10.1145/1183471.1183500.

[30] H. Cramer, M. Rost, and L. E. Holmquist. Performing a Check-In: Emerging Practices, Norms and 'Conflicts' in Location-Sharing Using Foursquare. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 57–66, Stockholm, Sweden, Sept. 2011. https://doi.org/10.1145/2037373.2037384.

[31] J. Cranshaw, J. Mugan, and N. Sadeh. User-Controllable Learning of Location Privacy Policies With Gaussian Mixture Models. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pages 1146–1152, San Francisco, CA, USA, Aug. 2011.

[32] M. J. Culnan and P. K. Armstrong. Information Privacy Concerns, Procedural Fairness, and Impersonal Trust: An Empirical Investigation. *Organization Science*, 10(1):104–115, Feb. 1999. https://doi.org/10.1287/orsc.10.1.104.

[33] G. Danezis. Inferring Privacy Policies for Social Networking Services. In *Proceedings of the 2nd ACM Workshop on Security and Artificial Intelligence*, pages 5–10, Chicago, IL, USA, Nov. 2009. https://doi.org/10.1145/1654988.1654991.

[34] T. H. Dao, S. R. Jeong, and H. Ahn. A Novel Recommendation Model of Location-Based Advertising: Context-Aware Collaborative Filtering using GA approach. *Expert Systems with Applications*, 39(3):3731–3739, Feb. 2012. https://doi.org/10.1016/j.eswa.2011.09.070.

[35] M. De Domenico, A. Lima, and M. Musolesi. Interdependence and Predictability of Human Mobility and Social Interactions. *Pervasive and Mobile Computing*, 9(6):798–807, Dec. 2013. https://doi.org/10.1016/j.pmcj.2013.07.008.

[36] M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro. Semantics-Aware Content-Based Recommender Systems. In *Recommender Systems Handbook*, pages 119–159. 2015. https://doi.org/10.1007/978-1-4899-7637-6_4.

[37] C. Dong, H. Jin, and B. P. Knijnenburg. Predicting Privacy Behavior on Online Social Networks. In *Proceedings of the 9th International Conference on Web and Social Media*, pages 91–100, Oxford, UK, May 2015.

[38] C. Dwork. Differential Privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, pages 2:1–2:12, Venice, Italy, July 2006. https://doi.org/10.1007/11787006_1.

[39] M. D. Ekstrand, M. Ludwig, J. A. Konstan, and J. T. Riedl. Rethinking the Recommender Research Ecosystem: Reproducibility, Openness, and LensKit. In *Proceedings of the*

5th ACM Conference on Recommender Systems, pages 133–140, Chicago, IL, USA, Oct. 2011. https://doi.org/10.1145/2043932.2043958.

[40] M. J. Eppler and J. Mengis. The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines. *The Information Society*, 20(5):325–344, Nov. 2004. https://doi.org/10.1080/01972240490507974.

[41] L. Fang and K. LeFevre. Privacy Wizards for Social Networking Sites. In *Proceedings of the 19th International Conference on World Wide Web*, pages 351–360, Raleigh, NC, USA, Apr. 2010. https://doi.org/10.1145/1772690.1772727.

[42] H. Fu, Y. Yang, N. Shingte, J. Lindqvist, and M. Gruteser. A Field Study of Run-Time Location Access Disclosures on Android Smartphones. In *Proceedings of the NDSS Workshop on Usable Security*, San Diego, CA, USA, Feb. 2014. https://doi.org/10.14722/usec.2014.23044.

[43] S. Furnell. Managing Privacy Settings: Lots of Options, but Beyond Control? *Computer Fraud & Security*, 2015(4):8–13, Apr. 2015. https://doi.org/10.1016/S1361-3723(15)30027-0.

[44] A. Gallagher, D. Joshi, J. Yu, and J. Luo. Geo-location Inference from Image Content and User Tags. In *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 55–62, Miami, FL, USA, June 2009. https://doi.org/10.1109/CVPRW.2009.5204168.

[45] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Information Retrieval*, 4(2):133–151, 2001. https://doi.org/10.1023/A:1011419012209.

[46] M. Gruteser and D. Grunwald. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pages 31–42, San Francisco, CA, USA, May 2003. https://doi.org/10.1145/1066116.1189037.

[47] I. Gunes, C. Kaleli, A. Bilge, and H. Polat. Shilling Attacks Against Recommender Systems: A Comprehensive Survey. *Artificial Intelligence Review*, 42(4):767–799, Dec. 2014. https://doi.org/10.1007/s10462-012-9364-9.

[48] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA Data Mining Software: An Update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, Nov. 2009. https://doi.org/10.1145/1656274.1656278.

[49] M. Harbach, M. Hettig, S. Weber, and M. Smith. Using Personal Examples to Improve Risk Communication for Security & Privacy Decisions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2647–2656, Toronto, Ontario, Canada, May 2014. https://doi.org/10.1145/2556288.2556978.

[50] W. He, Y. Huang, K. Nahrstedt, and B. Wu. Message Propagation in ad-hoc-based Proximity Mobile Social Networks. In *Proceedings of the 8th IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 141–146, Mannheim, Germany, Mar. 2010. https://doi.org/10.1109/PERCOMW.2010.5470617.

[51] B. Henne, C. Kater, and M. Smith. On Usable Location Privacy for Android with Crowd-Recommendations. In *Proceedings of the 7th International Conference on Trust and Trustworthy Computing*, pages 74–82, Heraklion, Crete, June 2014. https://doi.org/10.1007/978-3-319-08593-7_5.

[52] S. R. Hiltz and M. Turoff. Structuring Computer-Mediated Communication Systems to Avoid Information Overload. *Communications of the ACM*, 28(7):680–689, July 1985. https://doi.org/10.1145/3894.3895.

[53] T. K. Ho. Random Decision Forests. In *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282, Aug. 1995. https://doi.org/10.1109/ICDAR.1995.598994.

[54] R. Hoyle, S. Patil, D. White, J. Dawson, P. Whalen, and A. Kapadia. Attire: Conveying Information Exposure through Avatar Apparel. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work Companion*, pages 19–22, San Antonio, TX, USA, Feb. 2013. https://doi.org/10.1145/2441955.2441961.

[55] L. Hu and P. M. Bentler. Cutoff Criteria for Fit Indexes in Covariance Structure Analysis: Conventional Criteria Versus New Alternatives. *Structural Equation Modeling: A Multidisciplinary Journal*, 6(1):1–55, 1999. https://doi.org/10.1080/10705519909540118.

[56] L. Humphreys. Mobile Social Networks and Social Practice: A Case Study of Dodgeball. *Journal of Computer-Mediated Communication*, 13(1):341–360, Oct. 2007. https://doi.org/10.1111/j.1083-6101.2007.00399.x.

[57] C.-C. Hung, C.-W. Chang, and W.-C. Peng. Mining Trajectory Profiles for Discovering User Communities. In *Proceedings of the 2009 International Workshop on Location Based Social Networks*, pages 1–8, Seattle, WA, USA, Nov. 2009. https://doi.org/10.1145/1629890.1629892.

[58] L. Hutton and T. Henderson. An Architecture for Ethical and Privacy-Sensitive Social Network Experiments. *ACM SIGMETRICS Performance Evaluation Review*, 40(4):90–95, Mar. 2013. https://doi.org/10.1145/2479942.2479954.

[59] L. Hutton and T. Henderson. "I Didn't Sign Up for This!" : Informed Consent in Social Network Research. In *Proceedings of the 9th International AAAI Conference on Web and Social Media*, pages 178–187, Oxford, UK, May 2015.

[60] Q. Ismail, T. Ahmed, A. Kapadia, and M. K. Reiter. Crowdsourced Exploration of Security Configurations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 467–476, Seoul, Republic of Korea, Apr. 2015. https://doi.org/10.1145/2702123.2702370.

[61] H. Jin, G. Saldamli, R. Chow, and B. P. Knijnenburg. Recommendations-Based Location Privacy Control. In *Proceedings of the 2013 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 401–404, San Diego, CA, USA, Mar. 2013. https://doi.org/10.1109/PerComW.2013.6529526.

[62] G. H. John and P. Langley. Estimating Continuous Distributions in Bayesian Classifiers. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pages 338–345, Montréal, Qué, Canada, Aug. 1995.

[63] C.-M. Karat, C. Brodie, and J. Karat. Usable Privacy and Security for Personal Information Management. *Communications of the ACM*, 49(1):56–57, Jan. 2006. https://doi.org/10.1145/1107458.1107491.

[64] P. G. Kelley, M. Benisch, L. F. Cranor, and N. Sadeh. When are Users Comfortable Sharing Locations with Advertisers? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2449–2452, Vancouver, BC, Canada, May 2011. https://doi.org/10.1145/1978942.1979299.

[65] P. G. Kelley, P. Hankes Drielsma, N. Sadeh, and L. F. Cranor. User-Controllable Learning of Security and Privacy Policies. In *Proceedings of the 1st ACM Workshop on Workshop on AISec*, pages 11–18, Alexandria, VA, USA, Oct. 2008. https://doi.org/10.1145/1456377.1456380.

[66] D. A. Kenny, B. Kaniskan, and D. B. McCoach. The Performance of RMSEA in Models With Small Degrees of Freedom. *Sociological Methods & Research*, 44(3):486–507, Aug. 2015. https://doi.org/10.1177/0049124114543236.

[67] A. Keränen, J. Ott, and T. Kärkkäinen. The ONE Simulator for DTN Protocol Evaluation. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, Rome, Italy, Mar. 2009. https://doi.org/10.4108/ICST.SIMUTOOLS2009.5674.

[68] B. P. Knijnenburg and A. Kobsa. Helping Users with Information Disclosure Decisions: Potential for Adaptation. In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, pages 407–416, Santa Monica, CA, USA, Mar. 2013. https://doi.org/10.1145/2449396.2449448.

[69] B. P. Knijnenburg, A. Kobsa, and H. Jin. Dimensionality of Information Disclosure Behavior. *International Journal of Human-Computer Studies*, 71(12):1144–1162, Dec. 2013. https://doi.org/10.1016/j.ijhcs.2013.06.003.

[70] B. P. Knijnenburg and M. C. Willemsen. Evaluating Recommender Systems with User Experiments. In *Recommender Systems Handbook*, pages 309–352. 2015. https://doi.org/10.1007/978-1-4899-7637-6_9.

[71] B. P. Knijnenburg, M. C. Willemsen, Z. Gantner, H. Soncu, and C. Newell. Explaining the User Experience of Recommender Systems. *User Modeling and User-Adapted Interaction*, 22(4-5):441–504, Oct. 2012. https://doi.org/10.1007/s11257-011-9118-4.

[72] A. Kobsa, B. P. Knijnenburg, and B. Livshits. Let's Do It at My Place Instead?: Attitudinal and Behavioral Study of Privacy in Client-Side Personalization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 81–90, Toronto, Ontario, Canada, Apr. 2014. https://doi.org/10.1145/2556288.2557102.

[73] B. Kölmel and S. Alexakis. Location Based Advertising. In *Proceedings of the 1st International Conference on Mobile Business*, pages 1–7, Athens, Greece, July 2002.

[74] Y. Koren, R. Bell, and C. Volinsky. Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8):30–37, Aug. 2009. https://doi.org/10.1109/MC.2009.263.

[75] S. Korff and R. Böhme. Too Much Choice : End-User Privacy Decisions in the Context of Choice Proliferation. pages 69–87, Menlo Park, CA, USA, July 2014.

[76] V. Kostakos, J. Venkatanathan, B. Reynolds, N. Sadeh, E. Toch, S. A. Shaikh, and S. Jones. Who's Your Best Friend?: Target Privacy Attacks in Location-Sharing Social Networks. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, pages 177–186, 2011. https://doi.org/10.1145/2030112.2030138.

[77] D. Kotz and T. Henderson. CRAWDAD: A Community Resource for Archiving Wireless Data at Dartmouth. *IEEE Pervasive Computing*, 4(4):12–14, Oct. 2005. https://doi.org/10.1109/MPRV.2005.75.

[78] J. Krumm. A Survey of Computational Location Privacy. *Personal and Ubiquitous Computing*, 13(6):391–399, Oct. 2008. https://doi.org/10.1007/s00779-008-0212-5.

[79] S. K. Lam and J. Riedl. Shilling Recommender Systems for Fun and Profit. In *Proceedings of the 13th International Conference on World Wide Web*, pages 393–402, New York, NY, USA, May 2004. https://doi.org/10.1145/988672.988726.

[80] M. Langheinrich. Privacy by Design – Principles of Privacy-Aware Ubiquitous Systems. In *Proceedings of the International Conference on Ubiquitous Computing*, pages 273–291, Atlanta, GA, USA, Sept. 2001. https://doi.org/10.1007/3-540-45427-6_23.

[81] M. Langheinrich. A Privacy Awareness System for Ubiquitous Computing Environments. In *Proceedings of the 4th International Conference on Ubiquitous Computing*, pages 237–245, Göteborg, Sweden, Sept. 2002. https://doi.org/10.1007/3-540-45809-3_19.

[82] S. Lee, K. J. Kim, and S. S. Sundar. Customization in Location-Based Advertising: Effects of Tailoring Source, Locational Congruity, and Product Involvement on Ad Attitudes. *Computers in Human Behavior*, 51(Part A):336–343, Oct. 2015. https://doi.org/10.1016/j.chb.2015.04.049.

[83] D. Lian and X. Xie. Learning Location Naming from User Check-In Histories. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 112–121, Chicago, IL, USA, Nov. 2011. https://doi.org/10.1145/2093973.2093990.

[84] J. Lindqvist, J. Cranshaw, J. Wiese, J. Hong, and J. Zimmerman. I'm the Mayor of My House: Examining Why People Use Foursquare - a Social-Driven Location Sharing Application. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2409–2418, Vancouver, BC, Canada, May 2011. https://doi.org/10.1145/1978942.1979295.

[85] Y. Liu, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Analyzing Facebook Privacy Settings: User Expectations vs. Reality. In *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, pages 61–70, Berlin, Germany, Nov. 2011. https://doi.org/10.1145/2068816.2068823.

[86] M. Madejski, M. Johnson, and S. M. Bellovin. A Study of Privacy Settings Errors in an Online Social Network. In *Proceedings of the 2012 IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 340–345, Lugano, Switzerland, Mar. 2012. https://doi.org/10.1109/PerComW.2012.6197507.

[87] P. Maes. Agents that Reduce Work and Information Overload. *Communications of the ACM*, 37(7):30–40, July 1994. https://doi.org/10.1145/176789.176792.

[88] A. McAfee and E. Brynjolfsson. Big Data: The Management Revolution. *Harvard Business Review*, 90(10):61–68, Oct. 2012.

[89] S. M. McNee, J. Riedl, and J. A. Konstan. Making Recommendations Better: An Analytic Model for Human-Recommender Interaction. In *Proceedings of the CHI '06 Extended Abstracts on Human Factors in Computing Systems*, pages 1103–1108, Montréal, Québec, Canada, Apr. 2006. https://doi.org/10.1145/1125451.1125660.

[90] J. T. Meyerowitz and R. R. Choudhury. Realtime Location Privacy via Mobility Prediction: Creating Confusion at Crossroads. In *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications*, pages 2:1–2:6, Santa Cruz, CA, USA, Feb. 2009. https://doi.org/10.1145/1514411.1514413.

[91] M. Miettinen, S. Heuser, W. Kronz, A.-R. Sadeghi, and N. Asokan. ConXsense: Automated Context Classification for Context-Aware Access Control. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, pages 293–304, Kyoto, Japan, June 2014. https://doi.org/10.1145/2590296.2590337.

[92] B. Mobasher, R. Burke, R. Bhaumik, and J. Sandvig. Attacks and Remedies in Collaborative Recommendation. *IEEE Intelligent Systems*, 22(3):56–63, May 2007. https://doi.org/10.1109/MIS.2007.45.

[93] M. Mondal, Y. Liu, B. Viswanath, G. K. P., and A. Mislove. Understanding and Specifying Social Access Control Lists. In *Proceedings of the Symposium on Usable Privacy and Security*, pages 271–283, Menlo Park, CA, USA, July 2014.

[94] J. Mugan, T. Sharma, and N. Sadeh. Understandable Learning of Privacy Preferences Through Default Personas and Suggestions. Technical Report CMU-ISR-11-112, Institute for Software Research, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, Aug. 2011. Online at http://reports-archive.adm.cs.cmu.edu/anon/isr2011/abstracts/11-112.html.

[95] K. D. Naini, I. S. Altingovde, R. Kawase, E. Herder, and C. Niederée. Analyzing and Predicting Privacy Settings in the Social Web. In *Proceedings of the 23rd International Conference on User Modeling, Adaptation and Personalization*, pages 104–117, Dublin, Ireland, June 2015. https://doi.org/10.1007/978-3-319-20267-9_9.

[96] Nina D.Ziv and B. Mulloth. An Exploration on Mobile Social Networking: Dodgeball as a Case in Point. In *Proceedings of the 2006 International Conference on Mobile Business*, pages 21–21, Copenhagen, Denmark, June 2006. https://doi.org/10.1109/ICMB.2006.8.

[97] X. Ning, C. Desrosiers, and G. Karypis. A Comprehensive Survey of Neighborhood-Based Recommendation Methods. In *Recommender Systems Handbook*, pages 37–76. 2015. https://doi.org/10.1007/978-1-4899-7637-6_2.

[98] A. Noulas, S. Scellato, C. Mascolo, and M. Pontil. An Empirical Study of Geographic User Activity Patterns in Foursquare. In *Proceedings of the 5th International AAAI Conference on Weblogs and Social Media*, pages 570–573, Barcelona, Spain, July 2011.

[99] J. O'Donovan and B. Smyth. Trust in Recommender Systems. In *Proceedings of the 10th International Conference on Intelligent User Interfaces*, pages 167–174, San Diego, CA, USA, Jan. 2005. https://doi.org/10.1145/1040830.1040870.

[100] X. Page, A. Kobsa, and B. P. Knijnenburg. Don't Disturb My Circles! Boundary Preservation is at the Center of Location-Sharing Concerns. In *Proceedings of the 6th International AAAI Conference on Weblogs and Social Media*, pages 266–273, Dublin, Ireland, June 2012.

[101] G. Pallapa, S. K. Das, M. Di Francesco, and T. Aura. Adaptive and Context-Aware Privacy Preservation Exploiting User Interactions in Smart Environments. *Pervasive and Mobile Computing*, 12:232–243, June 2014. https://doi.org/10.1016/j.pmcj.2013.12.004.

[102] G. Pallis, D. Zeinalipour-Yazti, and M. D. Dikaiakos. Online Social Networks: Status and Trends. In *New Directions in Web Data Management 1*, pages 213–234. 2011. https://doi.org/10.1007/978-3-642-17551-0_8.

[103] I. Parris and F. Ben Abdesslem. CRAWDAD data set st_andrews/locshare (v. 2011-10-12). Downloaded from http://crawdad.org/st_andrews/locshare/, Oct. 2011. https://doi.org/10.15783/C7WW2F.

[104] S. Patil, Y. L. Gall, A. J. Lee, and A. Kapadia. My Privacy Policy: Exploring End-User Specification of Free-Form Location Access Rules. In *Proceedings of the International Conference on Financial Cryptography and Data Security*, pages 86–97, Kralendijk, Bonaire, Mar. 2012. https://doi.org/10.1007/978-3-642-34638-5_8.

[105] S. Patil, G. Norcie, A. Kapadia, and A. Lee. "Check Out Where I Am!": Location-Sharing Motivations, Preferences, and Practices. In *Proceedings of the CHI '12 Extended Abstracts on Human Factors in Computing Systems*, pages 1997–2002, Austin, TX, USA, May 2012. https://doi.org/10.1145/2212776.2223742.

[106] S. Patil, R. Schlegel, A. Kapadia, and A. J. Lee. Reflection or Action ?: How Feedback and Control Affect Location Sharing Decisions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 101–110, Toronto, Ontario, Canada, May 2014. https://doi.org/10.1145/2556288.2557121.

[107] J. Pearl. The Causal Foundations of Structural Equation Modeling. In *Handbook of Structural Equation Modeling*, pages 68–91. 2012.

[108] L. Pelusi, A. Passarella, and M. Conti. Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks. *IEEE Communications Magazine*, 44(11):134–141, Nov. 2006. https://doi.org/10.1109/mcom.2006.248176.

[109] H. Polat and W. Du. Achieving Private Recommendations using Randomized Response Techniques. In *Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 637–646, Singapore, Apr. 2006. https://doi.org/10.1007/11731139_73.

[110] P. Pu, L. Chen, and R. Hu. A User-Centric Evaluation Framework for Recommender Systems. In *Proceedings of the 5th ACM conference on Recommender Systems*, pages 157–164, Chicago, IL, USA, Oct. 2011. https://doi.org/10.1145/2043932.2043962.

[111] J. R. Quinlan. *C4. 5: Programs for Machine Learning*. Elsevier, 1992.

[112] R. Ravichandran, M. Benisch, P. G. Kelley, and N. M. Sadeh. Capturing Social Networking Privacy Preferences: Can Default Policies Help Alleviate Tradeoffs between Expressiveness and User Burden? In *Proceedings of the 9th International Symposium on Privacy Enhancing Technologies*, pages 1–18, Seattle, WA, USA, Aug. 2009. https://doi.org/10.1007/978-3-642-03168-7_1.

[113] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, NC, USA, Oct. 1994. https://doi.org/10.1145/192844.192905.

[114] F. Ricci, L. Rokach, and B. Shapira. Recommender Systems: Introduction and Challenges. In *Recommender Systems Handbook*, pages 1–34. 2015. https://doi.org/10.1007/978-1-4899-7637-6_1.

[115] J. J. Rodríguez, L. I. Kuncheva, and C. J. Alonso. Rotation Forest: A New Classifier Ensemble Method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1619–1630, Oct. 2006. https://doi.org/10.1109/TPAMI.2006.211.

[116] Y. Rosseel. lavaan: An R Package for Structural Equation Modeling. *Journal of Statistical Software*, 48(2):1–36, 2012. https://doi.org/10.18637/jss.v048.i02.

[117] C. Ruiz Vicente, D. Freni, C. Bettini, and C. S. Jensen. Location-Related Privacy in Geo-Social Networks. *IEEE Internet Computing*, 15(3):20–27, Feb. 2011. https://doi.org/10.1109/MIC.2011.29.

[118] N. Sadeh, J. Hong, L. Cranor, I. Fette, P. Kelley, M. Prabaker, and J. Rao. Understanding and Capturing People's Privacy Policies in a Mobile Social Networking Application. *Personal and Ubiquitous Computing*, 13(6):401–412, Aug. 2009. https://doi.org/10.1007/s00779-008-0214-3.

[119] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-Based Access Control Models. *Computer*, 29(2):38–47, Feb. 1996. https://doi.org/10.1109/2.485845.

[120] R. S. Sandhu and P. Samarati. Access Control: Principle and Practice. *IEEE Communications Magazine*, 32(9):40–48, Sept. 1994. https://doi.org/10.1109/35.312842.

[121] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. NextPlace: A Spatio-temporal Prediction Framework for Pervasive Systems. In *Proceedings of the 9th International Conference on Pervasive Computing*, pages 152–169, San Francisco, CA, USA, June 2011. https://doi.org/10.1007/978-3-642-21726-5_10.

[122] S. Scellato, A. Noulas, and C. Mascolo. Exploiting Place Features in Link Prediction on Location-Based Social Networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1046–1054, San Diego, CA, USA, Aug. 2011. https://doi.org/10.1145/2020408.2020575.

[123] R. Schlegel, A. Kapadia, and A. J. Lee. Eyeing Your Exposure: Quantifying and Controlling Information Sharing for Improved Privacy. In *Proceedings of the 7th Symposium on Usable Privacy and Security*, number 14, Pittsburgh, PA, USA, July 2011. https://doi.org/10.1145/2078827.2078846.

[124] B. Schmid-Belzt, H. Laamanen, S. Poslad, and A. Zipf. Location-Based Mobile Tourist Services: First User Experiences. In *Proceedings of the 10th International Conference on Information Technology in Travel & Tourism*, Helsinki, Finland, Jan. 2003.

[125] F. Shih, I. Liccardi, and D. Weitzner. Privacy Tipping Points in Smartphones Privacy Preferences. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, pages 807–816, Seoul, Republic of Korea, Apr. 2015. https://doi.org/10.1145/2702123.2702404.

[126] M. Spreitzer and M. Theimer. Providing Location Information in a Ubiquitous Computing Environment (Panel Session). *ACM SIGOPS Operating Systems Review*, 27(5):270–283, Dec. 1993. https://doi.org/10.1145/173668.168641.

[127] J. Staiano, N. Oliver, B. Lepri, R. de Oliveira, M. Caraviello, and N. Sebe. MoneyWalks: A Human-Centric Study on the Economics of Personal Mobile Data. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 583–594, Seattle, WA, USA, Sept. 2014. https://doi.org/10.1145/2632048.2632074.

[128] P. Symeonidis, D. Ntempos, and Y. Manolopoulos. Location-Based Social Networks. In *Recommender Systems for Location-based Social Networks*, pages 35–48. 2014. https://doi.org/10.1007/978-1-4939-0286-6_4.

[129] K. P. Tang, J. Lin, J. I. Hong, D. P. Siewiorek, and N. Sadeh. Rethinking Location Sharing: Exploring the Implications of Social-Driven vs. Purpose-Driven Location Sharing. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, pages 85–94, Copenhagen, Denmark, Sept. 2010. https://doi.org/10.1145/1864349.1864363.

[130] E. Toch. Crowdsourcing Privacy Preferences in Context-Aware Applications. *Personal and Ubiquitous Computing*, 18(1):129–141, Jan. 2014. https://doi.org/10.1007/s00779-012-0632-0.

[131] E. Toch, J. Cranshaw, P. H. Drielsma, J. Y. Tsai, P. G. Kelley, J. Springfield, L. Cranor, J. Hong, and N. Sadeh. Empirical Models of Privacy in Location Sharing. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, pages 129–138, Copenhagen, Denmark, Sept. 2010. https://doi.org/10.1145/1864349.1864364.

[132] E. Toch, J. Cranshaw, P. Hankes-Drielsma, J. Springfield, P. G. Kelley, L. Cranor, J. Hong, and N. Sadeh. Locaccino: a Privacy-Centric Location Sharing Application. In *Proceedings of the 12th ACM International Conference Adjunct Papers on Ubiquitous Computing*, pages 381–382, Copenhagen, Denmark, Sept. 2010. https://doi.org/10.1145/1864431.1864446.

[133] J. Y. Tsai, P. Kelley, P. Drielsma, L. F. Cranor, J. Hong, and N. Sadeh. Who's viewed you?: the Impact of Feedback in a Mobile Location-Sharing Application. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2003–2012, Boston, MA, USA, Apr. 2009. https://doi.org/10.1145/1518701.1519005.

[134] J. Y. Tsai, P. G. Kelley, L. F. Cranor, and N. Sadeh. Location-Sharing Technologies: Privacy Risks and Controls. In *Proceedings of the 37th Research Conference on Communication, Information and Internet Policy*, Arlington, VA, USA, Sept. 2009.

[135] J. Venkatanathan, D. Ferreira, M. Benisch, J. Lin, E. Karapanos, V. Kostakos, N. Sadeh, and E. Toch. Improving Users' Consistency When Recalling Location Sharing Preferences.

In *Proceedings of the 13th IFIP TC 13 International Conference on Human-Computer Interaction – INTERACT 2011*, pages 380–387, Lisbon, Portugal, Sept. 2011. https://doi.org/10.1007/978-3-642-23774-4_31.

[136] S. Vihavainen, A. Lampinen, A. Oulasvirta, S. Silfverberg, and A. Lehmuskallio. The Clash between Privacy and Automation in Social Media. *IEEE Pervasive Computing*, 13(1):56–63, Jan. 2014. https://doi.org/10.1109/MPRV.2013.25.

[137] J. Wang, N. Wang, and H. Jin. Context Matters?: How Adding the Obfuscation Option Affects End Users' Data Disclosure Decisions. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, pages 299–304, Sonoma, CA, USA, Mar. 2016. https://doi.org/10.1145/2856767.2856817.

[138] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The Active Badge Location System. *ACM Transactions on Information Systems*, 10(1):91–102, Jan. 1992. https://doi.org/10.1145/128756.128759.

[139] L.-Y. Wei, Y. Zheng, and W.-C. Peng. Constructing Popular Routes from Uncertain Trajectories. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 195–203, Beijing, China, Aug. 2012. https://doi.org/10.1145/2339530.2339562.

[140] M. Weiser. The Computer for the 21st Century. *Scientific American*, 265(3):94–104, 1991. https://doi.org/10.1038/scientificamerican0991-94.

[141] H. Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009. Online at http://ggplot2.org.

[142] C. Williams, B. Mobasher, R. Burke, J. Sandvig, and R. Bhaumik. Detection of Obfuscated Attacks in Collaborative Recommender Systems. In *Proceedings of the ECAI 2006 Workshop on Recommender Systems*, pages 19 – 23, Riva del Garda, Italy, Aug. 2006.

[143] X. Xiao, Y. Zheng, Q. Luo, and X. Xie. Finding Similar Users using Category-Based Location History. In *Proceedings of the 18th SIGSPATIAL International Conference on*

Advances in Geographic Information Systems, pages 442–445, San Jose, CA, USA, Nov. 2010. https://doi.org/10.1145/1869790.1869857.

[144] X. Xiao, Y. Zheng, Q. Luo, and X. Xie. Inferring Social Ties between Users with Human Location History. *Journal of Ambient Intelligence and Humanized Computing*, 5(1):3–19, Feb. 2014. https://doi.org/10.1007/s12652-012-0117-z.

[145] J. Xie, B. P. Knijnenburg, and H. Jin. Location Sharing Privacy Preference: Analysis and Personalized Recommendation. In *Proceedings of the 19th International Conference on Intelligent User Interfaces*, pages 189–198, Haifa, Israel, Feb. 2014. https://doi.org/10.1145/2557500.2557504.

[146] T. Xu and Y. Cai. Feeling-Based Location Privacy Protection for Location-Based Services. In *Proceedings of the 16th ACM Conference on Computer and Communications Security*, pages 348–357, Chicago, IL, USA, Nov. 2009. https://doi.org/10.1145/1653662.1653704.

[147] Y. Ye, Y. Zheng, Y. Chen, J. Feng, and X. Xie. Mining Individual Life Pattern Based on Location History. In *Proceedings of the 10th International Conference on Mobile Data Management: Systems, Services and Middleware*, pages 1–10, Taipei, Taiwan, May 2009. https://doi.org/10.1109/MDM.2009.11.

[148] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang. Geographical Topic Discovery and Comparison. In *Proceedings of the 20th International Conference on World Wide Web*, pages 247–256, Hyderabad, India, Apr. 2011. https://doi.org/10.1145/1963405.1963443.

[149] J. J.-C. Ying, E. H.-C. Lu, W.-C. Lee, T.-C. Weng, and V. S. Tseng. Mining User Similarity from Semantic Trajectories. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Location Based Social Networks*, pages 19–26, San Jose, CA, USA, Nov. 2010. https://doi.org/10.1145/1867699.1867703.

[150] X. Yu, A. Pan, L.-A. Tang, Z. Li, and J. Han. Geo-Friends Recommendation in GPS-based Cyber-physical Social Network. In *Proceedings of the 2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 361–368, Kaohsiung, Taiwan, July 2011. https://doi.org/10.1109/ASONAM.2011.118.

[151] J. Yuan, Y. Zheng, and X. Xie. Discovering Regions of Different Functions in a City Using Human Mobility and POIs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 186–194, Beijing, China, Aug. 2012. https://doi.org/10.1145/2339530.2339561.

[152] S. Zheng, P. Shi, H. Xu, and C. Zhang. Launching the New Profile on Facebook: Understanding the Triggers and Outcomes of Users' Privacy Concerns. In *Proceedings of the 5th International Conference on Trust and Trustworthy Computing*, pages 325–339, Vienna, Austria, June 2012. https://doi.org/10.1007/978-3-642-30921-2_19.

[153] Y. Zheng. Trajectory Data Mining: An Overview. *ACM Transactions on Intelligent Systems and Technology*, 6(3):29:1–29:41, May 2015. https://doi.org/10.1145/2743025.

[154] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. Urban Computing: Concepts, Methodologies, and Applications. *ACM Transactions on Intelligent Systems and Technology*, 5(3):38:1–38:55, Sept. 2014. https://doi.org/10.1145/2629592.

[155] Y. Zheng, X. Xie, and W.-Y. Ma. GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory. *IEEE Data Engineering Bulletin*, 33(2):32–40, 2010.

[156] Y. Zheng, L. Zhang, Z. Ma, X. Xie, and W.-Y. Ma. Recommending Friends and Locations Based on Individual Location History. *ACM Transactions on the Web*, 5(1):5:1–5:44, Feb. 2011. https://doi.org/10.1145/1921591.1921596.

[157] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining Correlation Between Locations Using Human Location History. pages 472–475, Seattle, WA, USA, Nov. 2009. https://doi.org/10.1145/1653771.1653847.

[158] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining Interesting Locations and Travel Sequences from GPS Trajectories. *Proceedings of the 18th International Conference on World Wide Web*, pages 791–800, Apr. 2009. https://doi.org/10.1145/1526709.1526816.

[159] A. H. Zins and U. Bauernfeind. Explaining Online Purchase Planning Experiences with Recommender Websites. In *Proceedings of the International Conference on Information*

and Communication Technologies in Tourism 2005, pages 137–148, Innsbruck, Austria, Jan. 2005. https://doi.org/10.1007/3-211-27283-6_13.

[160] A. Zipf and M. M. Jöst. Location-Based Services. In *Springer Handbook of Geographic Information*, pages 417–421. 2012. https://doi.org/10.1007/978-3-540-72680-7_21.

[161] A. Zipf and R. Malaka. Developing Location Based Services (LBS) for Tourism – The Service Provider's View. In *Proceedings of the 8th International Conference on Information and Communication Technologies in Tourism 2001*, pages 83–92, Montreal, Canada, 2001.