Year : 2018

PERSPECTIVES ON LOCATION PRIVACY AND MOBILITY PREDICTION WHEN USING LOCATION-BASED SERVICES

Moro Arielle

UNIL | Université de Lausanne

FACULTÉ DES HAUTES ÉTUDES COMMERCIALES

DÉPARTEMENT DES SYSTÈMES D'INFORMATION

PERSPECTIVES ON LOCATION PRIVACY AND MOBILITY PREDICTION WHEN USING LOCATION-BASED SERVICES

THÈSE DE DOCTORAT

présentée à la

Faculté des Hautes Études Commerciales
de l'Université de Lausanne

pour l'obtention du grade de
Docteure ès Sciences en systèmes d'information

par

Arielle MORO

Directeur de thèse
Prof. Benoît Garbinato

Jury

Prof. Olivier Cadot, Président
Prof. Kévin Huguenin, expert interne
Dr. Sonia Ben Mokhtar, experte externe
Prof. Pascal Felber, expert externe

LAUSANNE
2018

UNIL | Université de Lausanne

HEC Lausanne

Le Décanat
Bâtiment Internef
CH-1015 Lausanne

IMPRIMATUR

Sans se prononcer sur les opinions de l'autrice, la Faculté des Hautes Etudes Commerciales de l'Université de Lausanne autorise l'impression de la thèse de Madame Arielle MORO, titulaire d'un Bachelor en Informatique de Gestion de la Haute École de Gestion de Genève et d'un Master en Systèmes d'Information de l'Université de Lausanne, en vue de l'obtention du grade de docteure ès Sciences en Systèmes d'Information.

La thèse est intitulée :

PERSPECTIVES ON LOCATION PRIVACY AND MOBILITY PREDICTION WHEN USING LOCATION-BASED SERVICES

Lausanne, le 04 juillet 2018

Le doyen

Jean-Philippe Bonardi

Jury

Professor Benoît Garbinato
Professor at the Faculty of Business and Economics of the University of Lausanne.
Thesis supervisor.

Professor Olivier Cadot
Professor at the Faculty of Business and Economics of the University of Lausanne.
President of the Jury.

Professor Kévin Huguenin
Professor at the Faculty of Business and Economics of the University of Lausanne.
Internal expert.

Dr. Sonia Ben Mokhtar
Researcher at CNRS and head of the Distributed Systems and Information Retrieval group of INSA Lyon (DRIM Research Group).
External expert.

Professor Pascal Felber
Professor at the Computer Science Department of the University of Neuchâtel.
External expert.

University of Lausanne
Faculty of Business and Economics

Doctorate in Information Systems

I hereby certify that I have examined the doctoral thesis of

Arielle MORO

and have found it to meet the requirements for a doctoral thesis.
All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature: ______________________ Date: 5 juillet 2018

Prof. Benoît GARBINATO
Thesis supervisor

University of Lausanne
Faculty of Business and Economics

Doctorate in Information Systems

I hereby certify that I have examined the doctoral thesis of

Arielle MORO

and have found it to meet the requirements for a doctoral thesis.
All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature: ______________________ Date: 5 juillet 2018

Prof. Kévin HUGUENIN
Internal member of the doctoral committee

University of Lausanne
Faculty of Business and Economics

Doctorate in Information Systems

I hereby certify that I have examined the doctoral thesis of

Arielle MORO

and have found it to meet the requirements for a doctoral thesis.
All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature: Benmokhtar Date: 5 juillet 2018

Dr. Sonia BEN MOKHTAR
External member of the doctoral committee

University of Lausanne
Faculty of Business and Economics

Doctorate in Information Systems

I hereby certify that I have examined the doctoral thesis of

Arielle MORO

and have found it to meet the requirements for a doctoral thesis.
All revisions that I or committee members
made during the doctoral colloquium
have been addressed to my entire satisfaction.

Signature: P. Fel Date: 5 juillet 2018

Prof. Pascal FELBER
External member of the doctoral committee

Abstract

The democratization of the use of mobile devices and the creation of location-based services have modified our perception of the notion of privacy. Mobile devices and, more specifically, smartphones contain positioning systems that enable them to be located almost continuously. Location-based services are mobile applications that use location has a key parameter to provide information to users. They are extremely useful on a daily basis: *Google maps* can be used to easily find a meeting point; *Uber* can be used to find a ride to a destination; the mobile application of the public transportation of our city can be used to find the next metro/bus departures; to name only a few examples. Our mobile devices are constantly with us and record our location history. When this information is shared - even partially - with the location-based services, they can know much about us by exploring our history, deriving information about our politics, religion, and health, among others. This leads to privacy issues. In order to enrich the content location-based services provide to end-users or to simply create new content, these services can use mobility prediction of users. This is even more critical because these services must build predictive mobility models related to the users and need their entire location history to reach this prediction goal. Various solutions have been proposed in the literature to protect the privacy of users. However, they are not always adapted to protect it when the users use location-based services because of their lack of flexibility and because the users cannot have a fine-grained control over the privacy protection.

In this thesis, we address the location privacy issue related to users using location-based services, potentially including mobility prediction. We also explore the mobility behavior of users by analyzing the effects of spatio-temporal and demographic characteristics on their mobility rhythms.

In the first part of this thesis, we present a privacy-aware architecture that enables users to share frequently visited zones with location-based services. The users have the possibility to select the granularity of the shared zones. We also propose a new estimator of the level of location privacy contained in a sequence of locations shared with a location-based service. We evaluate this estimator by comparing its results with the level of success of different localization attacks performed on the sequence of shared locations. We also demonstrate how to use the estimator by applying three different privacy preserving protection mechanisms on the raw sequence of locations of users.

In the second part of this thesis, we add the mobility prediction to the analysis of the location privacy of users (when they use location-based services). We first present a system that enables us to compute future locations of users with a low number of locations, in order to reach the goal of building mobility prediction models on mobile devices. Then, we propose an architecture that contains a system that enables us to predict future location of users for location-based services. This architecture preserves location privacy of users and location-based service utility. The users can have a fine-grained control of their privacy because they must indicate the maximum level of location privacy they want to sacrifice when they use a service.

Finally, in the last part of this thesis, we focus on a privacy-aware mobility behavior analysis. This analysis is based on the evolution of spatio-temporal entropy of users and not on the analysis of movements amongst sensitive places visited by users as it is usually done in existing research works. In addition, we use Generalized Additive Models (GAMs) to study the effects of variables that can explain and predict mobility behavior of users.

Keywords - *Location privacy; Mobility prediction; Mobile device; Location-based service; Positioning system; System architecture; Location privacy preserving mechanism; Predictive model; Mobility behavior; Spatio-temporal entropy; Generalized additive model*

Résumé

La démocratisation de l'utilisation des appareils mobiles ainsi que le développement des services géolocalisés ont totalement modifié notre perception de la notion de vie privée. En effet, les appareils mobiles, et plus particulièrement les smartphones, sont dotés de systèmes de positionnement permettant de capturer leur localisation courante pratiquement en continu. Les services géolocalisés, quant à eux, sont des applications mobiles qui fournissent des informations en lien avec les localisations d'un utilisateur. Ils sont généralement extrêmement utiles, tant et si bien qu'ils nous accompagnent tout au long de notre journée: l'application *Google maps* peut être utilisée pour atteindre facilement le lieu exact d'un rendez-vous; l'application *Uber* pour trouver facilement un taxi; l'application mobile des transports publics de notre région pour trouver les prochains départs des métros/bus à proximité de notre position courante, et bien d'autres encore. Par conséquent, si ces appareils sont continuellement avec nous et que tout notre historique de localisations est transféré vers des services géolocalisés, cela constitue une menace majeure à notre vie privée car ils peuvent absolument tout connaître de nous en explorant cet historique. En effet, l'historique de nos localisations contient des éléments sensibles à notre sujet. Les entreprises, qui sont derrière les services géolocalisés, peuvent connaître les lieux que nous fréquentons sur une base regulière et également en extraire de multiples informations critiques: découvrir nos loisirs, notre lieu d'habitation, notre lieu de travail, mais également nos affiliations politiques ou encore nos choix religieux. Cela s'avère encore plus critique lorsque le service géolocalisé en question utilise la prédiction de la mobilité d'un utilisateur pour produire du contenu, car il construit alors un modèle de la mobilité de l'utilisateur, modèle ayant une capacité prédictive. De nombreuses solutions ont été proposées dans la littérature et ne sont pas toujours adéquates pour protéger la vie privée des utilisateurs lorsqu'ils utilisent des services géolocalisés. En effet, ces solutions peuvent souvent manquer de flexibilité en termes d'utilisation et l'utilisateur n'est pas nécessairement partie prenante au processus de protection.

Cette thèse aborde la protection de la vie privée des utilisateurs de services géolocalisés utilisant ou non la prédiction de la mobilité pour produire leur contenu.

Elle explore également les comportements de mobilité d'utilisateurs en analysant les effets des caractéristiques spatio-temporelles et démographiques sur leur rythmes de mobilité.

Dans la première partie de la thèse, nous présentons une architecture visant à partager des zones fréquemment visitées par des utilisateurs à des services géolocalisés, tout en respectant la vie privée des utilisateurs qui vont pouvoir contrôler la granularité des zones partagées. Nous proposons également une nouvelle mesure permettant de calculer le niveau de vie privée que les utilisateurs dévoilent à travers le partage de leur localisations avec les services géolocalisés. Cette mesure a été évaluée en comparant ses résultats avec le succès de plusieurs attaques réalisées sur des localisations d'utilisateurs protégées par différents mécanismes de protection.

Dans la deuxième partie de la thèse, nous ajoutons la prédiction de la mobilité à nos analyses. Nous présentons premièrement un système permettant de calculer les futures localisations des utilisateurs avec un faible nombre de localisations avec pour but final d'estimer s'il est possible de construire des modèles prédictifs de mobilité sur des appareils mobiles. Puis, nous proposons une architecture contenant un système permettant de prédire les futures localisations d'utilisateurs à des services géolocalisés tout en visant à protéger la vie privée des utilisateurs et à maintenir une bonne utilisation des services géolocalisés. Les utilisateurs sont acteurs de ce système car ils doivent indiquer le niveau maximal de vie privée qu'ils acceptent de sacrifier lors de l'utilisation d'un service.

Pour terminer, la dernière partie de la thèse se focalise sur les comportements de mobilité et leur analyse à partir du calcul de l'entropie spatio-temporelle adapté au domaine de la mobilité. Cette analyse est respectueuse de la vie privée des utilisateurs analysés car elle ne va en aucun cas extraire et analyser des informations sensibles liées aux utilisateurs, leur lieu de travail ou d'habitation par exemple. De plus, nous utilisons les modèles additifs généralisés, i.e., Generalized Additive Models (GAMs), afin d'explorer les variables qui ont une influence sur les comportements de mobilité des utilisateurs analysés et de prédire leur rythmes de mobilité.

Mots clés - *Protection de la vie privée; Prédiction de la mobilité; Appareils mobiles; Services géolocalisés; Systèmes de positionnement; Architecture système; Mécanismes de protection de la vie privée; Modèles prédictifs; Comportement de mobilité; Entropie spatio-temporelle; Modèles additifs généralisés*

Acknowledgements

First and foremost, I would like to thank my supervisor, **Professor Benoît Garbinato**, for guiding me during the long road of this thesis. I learned a lot under his supervision, how to develop ideas but also how to improve my writing skills in order to translate these ideas into scientific papers. He gave me the chance to do a PhD research in the best possible conditions and to explore a subject that is very important to me and extremely interesting. As I approached the end of my thesis research, he also gave me the opportunity to launch a data collection campaign, which was crucial in order to obtain the best expected data for my last paper. This thesis was a long road with multiple challenges, which helped me to become more confident by leaving my confort zone.

Secondly, I would like to thank **Dr. Sonia Ben Mokhtar**, **Professor Pascal Felber** and **Professor Kévin Huguenin** for having accepted to be the members of my jury. They provided me very interesting and valuable comments in order to improve the quality of my thesis. In addition, I thank **Professor Olivier Cadot** for having accepted to be the president of this jury.

I will also never forget with whom I learned my first computing skills and my passion for programming applications. For all of this, I would like to thank **HES Professor Peter Daehne**, **Senior Lecturer Michel Kuhne** at HEG Geneva and **Dr. Atika Laribi** who gave me the passion for learning new programming langages and creating algorithms, as well as the desire to continue my studies!

During my thesis research, I had the pleasure of working with **Professor Valérie Chavez-Demoulin** from the department of Operations at HEC Lausanne. I very warmly thank her for all her support and the knowledge I learned with our collaborative work.

I also had the chance to work with the most pleasant colleagues: **Vaibhav Kulkarni** and **Bertil Chapuis**. We collaborated on several research projects together and this was extremely enriching. I will sorely miss our office and our debates!

During my master's thesis research, I had the pleasure of working with **François Vessaz** and **Adrian Holzer**, discovering research work in distributed systems. This was decisive in my choice to do a thesis.

A sincere thank you also goes to all my former and current colleagues of the department and of the University of Lausanne: **Behnaz**, **Shabnam**, **Dina**, **Hazbi**, **Natali**, **Dana**, **Clément**, **Virginie**, **Marie**, **Natasha**, **Gaël**, **Martin**, **Matthieu**, **Benjamin**, **Nico**, **Gabriela**, **Flora**, **Perrine**, **Louis**, **Gianluca**, **Alain** and **Bastien**. In addition, I thank all members of the **PhDnet association**. Moreover, I give a special thank you to **Sarah Duplan**, **Caroline Kleinheny**, **Michel Schuepbach**, **Katja Schwab-Weis** and also **Anina Eggenberger** who were always present to help me during these years.

Along with doing research for this thesis, I was also a teaching assistant and I would like to thank all the students I had in class or during their master's thesis. I really enjoyed working, sharing knowledge, and discovering new ways to solve programming issues with them.

During my master's thesis and my PhD research at the department of Information Systems, I met very kind and interesting professors and I would like to warmly thank them as well: **Professor Thibault Estier**, **Professor Christine Legner** and **Professor Yves Pigneur**. I had the privilege of meeting several other inspiring professors at conferences, I would like to express my gratitude to them also.

Some of the research papers presented in this thesis were partially funded by the **Swiss National Science Foundation** in the context of the Project n°146714. I thank them for their support.

Finally, the last but not the least, I would like to thank my family very much for all the unconditional support I had during all my studies. My parents always helped me to find the domain that interested me the most by letting me try, sometimes fail, and learn from these experiences. They also gave me all the means to reach my goals in the best possible conditions. I will also have a thought for my two grandmothers who left me last year, I would have liked to share more things with them, especially at the end of this thesis work.

I saved the best for the end, I thank Lionel for all his support during this long road, we will have time to prepare and reach our common business and sporting goals now :)

Contents

Part I Location Privacy

Part II Location Privacy and Mobility Prediction

Part III Mobility Behavior Analysis

Year : 2018

PERSPECTIVES ON LOCATION PRIVACY AND MOBILITY PREDICTION WHEN USING LOCATION-BASED SERVICES

Moro Arielle

Chapter 1
Introduction

> *"Recent inventions and business methods call attention to the next step which must be taken for the protection of the person, and for securing to the individual what Judge Cooley calls the right* ***to be let alone****."*
> *The right to privacy, Samuel D. Warren and Louis D. Brandeis*
> *Harvard Law Review, Vol. 4, No. 5 (Dec. 15, 1890), pp. 193-220*

The word *privacy* appeared for the first time in a law article, published in the Harvard Law Review in 1890 [22], written by Samuel Warren and Louis Brandeis. More specifically, they advocated the *right to be let alone*. Since this date, the perception of this notion has continuously evolved, according to the evolutions of technologies in various domains. Major technological evolutions are, amongst others, linked to the evolution of devices, e.g., cameras and mobile devices, but also to the evolution of the means of communication, the Internet for example. In addition, the geo-political climate and terrorist attacks have contributed to the change of the perception of privacy by the population, and this is linked to the mass surveillance of people managed by governments, supposedly to protect citizens. The development of social networks has also changed the notion of privacy because, they are extremely valuable: distance is no longer a barrier between content-producers and content-consumers. Although social networks facilitate communication amongst users, they can easily capture personal information and extract value from them. Parallel to these evolutions, the legal framework has continuously been updated to be aligned with technological progress; it is still the case today with the European General Data Protection Regulation (EU GDPR[1]). History shows that there is always a significant delay, due to the discovery of new threats and/or abuses coming from these novelties, between the democratization of a new technology and the alignement of the legal framework. Today, this notion has recovered its popularity because of all the well-known privacy scandals, involving big companies and third parties, such as Wikileaks[2], Edward Snowden or, more recently, Cambridge Analytica/Facebook. All these cases triggered a high number of debates around the notion of *privacy* and, more importantly, how to efficiently protect citizens all over the world. We will now

[1] European General Data Protection Regulation description: `https://www.eugdpr.org/`

[2] Official website of Wikileaks: `https://wikileaks.org`

talk about mobile devices and location-based services and their impact on privacy in order to get closer to the subject of this thesis.

Location Privacy, Mobile Devices and Location-Based Services.

Two major technological evolutions contributed to the modification of the perception of privacy: The democratization of mobile devices and the creation of location-based services running on those devices. People did not pay attention to the privacy loss resulting from these two technological evolutions, probably due to the lack of knowledge and/or the absence of information about exposed threats. In truth, there was a complete privacy paradigm shift that created an open door to strong privacy losses. Mobile devices are with us most of the time, thus they can be used to track us constantly. These devices are able to locate themselves due to the location tracking system located on the operating system layer. According to the last statistical trends about desktops and mobile devices worldwide, the market share of mobile device users exceeds that of those of desktops.[3] These statistics confirm the popularity of mobile devices. The presence of location tracking systems in the mobile devices boosted the development of location-based services that use location as a key element for providing information to users. The use of location-based services is part of our daily lives, for example Google maps, Facebook, Instagram, Uber and Snapchat, as well as public transportation applications. To operate properly, location-based services must capture user locations with accuracy, which leads to a real location privacy threat. Indeed, if users cannot control the access to their locations, there is a clear privacy threat for them. All movements of a user define her behavior, as all her frequently visited places describe her tastes and other very private information. In summary, the information about a user's life is contained in her location history.

Depending on the mobile device model, a user can have means to control the access to or accuracy of her locations that are provided to location-based services. For example, iPhone users can specify if an application can always have the access to her locations, only when the application runs in the foreground, or never (iOS version 11.4). Whereas, Android users can modify the granularity of the locations captured by applications with several basic pre-defined options or also simply disable the location tracking (Android Marshmallow). Although all these privacy controls already exist, they do not truly benefit the users because the range of the location privacy preserving options proposed to them are not sufficient, and users sometimes do not know how to change these privacy parameters. Some location-based services are very convenient to use on a daily basis and, we must admit, it is sometimes difficult to avoid them, especially due to social pressure.

[3] Worldwide market share of desktops and mobile devices from March 2017 to March 2018: `http://gs.statcounter.com/platform-market-share/desktop-mobile/worldwide/#monthly-201703-201803`

In order to create new contents or to increase the quality of the existing contents displayed to users on their device, location-based services also try to take into account the predicted locations of users, such as their next movements or those in the future. To do so, they must build mobility prediction models of users. This requires a complete history of the user's movements; this is highly critical in terms of privacy because location history contains sensitive information. Thus, new architectures need to be created to facilitate this prediction goal from the point of view of location-based services and to still preserve user location privacy. More broadly, it is crucial to think how it could be possible to provide users a more fine-grained control of their location privacy when they use location-based services.

Therefore, we will present several research works in this thesis; they explore several perspectives about location privacy but also about the mobility behavior of users, including prediction. More formally, these works contain new system architectures regarding location privacy and/or mobility prediction, but also new methodologies for computing location privacy and for studying mobility behavior of users.

The remaining sections of this chapter are organized as follows. In Section 1.1, we present the background and several definitions of key terms used in this thesis. In Section 1.2, we detail the overall research question of the thesis as well as its subquestions. Then, in Section 1.3, we describe the research methodology used during the thesis. Finally, we conclude this chapter with a thesis overview and a list of publications written during this thesis in Sections 1.4 and 1.5 respectively.

1.1 Background and Definitions

As mentioned in the previous section, the key terms must be defined before going into more details in the thesis.

1.1.1 From Privacy to Location Privacy

Privacy is difficult to define because it relates to multiple domains that can be combined, such as law and computer science. As mentioned at the beginning of the introduction, privacy is primarily a right, the *right to be let alone*, as stated in [22]. In [20], Smith et al. present various definitions of privacy found by reviewing a large number of existing works around this notion. One of the definitions is extremely interesting: Privacy is defined as *the ability to manage information about oneself*, which is obviously linked to the notion of control of the information. If we change this definition to location privacy and, more specifically to the notion of sharing locations with location-based services, this becomes *the ability to manage how locations are* *shared with location-based services.* In [3], Duckham defines location privacy as *the right of individuals to control the collection, use, and communication of personal information about their location.* For this thesis, we choose the second rephrased definition because we focus mostly our research on the sharing of locations with location-based services. In order to control the sharing of her locations, a user must have the possibility to have different ways to share her locations by deciding what kind of information the location-based services should know about her. Below, we give some additional definitions of important terms linked to the privacy domain.

Location Privacy Preserving Mechanism.

A location privacy preserving mechanism is a mechanism that protects the location privacy of users. In [19], Shokri et al. define two categories of location privacy preserving mechanisms: in the first category, mechanisms are used to obfuscate a sequence of raw locations of users, and in the second category, mechanisms are used to anonymize this sequence. In this thesis, we only focus on the first category of location privacy preserving mechanisms. In this first category of mechanisms, we find various techniques for protecting locations, such as perturbing a location by adding noise, reducing the precision of a location, deleting locations or adding fake locations. Location perturbation can be achieved with different techniques, as described by Krumm in [7]. For example, spatial rounding transforms the coordinates of a main location into the coordinates of the closest vertex of a cell in which the main location is located. To make this transformation, we discretize space by creating a grid. We can also use spatial cloaking or Gaussian noise mechanisms to obfuscate location data. In this thesis, we use the term *blurring mechanisms* as a synonym of *obfuscation mechanisms.*

Adversary Model.

In [19], Shokri et al. indicate the importance of defining an adversary model when a location privacy preserving mechanism must be assessed. In other words, this definition enables to clearly define the threat a user will face. In the papers presented in this thesis, we use *threat model* as a synonym of *adversary model.* We must clearly state what the adversary is, its knowledge, as well as its goal that is expressed as the attack it will perform. In our context and throughout this thesis, the adversary is obviously a location-based service that collects and uses locations of a user. The knowledge of an adversary is crucial because this has a clear impact on the success of the attack it will perform. The more knowledge it has, the higher the success of the attack will be. According to [19], there exist two categories of attacks: presence / absence disclosure attacks and meeting disclosure attacks. The first category of attacks tries to create links between users and locations, whereas the other tries to find relationships amongst users. In the first category, there are two main sub-categories of attacks: tracking attacks and localization attacks. Tracking attacks consist in discovering a complete or a partial stream of locations of a user. In this thesis, we mainly focus on localization attacks, which consists in finding sensitive locations of a user at a given time.

1.1.2 From Locations to Mobility Prediction

As mentioned at the beginning of the introduction, to display new or more accurate content to users on their mobile devices, the next step of location-based services is to include mobility prediction as a key element. In order to achieve this location prediction goal, such services must create mobility prediction models to be able to compute the future locations of users. It usually starts with the location history of a user; the history contains the raw locations that are then processed to create the user's mobility prediction model that can then be queried in order to obtain her future locations. There exist various techniques for processing locations, these techniques clearly depend on the targeted prediction goals and their related models. In [5], one of the most intuitive models is described; it is a mobility Markov Chain model. A Markov Chain model contains states and transitions amongst them, where states and transitions are discovered by exploring the location history of a user. This structure basically helps to compute the subsequent state that will be reached by a user, according to her current state. The states describe frequently visited places by a user, commonly called clusters of locations. These clusters can share common spatial and/or temporal characteristics. In [4], Gambs et al. describe the main ways to compute these clusters. According to the above description, Markov Chain models may be limited in terms of prediction goal, unless we choose to extend the states of the Markov Chain with additional temporal characteristics, in order to be able to handle other types of prediction requests. In [6], Hendawi and Mokbel propose an extensive survey of existing queries and prediction functions including their related prediction models. Some functions are extremely interesting for computing short-time predictions, whereas others need a sufficiently long location history to provide long-term predictions. In addition, we can also use other predictors, such as decision tree models, classification-based learning models, e.g., k-Nearest Neighbors method (k-NN), or Artificial Neural Network (ANN) models. However, it is sometimes difficult to clearly understand mobility behaviors by using these models. For example, these models not give details regarding the influence of seasonal variables on a predicted variable. In order to better understand user mobility behaviors, there exist Generalized Additive Models (GAMs), with which it is possible to see the impact of variables on a predicted variable. In [18], Pearce et al. show how it is possible to use GAMs to explain relationships between events: for example, the influence on local meteorology on air quality. Hence, in the context of mobility behavior, GAMs could be used to answer the following question: What are the days that influence specific movements of a user?

1.2 Problem Statement and Research Questions

The main research question explored in this thesis is the following: *What kinds of algorithms and system architectures can preserve users' location privacy when they use location-based services that potentially include mobility prediction?* This question contains two main elements: The first element focuses on location privacy and the second element is linked to the prediction of mobility of users. In several works presented in this thesis, these two subjects are combined. Both subjects can also be related to the analysis of mobility behavior of users. Consequently, we chose to separate our research into three different parts: *location privacy*, *location privacy and mobility prediction* and *mobility behavior analysis.* Each part contains its own group of research questions, that are described hereafter.

Part I - Location Privacy.

Location-based services need locations to operate properly and to offer reliable services to users. As they need them, the first research question is focused on system architectures that could exist on the mobile device of a user in order to protect her privacy, without including the notion of mobility prediction.

Q1: *How is it possible to extract and model the transformations that result from the application of a blurring mechanism on a sequence of locations and include them into a metric for quantifying location privacy?*

Location privacy preserving mechanisms, more specifically blurring mechanisms, can affect the spatial and temporal dimensions of the location stream of a user. These transformations can generate spatio-temporal uncertainties when they are shared with location-based services because they are created to protect the location privacy of the user. However, the existing means of quantification are mainly focused on the effect of specific attacks but not necessarily on the inherent location privacy value of the locations shared by the user with a location-based service. If a user sends her raw locations, the value of the locations shared is very high, whereas when a blurring mechanism is applied on them before being shared with a location-based service, the value decreases. The first challenge is to analyze different blurring mechanisms and how they transform a sequence of locations of a user. The second challenge is to create a location privacy metric that includes these transformations. Finally, the last challenge is to assess this metric because it is extremely complex to compare different location privacy metrics. This is due to the fact that they belong to different contexts described by different adversary models.

Q2: *How can users control their location privacy when they are using location-based services?*

According to the literature, several algorithms have been proposed to protect the location privacy of users. In addition, some privacy parameters are present at the operating system level of mobile devices (e.g., iOS and Android), which can have an impact on the location tracking quality obtained by location-based services when they want to obtain and use the locations of users. For example, they give the option to select a basic pre-defined location tracking accuracy level to users. However, these existing options are not very flexible and do not enable users to control their privacy in a fine-grained manner. Our goal is first to design a mechanism that enables a fine-grained control of location privacy, and second to create a system architecture that will include this mechanism. At this level, the goal is to analyze what kind of locations or group of locations could be shared with a location-based service, instead of sending a complete stream of locations, and to design an architecture that could support it.

Part II - Location Privacy and Mobility Prediction.

This research question group merges both location privacy and mobility prediction, because mobility prediction is a real added-value for location-based services that enriches the content they provide to users. To preserve user location privacy, we need to create and propose new system architectures, hence it is important to analyze the specific context of mobility prediction in the context of location-based services.

Q3: *Is it possible to create mobility prediction models on local devices in order to preserve the location privacy of users?*

If location-based services need mobility prediction to operate, they will have to create predictive models that represent the user mobility. If they must create a predictive model of a user, they will have to access the complete user location history in order to build it, which is an important privacy issue. Consequently, we want to assess to which extent a predictive model of a user can be created in real time in the trusted part of her mobile device, i.e., operating-system layer. This question leads to two research sub-questions.

Q3.1: *To which extent could we avoid using a large amount of data to obtain relevant location predictions?*

In order to create and train a mobility prediction model, we usually use a large amount of data, e.g., 90% of a user dataset. If we want to create predictive mobility models on mobile devices, we must consider that they have limited resources, for example, power consumption. Intuitively, we must verify whether it is possible to avoid using a large volume of location data in order to create a predictive model that provides accurate predictions.

Q3.2: *How can major changes in the mobility of a user be detected?*

This question is linked to the update of the predictive mobility model, because the system must be always aligned with the current mobility of the users and cannot gather a large amount of locations to balance it automatically. To update the prediction model of a user, we must find mechanisms that detect major changes that occur in the mobility of the user.

Q4: *How can the location privacy of users be ensured while the location-based service utility is maintained in a mobility prediction context?*

The utility of a location-based service is crucial for the user because it will determine if she will continue to use it in the future. For example, if a location-based service provides inaccurate information to the user, she will simply decide to not use it anymore. By considering a location-based service that includes mobility prediction, we will analyze how to protect the location privacy of a user and to maintain the utility of the location-based service. Consequently, there is a tradeoff between location privacy and service utility, from the point of view of a user.

Part III - Mobility Behavior Analysis.

We all agree that our location history contains sensitive information about our lives. Several papers already highlighted this. The next research question focuses more specifically on the discovery of the mobility behavior of a user by exploring her locations and the effect of seasonal and demographic attributes on her movements. This analysis should be done in a privacy-aware manner in order to protect the location privacy of the user. Indeed, several existing works used frequently visited places to explore the mobility behavior of users, which is highly critical in terms of location privacy.

Q5: *Which metric could describe the mobility of a user in a privacy preserving manner?*

For this research, the first challenge is to create a metric that describes the mobility of a user as a rhythm, i.e., without extracting frequently visited places of the user. Then, we must find how this metric can be applied on the location history of a user in order to study her mobility behavior in a second time.

Q6: *What kind of seasonal or demographic variables could affect the mobility behavior of users?*

The second challenge is to select a model that captures the influence of different variables on one specific variable. Therefore, this model will help us to determine the variables (e.g., days of the week, working profile) that have an effect on the mobility of a user or a group of users.

1.3 Research Methodology

In order to explore our research questions and to realize all the work aforementioned, we follow a three-step process that includes the design and implementation of solutions, evaluations of the solutions through tests using real mobility traces of users, and the analysis of the obtained results. It is an iterative process similar to a standard application development process, with the exception that it is entirely research oriented. For our evaluations, we use several well-known datasets such as *Nokia* dataset [12], *Geolife* dataset [23] and *PrivaMov* dataset [1].

In addition, we launched a data-collection campaign, called *Breadcrumbs*, whose location data was used to realize the analysis of the last paper presented in this thesis. As it is a large part of our research, we explain in more detail the data-collection campaign, *Breadcrumbs*, hereafter.

Breadcrumbs Data-Collection Campaign.[4] The reason for our data collection campaign comes from the lack of rich user datasets, in terms of location-tracking frequency, the lack of ground truth related to user datasets (e.g., point of interest validated by the user) and the lack of demographic data that could explain the mobility behavior of users. We collected data for this campaign during three months - from the end of March 2018 to the end of June 2018. We created an iOS application, which was installed on the smartphones of the participants, and a server in order

[4] Breadcrumbs Data Collection Campaign was a collaborative work of three research laboratories of HEC Lausanne (University of Lausanne): Distributed Object Programming Lab (`http://doplab.unil.ch`), Information Security and Privacy Lab (`https://people.unil.ch/kevinhuguenin/`) and Business Information Systems and Architecture Lab (`https://wp.unil.ch/bisa/`)

to store all the data collected during the data-collection campaign. We selected 130 participants who were linked to the campus of the University of Lausanne (UNIL) or the campus of Swiss Federal Institute of Technology (EPFL) in Switzerland. The participants were mainly students; this means that the population was very homogeneous in terms of a working profile (i.e., the majority of them were full-time students) and age group (i.e., the majority of them were between 18 and 27 years old).

1.4 Thesis Overview

We chose a *thesis by publication* format. Each chapter can be read independently because it contains an entire research paper published in the proceedings of a conference or submitted to a journal for the last paper. Although the weaknesses of this format are that it can lead to content redundancies and some small terminology differences across the different papers, the advantage is that the thesis has self-contained chapters. The structure is composed of three main parts, following the three groups of research questions, described in Section 1.2.

Part I - Location Privacy.

This part presents our contributions to preserve location privacy of users when they interact with location-based services. We recall that this part does not include mobility prediction. This part is linked to research questions **Q1** and **Q2**, detailed in Section 1.2.

Chapter 2. A System-Level Architecture for Fine-Grained Privacy Control in Location-Based Services [13].

We address **Q2** in this chapter by presenting a new location privacy preserving architecture that enables users to use location-based services that require points of interest to provide location-aware information, as presented in paper [13]. We consider zones of interest of a user as her frequently visited places that are defined with a location and a radius. This architecture includes a service called *ShareZ*: it contains a component called a *privacy tree*, thus enabling us to create a tree of different levels of zones of interest. These levels are automatically computed, according to the number of zones of interest of a user. This tree structure can be used to send only specific zones of interest to location-based services that need them to operate instead of discovering them by exploring the location history of a user. The radius of these zones of interest corresponds to a level of the tree that is aligned with the location privacy preference given by the user. We present a comparison of existing blurring mechanisms with different levels of the proposed tree structure. We highlight that our tree structure provides reasonable privacy protection levels and a real flexibility.

Chapter 3. A Location Privacy Estimator Based on Spatio-temporal Location Uncertainties [14].

In this chapter, we present an answer to the research question **Q1** in [14]. The previous chapter highlighted that it is difficult to properly compare different blurring mechanisms and that this is an important lack in the literature. In this chapter, we present a new metric for describing the effect of different blurring mechanisms on a sequence of raw locations and for computing the location privacy level related to this sequence. We use space and time dimensions to describe a blurring effect on a raw location or several raw locations. In order to properly evaluate our metric, we compare the location privacy level obtained with our metric and the success level of different localization inference attacks on locations protected with three blurring mechanisms. More specifically, we evaluate the level of correlation between the results obtained with our metric and the success of these attacks. The results show that our location privacy metric gives reasonable results and can be used for different types of blurring mechanisms.

Part II - Location Privacy and Mobility Prediction.

This second part contains our contributions for preserving the location privacy of users who use location-based services for which mobility prediction is a key element for them. This part is linked to research questions **Q3**, **Q3.1**, **Q3.2** and **Q4**, detailed in Section 1.2.

Chapter 4. MobiDict - A Mobility Prediction System Leveraging Realtime Location Data Streams [10, 11].

In Chapter 4, we address research questions **Q3**, **Q3.1** and **Q3.2**. We propose a new mobility prediction system called MobiDict : it can process realtime location stream, as described in [10] (we also wrote a poster [11] that contains a description of the discovery of zones of interest with a pseudo-code). In this system, we implement two ways to detect mobility prediction model updates by using mobility behavior analysis. Intuitively, the detection of these updates can be seen as modifications in the mobility pattern of the user, for instance, when a significant change occurs in the set of her frequently visited places, i.e., her zones of interest, in order to efficiently update the models according to these modifications (because we assume that they are strongly linked to the models). By exploring her locations, we analyze the periodicity of movements of a user and the evolution of her zones of interest. We also present a new evaluation method that enables to directly capture prediction accuracies after each major update. These major updates trigger new training windows and new evaluation windows. For the evaluation, we also use several prediction methods in order to highlight the more accurate one. The final results demonstrate that it is also possible to start predicting with a low number of locations and that, theoretically, it could be implemented on a mobile device, preserving the location privacy of the user because the location history is not shared with a location-based service.

Chapter 5. ResPred: A Privacy Preserving Location Prediction System Ensuring Location-Based Service Utility [15].

In this chapter, we address research question **Q4** and present a privacy preserving location prediction system that ensures location-based service utility called *ResPred*, as presented in [15]. This system has two components: the first enables us to predict future locations according to a certain time in the future and the second preserves the location privacy of a user. Location-based services must request the system to obtain future locations of a user, instead of creating the prediction model themselves. The user must indicate the maximum location-based service utility she is willing to sacrifice in order to protect her privacy. And the location-based service must state its location-based service utility in order to operate properly. The evaluation of the system is done mainly from a location privacy/utility perspective, by comparing our location privacy mechanism to other existing blurring mechanisms. We also provide a brief evaluation of the mobility prediction model created in the first component of *ResPred*. The results show that our location privacy mechanism provides good results regarding location privacy and utility evaluations. Thus, the location privacy mechanism proposed helps to find an appropriate tradeoff between the location privacy the user wants to preserve and the location-based service utility that the user uses.

Part III: Mobility Behavior Analysis.

In part three, our contributions are focused on the analysis of mobility behavior and the variables that influence it; in particular, seasonal and demographic variables. We conducted this research by analyzing a user's movements in a privacy preserving manner, more specifically, by exploring the user's movements as rhythms, instead of her movements amongst her frequently visited places. This part is linked to the research questions **Q5** and **Q6**, described in Section 1.2.

Chapter 6. Analyzing Privacy-aware Mobility Behavior using the Evolution of Spatio-temporal Entropy [16].

In this chapter, we address research questions **Q5** and **Q6** and present the paper [16]. We use data obtained with the *Breadcrumbs* data-collection campaign. First, we describe how to extract rhythms from users' locations. To do so, we compute the spatio-temporal entropy at a very fine-grained scale, i.e., one hour. By using Generalized Additive Models (GAMs), we explore the effects of seasonal and demographic characteristics on the evolution of the spatio-temporal entropy of users. We also test the prediction of the mobility rhythms of users by injecting the evolution of spatio-temporal entropy of users into GAMs. The results show that GAMs are not only extremely interesting for predicting the mobility rhythms of users but also for understanding them.

1.5 List of Publications

This thesis is based on four publications published in conferences (workshop or main research track) and one publication submitted to a journal. Other collaborative works are also mentioned below. The list of accepted (or submitted) publications are exposed below in chronological order.

- François Vessaz, Benoît Garbinato, Arielle Moro, and Adrian Holzer. Developing, deploying and evaluating protocols with manetlab. In *Revised Selected Papers of the First International Conference on Networked Systems - Volume 7853*, NETYS 2013, pages 89–104. Springer-Verlag, 2013
- Arielle Moro and Benoît Garbinato. A system-level architecture for fine-grained privacy control in location-based services. In *2016 12th European Dependable Computing Conference (EDCC)*, pages 25–36, Sept 2016
- Vaibhav Kulkarni, Arielle Moro, and Benoît Garbinato. A mobility prediction system leveraging realtime location data streams: poster. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 430–432. ACM, 2016
- Vaibhav Kulkarni, Arielle Moro, and Benoît Garbinato. Mobidict: a mobility prediction system leveraging realtime location data streams. In *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming*, page 8. ACM, 2016
- Bertil Chapuis, Arielle Moro, Vaibhav Kulkarni, and Benoît Garbinato. Capturing complex behaviour for predicting distant future trajectories. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 64–73. ACM, 2016

- Arielle Moro and Benoît Garbinato. A location privacy estimator based on spatio-temporal location uncertainties. In *International Conference on Networked Systems*, pages 322–337. Springer, 2017
- Vaibhav Kulkarni, Arielle Moro, Bertil Chapuis, and Benoît Garbinato. Extracting hotspots without a-priori by enabling signal processing over geospatial data. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'17, pages 79:1–79:4. ACM, 2017
- Arielle Moro and Benoît Garbinato. Respred: A privacy preserving location prediction system ensuring location-based service utility. In *Proceedings of the 4th International Conference on Geographical Information Systems Theory, Applications and Management - Volume 1: GISTAM*, pages 107–118. INSTICC, SciTePress, 2018
- Arielle Moro, Benoît Garbinato, and Valérie Chavez-Demoulin. Discovering demographic data of users from the evolution of their spatio-temporal entropy. *arXiv preprint arXiv:1803.04240*, 2018
- Vaibhav Kulkarni, Arielle Moro, Chapuis Bertil, and Benoît Garbinato. Capstone: Mobility modeling on smartphones to achieve privacy by design (to appear). *Proceedings of the The 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications*, 2018
- Arielle Moro, Benoît Garbinato, and Valérie Chavez-Demoulin. Analyzing privacy-aware mobility behavior using the evolution of spatio-temporal entropy (under review). *Submitted to Knowledge and Information Systems (KAIS) international journal*, 2018

References

[1] Sonia Ben Mokhtar, Antoine Boutet, Louafi Bouzouina, Patrick Bonnel, Olivier Brette, Lionel Brunie, Mathieu Cunche, Stephane D 'alu, Vincent Primault, Patrice Raveneau, Herve Rivano, and Razvan Stanica. PRIVA'MOV: Analysing Human Mobility Through Multi-Sensor Datasets. In *NetMob 2017*, Milan, Italy, April 2017.

[2] Bertil Chapuis, Arielle Moro, Vaibhav Kulkarni, and Benoît Garbinato. Capturing complex behaviour for predicting distant future trajectories. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 64–73. ACM, 2016.

[3] Matt Duckham. Moving forward: Location privacy and location awareness. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, SPRINGL '10, pages 1–3, New York, NY, USA, 2010. ACM.

[4] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. Show me how you move and i will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, SPRINGL '10, pages 34–41, New York, NY, USA, 2010. ACM.

[5] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, MPM '12, pages 3:1–3:6, New York, NY, USA, 2012. ACM.

[6] Abdeltawab M. Hendawi and Mohamed F. Mokbel. Predictive spatio-temporal queries: A comprehensive survey and future directions. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, MobiGIS '12, pages 97–104, New York, NY, USA, 2012. ACM.

[7] John Krumm. Inference attacks on location tracks. In *Proceedings of the 5th International Conference on Pervasive Computing*, PERVASIVE'07, pages 127–143, Berlin, Heidelberg, 2007. Springer-Verlag.

[8] Vaibhav Kulkarni, Arielle Moro, Chapuis Bertil, and Benoît Garbinato. Capstone: Mobility modeling on smartphones to achieve privacy by design (to appear). *Proceedings of the The 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications*, 2018.

[9] Vaibhav Kulkarni, Arielle Moro, Bertil Chapuis, and Benoît Garbinato. Extracting hotspots without a-priori by enabling signal processing over geospatial data. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL'17, pages 79:1–79:4. ACM, 2017.

[10] Vaibhav Kulkarni, Arielle Moro, and Benoît Garbinato. Mobidict: a mobility prediction system leveraging realtime location data streams. In *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming*, page 8. ACM, 2016.

[11] Vaibhav Kulkarni, Arielle Moro, and Benoît Garbinato. A mobility prediction system leveraging realtime location data streams: poster. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 430–432. ACM, 2016.

[12] Juha K. Laurila, Daniel Gatica-Perez, Imad Aad, Blom Jan, Olivier Bornet, Trinh-Minh-Tri Do, Olivier Dousse, Julien Eberle, and Markus Miettinen. The mobile data challenge: Big data for mobile computing research. 2012.

[13] Arielle Moro and Benoît Garbinato. A system-level architecture for fine-grained privacy control in location-based services. In *2016 12th European Dependable Computing Conference (EDCC)*, pages 25–36, Sept 2016.

[14] Arielle Moro and Benoît Garbinato. A location privacy estimator based on spatio-temporal location uncertainties. In *International Conference on Networked Systems*, pages 322–337. Springer, 2017.

[15] Arielle Moro and Benoît Garbinato. Respred: A privacy preserving location prediction system ensuring location-based service utility. In *Proceedings of the 4th International Conference on Geographical Information Systems Theory, Applications and Management - Volume 1: GISTAM*, pages 107–118. INSTICC, SciTePress, 2018.

[16] Arielle Moro, Benoît Garbinato, and Valérie Chavez-Demoulin. Analyzing privacy-aware mobility behavior using the evolution of spatio-temporal entropy (under review). *Submitted to Knowledge and Information Systems (KAIS) international journal*, 2018.

[17] Arielle Moro, Benoît Garbinato, and Valérie Chavez-Demoulin. Discovering demographic data of users from the evolution of their spatio-temporal entropy. *arXiv preprint arXiv:1803.04240*, 2018.

[18] John L. Pearce, Jason Beringer, Neville Nicholls, Rob J. Hyndman, and Nigel J. Tapper. Quantifying the influence of local meteorology on air quality using generalized additive models. *Atmospheric Environment*, 45(6):1328 – 1336, 2011.

[19] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 247–262, Washington, DC, USA, 2011. IEEE Computer Society.

[20] H. Jeff Smith, Tamara Dinev, and Heng Xu. Information privacy research: An interdisciplinary review. *MIS Q.*, 35(4):989–1016, December 2011.

[21] François Vessaz, Benoît Garbinato, Arielle Moro, and Adrian Holzer. Developing, deploying and evaluating protocols with manetlab. In *Revised Selected Papers of the First International Conference on Networked Systems - Volume 7853*, NETYS 2013, pages 89–104. Springer-Verlag, 2013.

[22] Samuel D. Warren and Louis D. Brandeis. The right to privacy. *Harvard Law Review*, 4(5):193–220, 1890.

[23] Yu Zheng, Xing Xie, and Wei-Ying Ma. Geolife: A collaborative social networking service among user, location and trajectory. *Data Engineering*, page 32, 2010.

Part I
Location Privacy

Chapter 2

A System-level Architecture for Fine-grained Privacy Control in Location-based Services

Published in the proceedings of
The European Dependable Computing
Conference (EDCC'16)
Gothenburg, Sweden, September 5-9, 2016
Publisher: IEEE
Distinguished Paper

Abstract We introduce a system-level architecture providing fine-grained control over user privacy, in the context of location-based services accessed via mobile devices. In contrast with most mobile platforms today, users only have coarse-grained control over their privacy, either accepting to unconditionally stream their locations in order to use a service, or renouncing the service altogether. However, not all location-based services do require the same level of location accuracy and the same level of privacy renouncement. With this architecture, the user can adapt the tradeoff between location privacy and location accuracy. To achieve this, our architecture relies on three main elements: a trusted module extending the underlying mobile platform, a secure protocol between that module and untrusted applications offering location-based services, and a tree capturing user's zones of interest and organizing them in various accuracy levels. Untrusted mobile applications no longer receive user locations directly: the trusted module intercepts them to compute user's zones of interest and create the tree. The user can then decide what level of accuracy will be disclosed to what application. We evaluate this architecture from a privacy preserving point of view by comparing well-known blurring mechanisms and the tree included in our architecture.

2.1 Introduction

In less than a decade, mobile devices, in particular smartphones, have radically changed the way we consume digital services, be they web searching, online gaming, media streaming, etc. Basically, we can now access such services whenever we want and wherever we are. Furthermore, the ability of mobiles devices to locate themselves, either via the Global Positioning System (GPS) when outside or via WiFi or 3G/4G when indoor, has given birth to a new breed of digital services, so-called *location-based services.*

2.1.1 For better or for worse?

There is little doubt that location-based services can be very useful, ranging from simple one-shot queries about one's surroundings, e.g., when looking for nearby restaurants using applications such as Google Maps or Apple Maps,[1] to continuously tracking one's movements, e.g., while jogging using applications such as Runkeeper or Runtastic.[2] Location-based services come however at a price: *loss of location privacy.*

It is important to stress that location privacy is just one of many facets of privacy as a whole, yet a crucial one, and that this paper focuses exclusively on location privacy. Indeed, locations generated by mobile devices offer a powerful means to link virtually any data associated with a user to a very tangible aspect of her life: her physical location in the real world. For this reason, most mobile platforms, such as Apple iOS or Google Android, are trying to make users aware of this potential privacy loss, by explicitly asking them whether they are willing to share their locations with applications requesting them, either at installation time or at run time.

Unfortunately, this choice is usually binary: either the application is granted full access to the user locations or no access at all. Furthermore, once access has been granted, the user has no control over what the application will do with her location information. It could theoretically confine this information to the user mobile device, which is almost never the case, or forward it to some backend server, in order to compile statistics, use it later or even sell it to some third party.

A key problem here is that by giving away location information, users are actually revealing a lot about themselves, most of the time without even realizing it. In [22] for instance, it is shown that although most users may occasionally exhibit spontaneous behaviors, their moves bear strong regularity, which leads to high predictability of their future locations. Furthermore, according to [6] only four spatio-temporal coordinates are enough to uniquely identify 95% of the users of the dataset. As mentioned in the paper, this dataset contains locations, location antenna more specifically, of a large number of users, approximately 1.5 M. These locations were caught when users used their mobile device (e.g., when a user receives or initiates a call or a text message). Additionally, another paper [23] demonstrates the consequence of privacy leakage via a quantification of social inference from it. They found 90% inference accuracy concerning social and community relationships with only three week's user

[1] `http://google.ch/maps`, `http://apple.com/ios/maps`

[2] `http://runkeeper.com`, `http://runtastic.com`

data by using their leakage inference framework they create. The dataset used in this paper contains real mobility traces and Foursquare data.

2.1.2 Location accuracy vs. privacy

When it comes to locations, a rather straightforward way to control *privacy* consists in controlling *accuracy* (in this paper the term accuracy is a synonym of precision): the lower the location accuracy, the higher the location privacy. Obviously, knowing that Alice is located within a 100 meters range around the cathedral of Notre-Dame de Paris damages her privacy more than knowing that she stands within a 10 kilometers range (which boils down to simply say that Alice is somewhere in Paris). Furthermore, the privacy level associated with Alice's location is not only dependent on its geographical dimension but also on its time dimension: knowing that Alice was located within a range of 100 meters around Notre-Dame de Paris *between 8:00 am and 9:00 am yesterday* compromises her privacy more than knowing that she was there *between 8:00 am and 8:00 pm, some day last week.*

Based on this privacy-accuracy duality, a key question is the following: what is the level of accuracy a location-based service requires to fulfill its function and hence the level of privacy loss one has to accept to benefit from that service. A related key question is then: *given the level of accuracy required by some service, how can the mobile platform ensure that only the corresponding level of privacy will be lost?* This is precisely the question we address in this paper by proposing a system-level architecture for fine-grained control over privacy, in the context of location-based services. As implied above, this architecture must be implemented at the operating system level.

2.1.3 Contributions and roadmap

As already pointed out, today's mobile platform only offer the binary choice of revealing all or nothing in terms of user locations.[3] With our architecture in contrast, the user can decide what level of location privacy she wants to retain, which then translates to the level of accuracy the location-based service will be able to offer.

The remainder of this paper is structured as follows. After presenting our system model and defining the problem we address in Section 2.2, we describe the conceptual architecture and generic protocol proposed by this architecture to solve this problem in Section 2.3. Section 2.4 then focuses on the notion of privacy tree, which is at the

[3] Most mobile platforms also allow to grant location access to certain applications only when they are running in the foreground (while others can access them even when running in background). Still, users only have the choice to either accept or refuse continuously sharing their locations.

heart of our approach, while Section 2.5 presents key aspects of the implementation of the architecture on a mobile device. Then, Section 2.6 details how the location privacy is ensured by using this architecture. Section 2.7 proposes a quantitative evaluation of the privacy tree and compares it with existing approaches. In doing so, we introduce a spatial and temporal probabilistic measure of the privacy loss induced whenever some location information, even partially inaccurate, is provided to a location-based service. Finally, we discuss research results that are close to our approach in Section 2.8 and conclude the paper with future research directions in Section 2.9.

2.2 System model and problem statement

We consider a user moving on the surface of the earth with a mobile device that has the ability to locate itself, typically via the Global Positioning System (GPS)[4] or some other positioning means, e.g., WiFi positioning (WPS).[5] The architecture of this mobile device is depicted in Figure 2.1: Alice, the user, sits on the top and interacts with both the underlying *trusted operating system* and some untrusted location-based service. At the bottom, the system-level location provider is continuously pushing raw location information to the location access protocol above it. The latter processes this information according to some privacy policy and feeds the resulting altered location information to the untrusted location-based service, which can potentially forward this information across the network to some equally untrusted remote server.

We model the stream of raw locations originating from the location provider as sequence $L = \langle loc_1, loc_2, \ldots, loc_n \rangle$, where $loc_i = (\phi, \lambda, t)$ is a tuple representing an individual location. In this tuple, $\phi, \lambda \in \mathbb{R}$ represents a latitude and a longitude respectively, while $t \in \mathbb{N}$ represents the time when the location was captured. In the following, we sometimes use the notation $loc.\phi$, $loc.\lambda$ and $loc.t$ to designate specific parts of tuple loc. In addition, the duration between two consecutive locations in L does not exceed a constant Δt_{limit}, i.e., $loc_{i+1}.t - loc_i.t \leq \Delta t_{limit}$. This means that locations are captured in a regular manner.

The generic architecture presented in Figure 2.1 allows us to model most (if not all) relevant location privacy approaches. On iOS and Android for instance, the stream of raw locations received by the location access protocol is initially kept safe from the untrusted location-based service client (0). Then, the latter requests access to location information to the operating system (1), either at run time or at installation time. This request is forwarded to Alice in the form of a binary choice, in order to draw her attention on the potential loss of privacy (2). Assuming she

[4] http://www.schriever.af.mil/GPS

[5] http://en.wikipedia.org/wiki/Wi-Fi_positioning_system

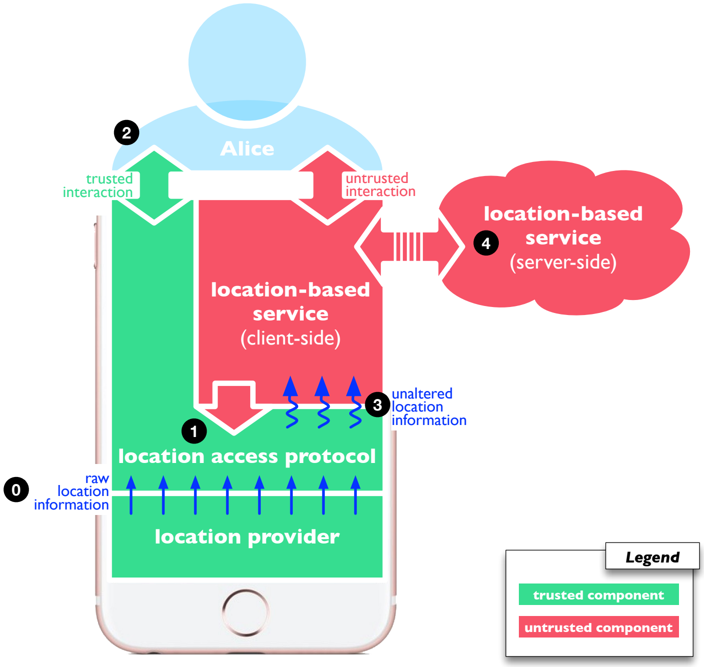


Fig. 2.1: Generic system architecture

accepts to grant access, the stream of raw locations is simply forwarded, unaltered, to the requesting location-based service client (3), which might then propagate it to its remote counterpart (4).

Yet finer-grained access protocols are both desirable and possible, which do not merely block or forward all raw location information. Rather, such access protocols should involve a parameterized alteration of the location information eventually provided to the untrusted location-based service and a more subtle interaction between trusted and untrusted components of the system. Providing such a location access protocol is precisely the problem we address in this paper.

2.3 The proposed architecture including ShareZ service

Our system-level architecture proposes a location access protocol that allows user to share their locations with location-based services, via zones of interest (i.e., frequently visited places by a user) of *various granularities*, in order control the level of privacy loss incurred by this sharing. That is, the notion of *privacy granularity* captures the accuracy-privacy duality inherent to sharing zones of interest: coarse-grained zones offer lower accuracy but higher privacy than sharing fine-grained zones.

Figure 2.2 sketches the proposed architecture, the included service called SHAREZ as well as its location access protocol, which are assumed to be integrated into the underlying mobile operating system, as prescribed by our generic system architecture. To illustrate the description hereafter, we assume that Alice is shopping in Paris and wants to be notified when passing by a shop with special offers. For this, she relies on a mobile application acting as the local client of some location-based shopping service.

(0) ShareZ continuously computes zones of interests of various granularities using raw locations;

(1) The location-based service client application requests location information in the form of zone of interests;

(2) ShareZ asks Alice to choose the privacy granularity and the zones of interest to share with that service;

(3) ShareZ pushes the selected zones with the chosen privacy granularity to the service;

(4) The client forwards those zones to the server, which computes related contextual information, i.e., coupons associated with special offers in our example;

(5) The server sends the contextual information with its period(s) of validity related to the zones it received back to the client;

(6) The client pushes this contextual information to ShareZ, with the associated zones;

(7) ShareZ monitors incoming raw locations and checks if they match some valid contextual information according to its period(s) of validity;

(8) As soon as ShareZ receives a raw location positioned in a zone associated with some contextual information, a coupon in our example, the latter is directly displayed to Alice by ShareZ;

(9) At this point, Alice might or not decide to act based on this information, e.g., use the coupon to benefit from some special offer. If she does, the location-based service has finally the ability to precisely locate Alice, directly or indirectly. If she however decides not to use the coupon, the service will never know that Alice was in that area.

It is crucial to indicate that the Steps 1, 2, 3 and 4 are carried out at the installation of the location-based service client application on the mobile device of the user. In addition, the user can update the zones of interests to this application at any time.

One might of course argue that knowing Alice's zones of interest is already an important breach in her privacy. Note however that Alice has the ability to prevent certain zones from being disclosed to the location-based service and to blur the zones she accepts to share, based on the privacy granularity she chose. In addition, a rather simple strategy that ShareZ can apply to further protect Alice's privacy consists in adding one or more fake zones to those passed to the service in Step 3.

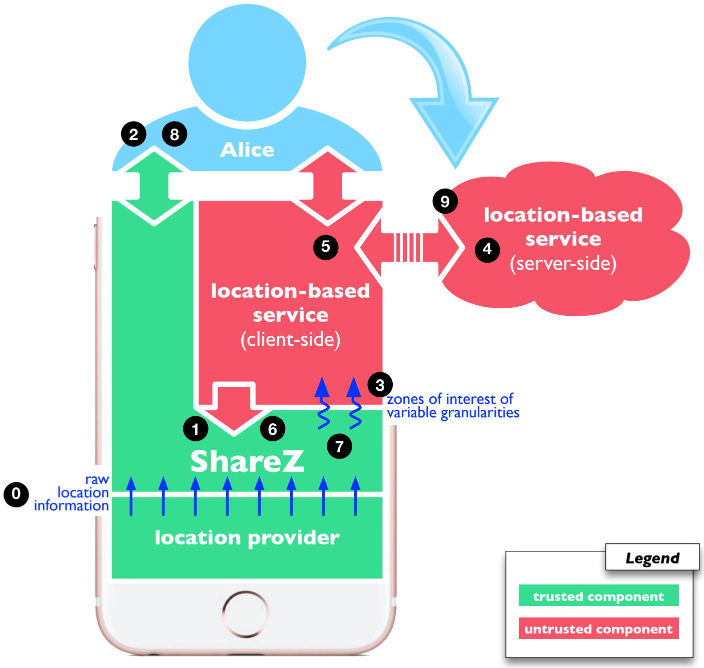

2
8
Alice
9
location-based service (server-side)
4
5
location-based service (client-side)
3
zones of interest of variable granularities
1
6
7
ShareZ
0
raw location information
location provider
Legend
trusted component
untrusted component

Fig. 2.2: The ShareZ location access protocol

Of course, such fakes zones are completely ignored during the location monitoring in Step 7. Yet the location-based service has no way to distinguish real zones of interest from fake ones (see research works in [5, 13]).

In Step 2, it is indicated that Alice must choose a privacy granularity as well as the zones of interest that she wants to share with the location-based service. This means that Alice expresses the maximum accuracy she accepts to reveal about her location, which is also the minimum privacy. In our internal architecture, this translates into a level in the privacy tree. In terms of user interface, a map highlighting the different privacy tree levels could be presented to Alice in order to let her choose this maximum accuracy in a graphical manner. Since her choice may be not adapted to the functionality of the location-based service, she can modify her choice at any time.

The main characteristics of the proposed architecture, including its service ShareZ, and its location access protocol are summarized hereafter.

Local. The protocol minimizes the communication with the location-based service client and hence drastically reduces the risk of location information leakage to remote parties.

Flexible. ShareZ allows users to set distinct privacy preferences for different location-based services and for different contexts, e.g., one might use a certain privacy granularity while shopping and another one when jogging.

Adaptive. ShareZ constantly updates the zones of interest it manages, based on the stream of raw locations received from the underlying location provider.

Encapsulated. ShareZ is the sole recipient of raw locations originating from the location provider and the protocol never forwards them directly to mobile applications.

2.4 Privacy tree

At the heart of the architecture, more specifically in the service ShareZ, and its concept of flexible privacy granularity lies the notion of *privacy tree*. In the following, we first formally define what zones of interest are, as well as the notion of *privacy distance*. We then introduce the notion of privacy tree and its significance when it comes to support various privacy granularities.

2.4.1 Zone of interest

In order to precisely define a zone of interest, we must introduce other fundamental definitions based on the definition of a user location (see Section 2.2) such as the definitions of cluster and cluster group.

2.4.1.1 Cluster

Intuitively, a cluster gathers locations sharing several common characteristics in terms of space and time. Let $\Delta d_{max} \in \mathbb{R}$ be a constant representing a distance (expressed in meters) and $\Delta t_{min} \in \mathbb{N}$ be a minimum time threshold (expressed in seconds). Moreover, we consider the two following functions: $centroid(\{loc_1, loc_2, ..., loc_n\})$, which computes and returns geographical coordinates that represent the centroid of the set of locations passed as parameters (i.e., the barycenter of the locations) and $distance(loc_i, loc_j)$, which simply computes and returns an Euclidian distance between two locations passed as parameters.

We define a cluster l' of L as a subsequence of consecutive locations of L, such that $l' = \langle loc_{s_i}, loc_{s_{i+1}}, \ldots, loc_{e_i} \rangle$. This subsequence must satisfy the two following conditions:

$$\begin{gathered}\forall k \in \{s_{i+1}, \ldots, e_i\}, \\ distance(centroid(loc_{s_i}, \ldots, loc_{k-1}), loc_k) \leq \Delta d_{max}\end{gathered} \tag{2.1}$$

$$loc_{e_i}.t - loc_{s_i}.t \geq \Delta t_{min} \tag{2.2}$$

From this cluster, defined as l', we can extract its centroid $centroid = (\phi, \lambda)$ ($\phi \in \mathbb{R}$ represents a latitude and $\lambda \in \mathbb{R}$ represents a longitude), which is simply the barycenter of all ϕ and λ of the locations contained in l'. We can also compute the radius of the cluster $\Delta r \in \mathbb{R}$, which is the maximum distance between the centroid of the cluster and a location of the set of locations extracted from l'. Therefore a cluster is a tuple $c = (\phi, \lambda, \Delta r, l')$. The notation *c.centroid* is used to designate the centroid of the cluster c. Hereafter, $C = \{c_1, \ldots, c_i, \ldots, c_m\}$ is the set of m clusters associated with the user, based on her sequence of locations. It is important to note that C must meet the following condition:

$$\forall c_i, c_j \in C, c_i.l' \cap c_j.l' = \emptyset \tag{2.3}$$

This first part of the clustering process is based on a well-known technique described in [8] and presented as *density-time cluster (DT cluster).*

2.4.1.2 Cluster group

Intuitively, a group of clusters contains all the clusters that can be gathered iff there exists an intersection between these clusters. Two clusters $c_i, c_j \in C$ are gathered in the same group of clusters g iff we have:

$$\begin{aligned}&distance(c_i.centroid, c_j.centroid) \\ &\qquad - \ (c_i.\Delta r + c_j.\Delta r) < 0\end{aligned} \tag{2.4}$$

So a group of clusters(s) is defined as tuple $g = (\phi, \lambda, \Delta r, \{c_1, c_2, \cdots\})$, where $\phi \in \mathbb{R}$ represents a latitude, $\lambda \in \mathbb{R}$ represents a longitude, $\Delta r \in \mathbb{R}$ represents its radius in meters and the array of clusters from which g is formed. The centroid of a group of clusters is represented by tuple (ϕ, λ), which is simply the mean of the centroids of the clusters in g. Hereafter, $G = \{g_1, g_2, \cdots\}$ is the set of cluster groups associated with the user, based on her set of clusters C.

2.4.1.3 Zone of interest

Intuitively, a zone of interest is a delimited zone that is frequently visited by a user in everyday life. Let $minVisitNb \in \mathbb{N}$ be a constant representing the minimum number of visits and let $visitThreshold \in \mathbb{N}$ that is a maximum threshold of visits. Moreover, let $size(g)$ be a function that computes and returns the number of clusters of the group passed as a parameter and let $meanVisitNb(G)$ be a function that computes and returns the mean number of visits among all the cluster groups passed as parameters. The number, returned by $meanVisitNb(G)$, frequently changes over time depending on the mobility behavior of the user if we consider that the discovery process works in a sequential manner. It is important to note that $minVisitNb$ must be equal to the value returned by $meanVisitNb(G)$ until reaching the $visitThreshold$, which is the maximum number of visits to transform a cluster group into a zone of interest. A group of clusters $g \in G$ becomes a zone of interest z iff it follows Equation 2.5:

$$\begin{aligned} size(g) &\geq minVisitNb \\ \mid \; minVisitNb &= meanVisitNb(G) \\ AND \;\; minVisitNb &<= visitThreshold \end{aligned} \tag{2.5}$$

Formally, a zone of interest z is a tuple $z = (\phi, \lambda, \Delta r, g)$, where $\phi \in \mathbb{R}$ represents a latitude, $\lambda \in \mathbb{R}$ represents a longitude, $\Delta r \in \mathbb{R}$ represents its radius in meters and g the group of clusters. The centroid of z is represented by (ϕ, λ), which is simply the centroid computed from group g. Hereafter, $Z = \{z_1, z_2, \cdots, z_n\}$ is the set of zones of interest associated with the user, based on her set of cluster groups G.

2.4.2 Privacy distance

The *privacy distance* is simply a cursor expressing the *accuracy* of the shared location information, which perfectly illustrates the inherent tradeoff between accuracy and privacy. With ShareZ, the user is responsible for setting this cursor. Let $\Delta d_a \in \mathbb{R}$ be a constant representing this accuracy (in meters): the greater Δd_a, the higher the location privacy offered by zones of interest respecting this privacy distance.

2.4.3 Privacy tree structure

Privacy trees follow a similar hierarchical structure as R-trees, which were introduced by Guttman to index geo-located objects (i.e., leaves of the tree), by grouping and representing them via minimum bounding rectangles at each hierarchical level [10]. In ShareZ, the zones of interest of a user are the leaves of her privacy tree and each hierarchical level closer to the root covers a set of zones found at the lower levels, as illustrated in Figures 2.3 and 2.4. So each level of the tree represents a different privacy granularity.

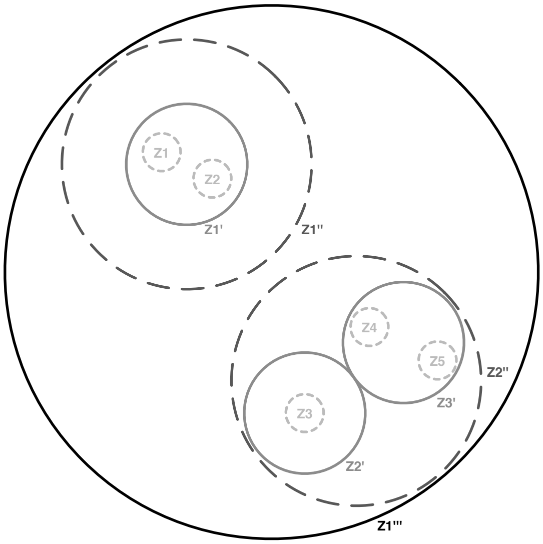


Fig. 2.3: Privacy tree – Spatial structure

Figures 2.3 and 2.4 present a privacy tree based on five zones of interest. The process of building this tree goes as follows. The first step consists in trying to gather close zones by computing the smallest distance between two zones among all of them. A group containing close zones gives birth to a new upper zone with a new centroid. If a zone of interest cannot be gathered with another one, an upper zone is created around it. For example, zone z3 has the same centroid as its upper zone z2'. Note also that the radius of various zones found at a given privacy level are not necessarily equal.

Since this process is realized sequentially in real time (i.e., each time that a new user raw location is caught), it may take time to highlight all zones of interest of the user. In this context and previously mentioned in Section 2.2, the $minVisitNb$ and the $visitThreshold$ enable to discover zones of interest related to the user when she starts using her mobile device. Since $minVisitNb$ corresponds to the current mean number of visits, this value is obviously 0 at the beginning. Then, $minVisitNb$ evolves over time until reaching the $visitThreshold$, which is the maximum number of visits. If this mechanism was not included in the discovery process, the latter would not be able to discover zones of interest of the user from the beginning.

2.5 Implementation of the architecture on a mobile device

The implementation of the architecture has an impact on the two main components presented in Figure 2.2: trusted and untrusted components, i.e., the operating system

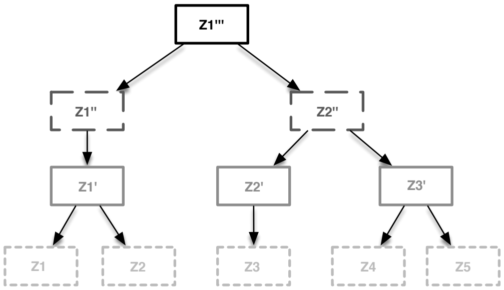


Fig. 2.4: Privacy tree – Tree structure

including the service ShareZ and the location-based service respectively. In this section, we present the key aspects of the implementation of this architecture on a mobile device from the two following points of view: operating system and location-based service.

2.5.1 On the operating system side

In order to implement our architecture at the operating-system level, we need to create ShareZ as well as the application programming interface (API) that can be used by developers to receive zones of interest of user and share contextual information. ShareZ is able to directly interact with the location provider in order to discover the zones of interest of the user and update the related privacy tree. The location provider, which gets location via GPS, WiFi or 3G/4G, provides the stream of raw locations to ShareZ, which is necessary to update the privacy tree. This service must also be able to store the privacy preferences set by the user for each application she is using (e.g., the privacy distance, the zones of interest already shared). In addition, this service must also memorize the contextual information, provided by the location-based service, and display it when it is appropriate according to the current location of the user and the validity period of the content. As described in Figure 2.2, we consider that ShareZ is safe because it is included at the operating system layer, which is also a trusted component. In order to prevent some low level kernel attacks we could also implement a TrustZone technique to isolate critical data transactions but it is out of the scope of the paper. We assume that both the operating system including ShareZ and the location provider are trusted components. Moreover, there is no data alteration during the data sharing between the trusted and untrusted components under the assumption that this exchange is realized in a safe manner. The data sharing implementation is also out of the scope of the paper.

2.5.2 On the location-based service side

If this architecture is implemented at the operating system level, developers of location-based services need to modify the way they obtain user locations. Their code must change in accordance with the new API exposed by the operating system library and, more specifically, by ShareZ. This is already the case when a new version of the API of the operating system is released. Therefore, they must adapt their location-based service to the protocol of the architecture. This has an impact on the way to provide information to the user. In return, a location-based service improves its level of respect for user location privacy. One limit of our architecture is that it does not work with location-based service requiring frequent and precise location updates such as running applications and personal navigation applications. Regarding these specific cases, user should be able to share her raw locations, during a limited period of time, in order to reduce her location privacy loss.

2.6 Location privacy ensured by the architecture

It is important to recall that we focus on location privacy only and not on other user privacy issues. Location privacy also aims at enhancing user privacy. As explained in the previous section, we assume that ShareZ is implemented at the operating system level, which is considered as a safe and trusted component. Considering this, we discuss two main privacy levels, both aiming at ensuring location privacy of the user: at a low level with the privacy tree and at a high level with the architecture protocol.

2.6.1 At the level of the privacy tree

At the privacy tree level, the structure of the tree itself enables to protect the location privacy. With multiple levels of location privacy, computed from the zones of interest of the user, the latter is able to select the most appropriate privacy distance for the location-based service. This tree is frequently updated according to the user mobility behavior. As a result, we cannot predict in advance its structure because it may evolve over time according to the new or outdated zones of interest. In addition, the privacy tree structure is only computed from the stream of user raw location, which is only accessible at the trusted component level. It is also important to indicate that we cannot prevent developers to use other way to obtain user locations such as beacons for instance. However, it is not a privacy threat because there does not exist a sufficient number of beacons used to locate users to extract a complete overview of the mobility behavior of a user.

2.6.2 At the level of the architecture protocol

At the architecture protocol level, the location privacy is mainly ensured by two crucial steps: the sharing of zones of interest (Step 0 to 3 in Figure 2.2) and the sharing of location-based content (Step 4 to 9 in Figure 2.2). Firstly, the sharing of zones of interest (from ShareZ to the location-based service) protects the location privacy of the user, because no raw locations are sent to the location-based service. In addition, even if zones of interest of the lowest level of the privacy tree are shared, they do not correspond to precise locations. The user can also decide the zones of interest that she wants to share with the location-based service, that may reinforce the location privacy. Secondly, the sharing of content, from the location-based service to ShareZ, aims at preserving the location privacy of the user because the content is gathered locally at the trusted component and only displayed when the current location of the user is in the zone of interest linked to this content. Consequently, the location-based service never knows the exact location of the user. However, if the user must interact with the location-based service to use a specific offer, the latter may infer the location of the user by linking the offer selected by the user with her related zone of interest.

2.7 Evaluation

This section has two different goals: (1) presenting a generic location privacy indicator according to space and time dimensions, and (2) comparing our privacy tree solution to three other location privacy preserving mechanisms: *Gaussian alteration*, *sampling* and *spatial cloaking*. It is important to indicate that we decide to assess the low privacy level, which concerns the privacy tree, because it is the heart of our architecture in terms of privacy preserving.

Firstly, we present the dataset we use for our evaluation, which is based on a real life experiment carried out by Nokia between 2009 and 2011. Secondly, we introduce the threat model as well as the privacy indicator. Further, we explain the classification of the users contained in the Nokia dataset. Next, we detail the chosen scenarios as well as the different blurring strategies we implemented. Finally, we present and discuss the obtained privacy indicator results of our experiments.

2.7.1 Nokia dataset

For this analysis, we use a dataset provided by Nokia and containing real mobility traces of users. This dataset was collected in the Lake Geneva region in Switzerland (Europe) from October 2009 to March 2011. A Nokia N95 mobile device was given to all volunteers participating to the data collection campaign. The whole process of this campaign is explained in detail in [16]. In addition, the data was of course anonymized in order to preserve user privacy, i.e., the dataset does not allow to infer the identity of the users who participated in this data collection campaign. To summarize, the dataset contains 188 users. This data comes from different sources, such as locations from GPS or GPS WLAN, phone and SMS logs, accelerometer, application usage, etc. Although this data is rich and abundant, we only use the raw location data coming from GPS or GPS-WLAN sources.

2.7.2 Threat model and privacy indicator metric

As already pointed out, we only focus in this paper on *location privacy*. In particular, we do not distinguish cases where the user identity is already known by the location-based service and the latter wants to discover her mobility pattern, from scenarios where the user identity is not known and the location-based service is trying to discover it based on her locations. Yet controlling the access to locations by location-based services can have a significant impact on privacy as a whole in both cases.

We assume a threat model including a possible adversary represented by a location-based service, which is an untrusted component as depicted in Figure 2.2. This adversary wants to infer personal information related to a user based on her locations received from the operating system layer. In this case, we consider that the locations shared with the location-based service are critical as well as their accuracy. Their accuracy inevitably influences the process used by the location-based service to infer sensitive user data. Consequently, we propose a metric that is a privacy indicator aiming at highlighting the degree of alteration of the user locations sent to the location-based service. This alteration takes space and time dimensions. The privacy indicator enables to compute this level of alteration and takes values between 0 and 1 included. The higher the value of the privacy indicator, the higher the protection of the user locations sent. Conversely, the lower the value of the privacy indicator, the lower the protection of the user locations sent.

Remember that in our generic architecture (Figure 2.1), the purpose of the location access protocol is to alter raw locations of the form (ϕ, λ, t) in order to achieve location privacy. The nature of this alteration depends on the location access protocol and can act on both the *spatial dimension* of locations, i.e., ϕ and λ, and on their *temporal dimension*, i.e., t. To model this, we introduce function F as follows:

$$F(\langle loc_1, loc_2, \cdots \rangle) \mapsto \{(z_1, \mathit{\Delta} t_1), \cdots, (z_n, \mathit{\Delta} t_n)\}$$

where $z_i = (\phi_i, \lambda_i, \mathit{\Delta} r_i)$ represents the *spatial alteration* and $\mathit{\Delta} t_i$ represents the *temporal alteration* of the location information. It is important to note that these alterations are computed in parallel with the computation of the blurred locations directly sent to a possible adversary, which is the location-based service in our context. Both z_i and Δt_i are used to compute our global privacy indicator, as shown in Equation 2.6. That is, the global privacy indicator is the mean of the individual privacy indicators of all these alterations.

$$Privacy = \frac{1}{n} \times \sum_{i=1}^{n} Privacy(z_i, \Delta t_i) \tag{2.6}$$

Note that the number of $(z_i, \Delta t_i)$ tuples can significantly differ from the number of raw locations, e.g., in ShareZ, the privacy tree is the actual result of alteration function F. The calculation of the privacy of a single $(z_i, \Delta t_i)$ tuple is described in Equation 2.7. This second Equation is the sum of the spatial alteration and the temporal alteration where α and β are respectively factors of them and are complementary, i.e., $\beta = 1 - \alpha$.

$$Privacy(z_i, \Delta t_i) = \frac{(\alpha \times P_{space}(z_i)) + (\beta \times P_{time}(\Delta t_i))}{(\alpha + \beta)} \tag{2.7}$$

The spatial alteration is presented in Equation 2.8, where the minimum between the area of the zone and the maximum area, called z_{max}, is divided by this maximum area. It means that, when this maximum area is reached, the user cannot lose more privacy because her privacy is fully ensured.

$$\begin{aligned} P_{space}(z_i) &= \frac{min(Area(z_i), Area(z_{max}))}{Area(z_{max})} \\ &= \frac{min(z_i.\Delta r^2, z_{max}.\Delta r^2)}{z_{max}.\Delta r^2} \end{aligned} \tag{2.8}$$

The time alteration is presented in Equation 2.9, where Δt_{max} is the time threshold beyond which the user cannot gain more privacy. The equation is therefore the division of the minimum between $z_i.\Delta t$ and Δt_{max} by Δt_{max}.

$$P_{time}(\Delta t_i) = \frac{min(\Delta t_i, \Delta t_{max})}{\Delta t_{max}} \tag{2.9}$$

2.7.3 User mobility behavior classification

We choose to classify users of the Nokia dataset according to their mobility behavior in terms of distance in order to see if the latter might have an influence on the results obtained for the different experiments that are explained in the next section. Among the 188 users contained in the dataset, we started by selecting all the users having GPS and GPS-WLAN data and found 184 users. Then, we decided to compute the

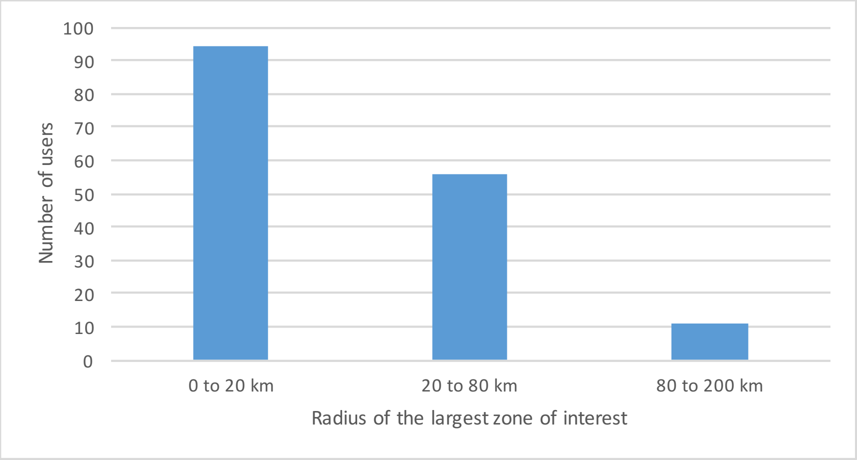


Fig. 2.5: User mobility behaviors

average of all mean time difference between two successive locations of all users in order to find an appropriate threshold. The latter was presented in the system model in Section 2.2 as Δt_{limit}. After computation, we find a mean of 568 seconds. Consequently, we decided to set the value of Δt_{limit} to 600 seconds, which indicates that the locations are captured in a regular manner. We select all users having a mean time difference lower than 600 seconds and obtain a final set of 161 users. The tracking duration of all these users varies from less than 1 day to 567 days.

After exploring the radius of the largest zone of interest of each remaining user, obtained with the privacy tree algorithm, we find three groups of users as described in Figure 2.5. In order to illustrate our classification, Figures 2.6, 2.7 and 2.8 describe the mobility traces of users belonging to each group. To be more precise, User 1 travels very long distances (i.e., a radius of the largest zone of interest of approximately 180 km), User 2 travels medium distances (i.e., a radius of approximately 60 km) and User 3 travels short distances (i.e., a radius of approximately 19 km) belonging to group 3, 2 and 1 respectively. In addition, Figures 2.9, 2.10 and 2.11 show the three different privacy trees obtained for each user. Concerning these three users, their locations were taken during a time period of about 500 days.

2.7.4 Scenarios and blurring strategies

We consider two scenarios involving two distinct location-based services. The first scenario focuses on a social network that offers various location-based discounts (e.g., travel discounts, shopping discounts, etc.) according to their space/time context. In this context, we also consider that users select one offer per month on average. This amount is called *utility*, which is the utility perceived by the user when she uses the service. The second scenario concerns a green location-based mobile application

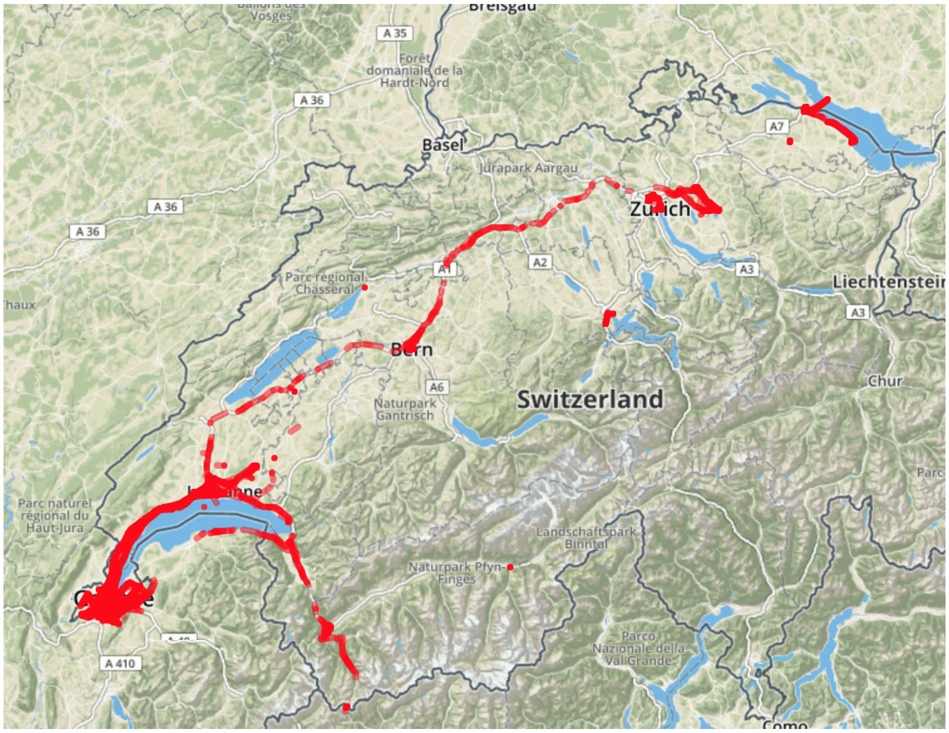


Fig. 2.6: Mobility traces of user 1

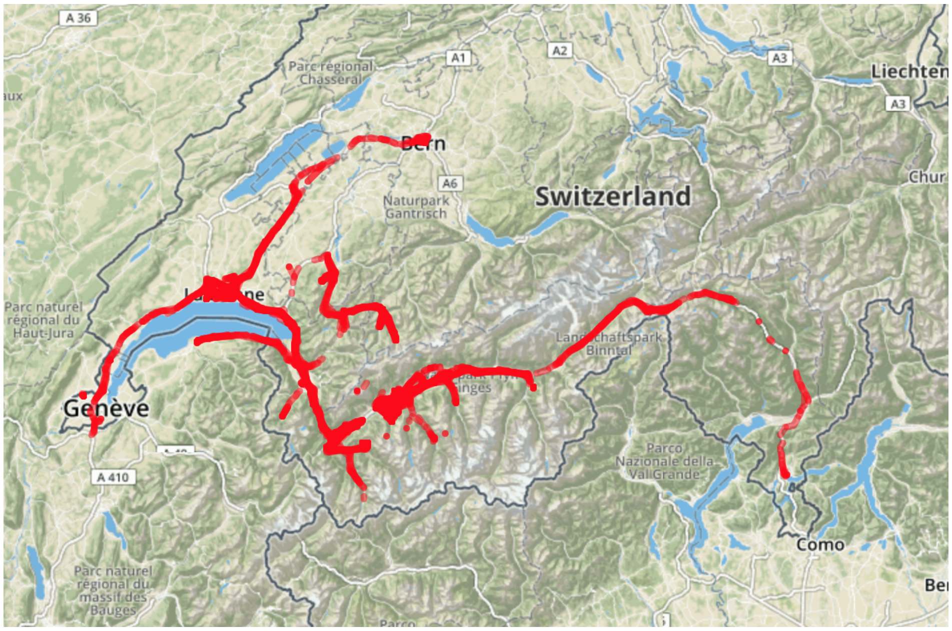


Fig. 2.7: Mobility traces of user 2

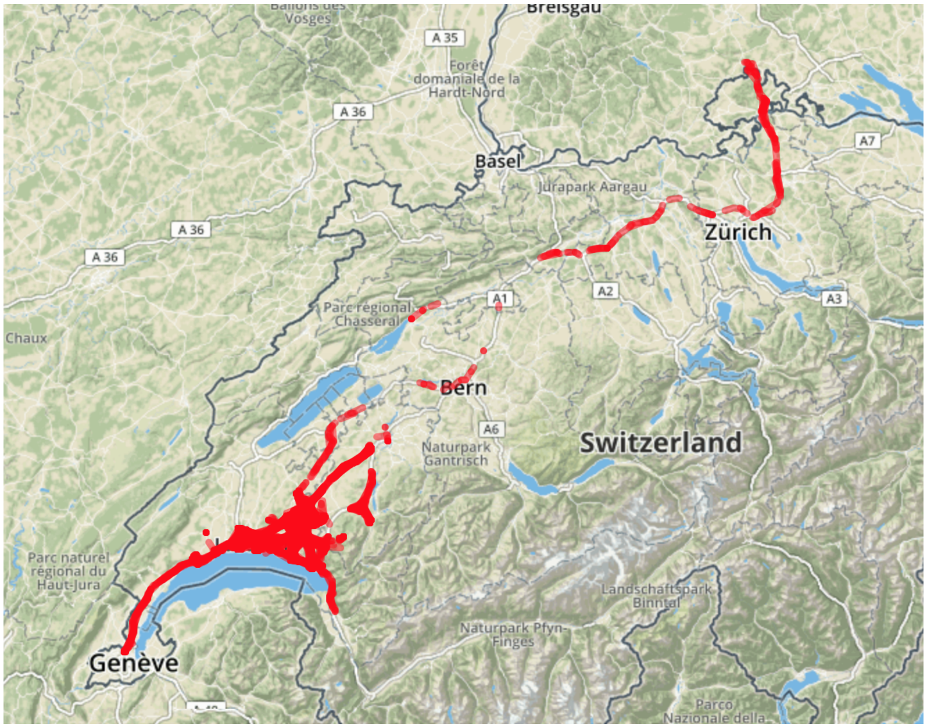


Fig. 2.8: Mobility traces of user 3

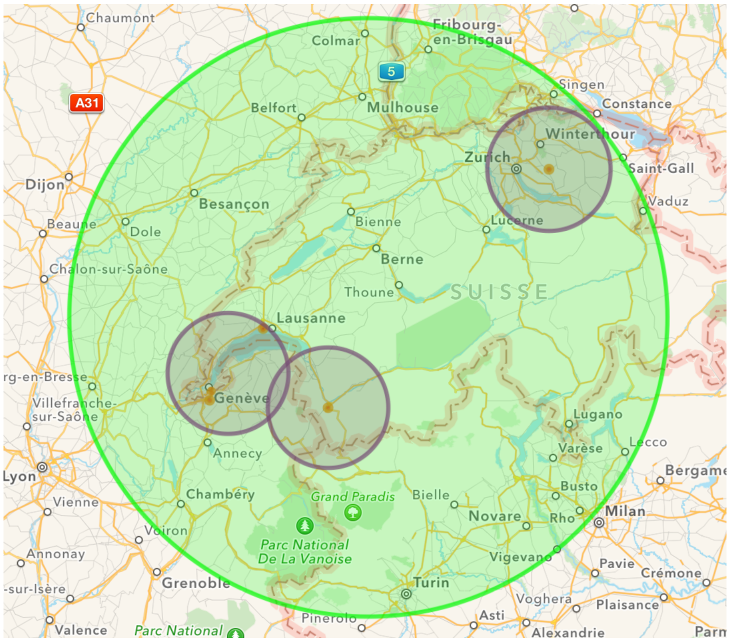


Fig. 2.9: Privacy tree of user 1 (7 zones of interest and 4 privacy tree levels)

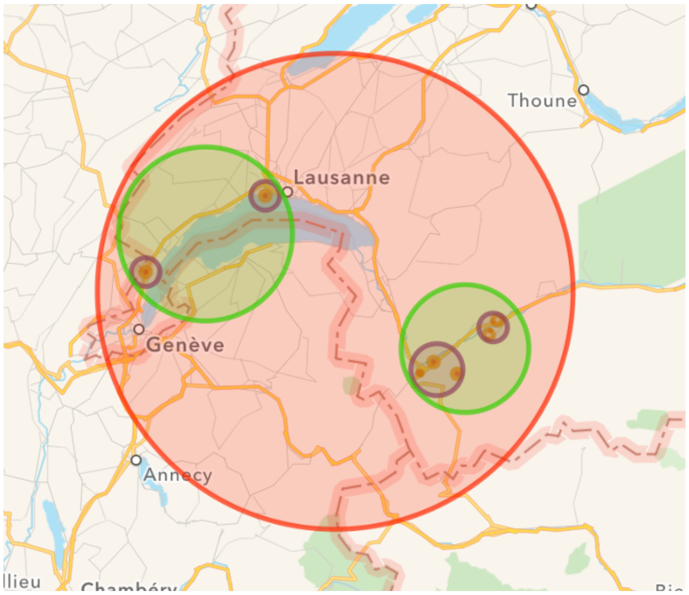


Fig. 2.10: Privacy tree of user 2 (10 zones of interest and 5 privacy tree levels)

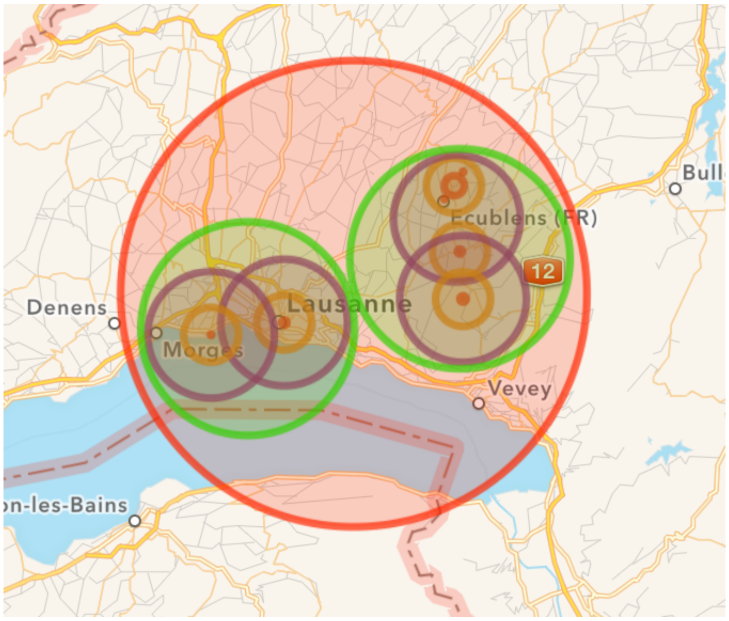


Fig. 2.11: Privacy tree of user 3 (7 zones of interest and 5 privacy tree levels)

that provides train and public transportation schedules to users according to their spatial and temporal context. The utility of this service is greater than the previous because users read the location-based information on an average of two times per day at least.

Regarding these two scenarios, we explore the impact of different blurring strategies applied to user locations. Furthermore, we also consider a case, i.e., worst case used as a reference, where all user locations are sent to the location-based service without any blurring strategy (i.e., spatial and temporal alterations are equal to 0 for all sent locations). The four selected blurring strategies are the following: Gaussian alteration, sampling, spatial cloaking and our privacy tree. The first three strategies are carried out in an offline buffered manner depending on the time window used to obfuscate the raw locations. Above all, it is important to note that the spatial cloaking technique is traditionally use to preserve the anonymity of users but we will use it as a blurring technique applied to one user as explained in Krumm's paper [14].

Gaussian alteration. The Gaussian alteration is a blurring strategy that consists in altering a location by adding Gaussian random noise on the latitude and the longitude of the original location. This Gaussian random noise depends on two variables: the mean which is the value where the curve is at the top (e.g., the latitude or the longitude of the original location) and the chosen standard deviation that may be expressed in meters. This strategy only has an impact on the space dimension of the location. In this context, each new location is spatially blurred and, therefore, a spatial alteration is generated. Regarding the spatial alteration, i.e., $z_i = (\phi_i, \lambda_i, \varDelta r_i)$, the centroid of the zone corresponds to the blurred location and its radius is the distance between the blurred location and the original location. However, the temporal alteration, i.e., $\varDelta t_i$, is 0 because there is no temporal alteration.

Sampling. The sampling strategy is a technique enabling to summarize several locations into a single location. In order to reach this goal, there exists different means of sampling. For our experiments, we decide to sample according to a time window. Consequently, we divide the user dataset into several sequences of locations. Each sequence has a duration equal to the time window. Then, we summarize locations of each sequence into a single location. We create a new location by computing the mean of all the latitudes and the mean of all the longitudes. In this context, spatial and temporal alterations are generated for each blurred location, which is computed from one sequence of locations. Regarding the spatial alteration, the centroid of the zone equals to the blurred location and its radius is the maximum distance between the centroid of the blurred location and the farthest original location included in the sequence used to compute the sample. In this context, the temporal alteration corresponds to the duration between the first location and the last location of the sequence used to compute the sample.

Spatial cloaking. As mentioned previously, spatial cloaking can be applied to one user only (see [14]) and, in our experiments, we use it as a blurring technique even if it is originally an anonymization technique. We assume that all zones of

Strategy	Spatial Alteration	Temporal Alteration
None (worst case)	No (real location sent)	No (real timestamp sent)
Gaussian alteration	Yes	No (real timestamp sent)
Sampling	Yes	Yes (time window)
Spatial cloaking	Yes/No	Yes/No
Privacy tree	Yes	Yes (no timestamp sent)

Table 2.1: Overview of alteration strategies

interest of a user, as precise as possible, are sensitive regions. The spatial cloaking works as follows: all user locations located in cloaked regions are deleted. In our implementation of the spatial cloaking technique, we simply create cloaked regions for each zone of interest of a user that must always contain the centroid of the related zone of interest. If the radius of the cloaked region is very small, the centroid of the cloaked region is obviously very close to the centroid of the zone of interest. Every time a user enters in a cloaked region, the original locations are deleted and, spatial and temporal alterations are created. Regarding the spatial alteration, the centroid of the zone and its radius are the centroid and the radius of the cloaked region respectively. The temporal alteration is the duration between the first location and the last successive location deleted for a specific cloaked region considering that the blurring process works in a sequential manner. For all remaining locations, which are not located in a cloaked region, spatial and temporal alterations are equal to 0 because they are sent without any alteration.

Privacy tree. Section 2.4 already explains the creation of a privacy tree as well as its different characteristics. It helps to blur user locations from the zones of interest of the user and an aggregation technique to build the privacy tree. In the analysis, we compute the alterations of all user's zones of interest of each selected level of the privacy tree (e.g. most precise, intermediate and less precise level explained below). Spatial and temporal alterations are generated for each zone of interest shared. Consequently, the spatial alteration corresponds to a zone having a centroid and a radius equal to the centroid and the radius of the zone of interest previously created. The temporal alteration is 1 (i.e., fully ensured) because no temporal information is revealed with the zones of interest.

Table 2.1 summarizes the chosen blurring strategies. In next section, the parameters of all blurring strategies are indicated.

2.7.5 Evaluation method, experimental settings and results

In this section, we present the experimental settings as well as the privacy indicator results achieved for each group of users.

2.7.5.1 Evaluation method and experiment settings

In order to run all the following experiments explained below, we create a cocoa application implemented in Objective-C containing the implementations of the four blurring strategies as well as the calculation of the privacy indicator. For all experiments, we chose a radius of z_{max} equal to 1000 meters under the assumption that it is a sufficient radius from which users consider that their location privacy is spatially ensured. This value is appropriate in highly dense urban areas but different radius should be chosen in other contexts, especially when users frequently travel. We also chose a Δt_{max} of the duration of one day meaning that when this threshold is exceeded, users think that their location privacy is temporally ensured. Consequently, when these two thresholds are exceeded, the location privacy is entirely preserved and the privacy indicator is equal to 1. Finally, for all the experiments we consider the factors α, i.e., spatial privacy factor, and β, i.e., temporal privacy factor, are equal (i.e., 0.5 for both factors).

The parameters of each blurring strategies are chosen in order to highlight when the privacy indicator results reach a maximum value, i.e., 0.5 or 1 depending on the strategy. Concerning the Gaussian alteration strategy, we select several standard deviations from 0.001 to 0.1 corresponding to different distance noise from approximately 120 to 11880 meters. Regarding the sampling strategy, we also take several different durations for the time window: from 30 minutes to 8 months. About the spatial cloaking strategy, we choose several different distances for the radius of the cloaked regions: from 1000 to 1000000 meters. And finally, regarding the privacy tree, three levels of privacy are taken into account: the most accurate, i.e., the level of the leaves of the tree (i.e., the zones of interest of a user), the intermediate level (i.e., approximately the mean between the most accurate level and the less accurate), and the level of the root node, which is the less accurate level. During an experiment, if a user has a privacy tree with only one level, we consider that the privacy indicator of the intermediate and that of the level of the root node also correspond to the level of the leaves. In addition, if a user has a privacy tree with two levels, the privacy indicator of the intermediate level is equal to that of the level of the leaves. User clusters are discovered with Δd_{max} of 60 meters and Δt_{min} of 900 seconds (i.e., 15 minutes). To highlight the zones of interest, we consider a $visitThreshold$ of 10. Finally, it is also important to mention that the zones of interest of the privacy tree are generated with the same parameters for all the users.

2.7.5.2 Results

The privacy indicator results obtained for each user group according to each blurring strategy are shown in Figures 2.12, 2.13, 2.14 and 2.15. The ”worst” scenario for which all locations are sent without any blurring strategy is not indicated in the diagrams because the privacy indicator is equal to 0. To begin, we can clearly see that the privacy indicator results are very close for all groups of users in Figures 2.12 and 2.13, unlike Figures 2.14 and 2.15. Regarding the spatial cloaking strategy for a radius of 100000 meters, the obtained privacy indicator results are different for each user group. This difference is explained by the fact that user groups have different travel distances. About the privacy tree, the privacy indicator results show indeed that they are different for the intermediate level. This difference can be easily explained: if a user travels long distances, the created privacy tree is automatically larger than that of a user travelling medium distances. Consequently, the alterations of the users travelling long distances are most significant than those of users moving over short distances. We observe the same if we compare the results obtained for medium and small distances of the intermediate level of the privacy tree.

In Figure 2.12, we can see that it is impossible to reach a privacy indicator result greater than 0.5 for a Gaussian alteration. This is explained by the fact that there is no temporal alteration when this strategy is used, unlike other blurring strategies. Figure 2.13 shows we can reach a suitable protection from a sampling with a time window duration of approximately 2 days. However, it seems to be rather unrealistic to send user locations every 2 days or more. Regarding the spatial cloaking strategy in Figure 2.14, the larger the radius of the cloaked regions, the higher the privacy indicator results. It also means that when 1 is reached, all user locations are contained in one large cloaked region. This strategy is interesting because it protects the sensitive regions of the user according to a specific radius distance around them. Concerning the privacy indicator results obtained for the privacy tree (see Figure 2.15), the most precise level (i.e., at the original zones of interest level of the users) already provides good results because they are around 0.5. These good results are explained by the non-disclosure of the temporal information. The results of the intermediate level strongly depend on the radius of the computed zones of interest of the intermediate level of the privacy tree. Then, the obtained results for the less precise level are very high, i.e., close to 1, because this level (i.e., at the root level of the privacy tree) mainly exceeds or is close to 1000 meters, which is the radius of the maximum area z_{max} of the space privacy (see Equation 2.8). Although the spatial cloaking strategy offers a good spatial and temporal protection, this strategy does not include a structure containing several encapsulated privacy levels. The privacy tree is a structure with various privacy levels, which makes it flexible. To conclude, the privacy tree strategy is an appropriate approach to protect location privacy of the user.

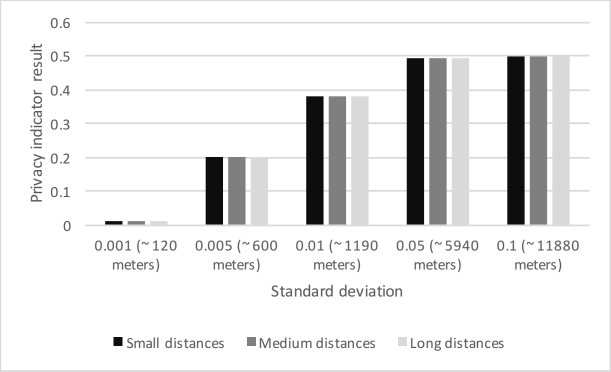


Fig. 2.12: Gaussian alteration - Privacy indicator results

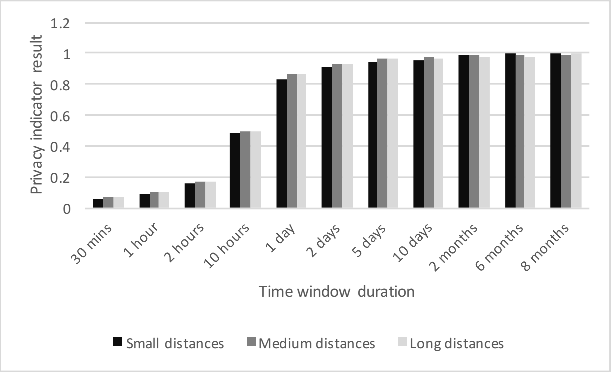


Fig. 2.13: Sampling - Privacy indicator results

According to these results, if we resume the two scenarios described at the beginning with the two location-based mobile applications, we can now argue that the privacy tree is an appropriate blurring strategy to ensure location privacy. More specifically, we lose less privacy by only allowing a certain level of accuracy/privacy especially when the utility of the application is not high (i.e., in the case of the social network). In addition, thanks to the privacy tree we can also decide to lose more privacy for the green application, which is more useful for us (i.e., high utility), compared to the social network that is considered as less useful (i.e., low utility).

2.8 Related work

In the following, we discuss two research subjects included in our work: *location protection strategies*, and *privacy architectures and frameworks*.

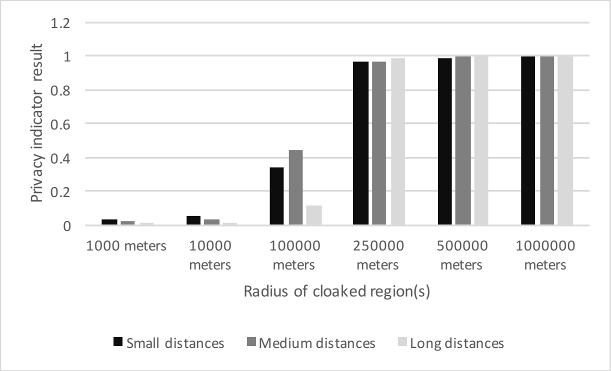


Fig. 2.14: Spatial cloaking - Privacy indicator results

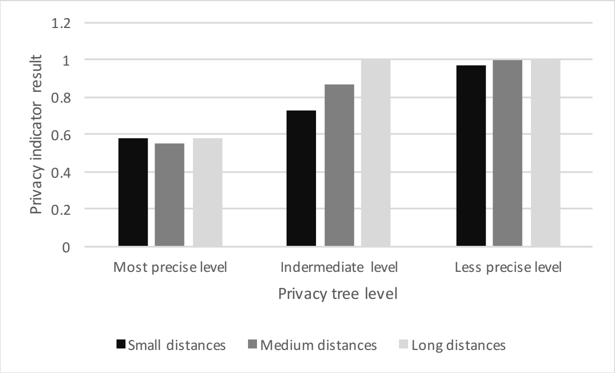


Fig. 2.15: Privacy tree - Privacy indicator results

2.8.1 Location protection strategies

There exists various strategies that enable to protect locations, such as mix-zones, spatial cloaking, perturbation, aggregation and many others as described in [8]. As presented in [8, 11, 14, 15], some of these strategies focus on anonymization (i.e., the impossibility of linking a user and an action) and others on location blurring or obfuscation (i.e., hiding a location). To begin with the anonymization strategies, Beresford and Stajano introduce the concept of *mix-zone* in [4] to achieve anonymization. In a mix-zone, precise locations of users are not computed. This guarantees the privacy of the user in this area and the anonymity of her moves from one zone to another zone thanks to a pseudonym mechanism. More specifically, when a user enters in a mix-zone, her pseudonym changes and the third party application does not obtain precise locations during her moves in the zone. Another technique helping to spatially anonymize a user is known as *spatial cloaking* [20]. If a user is located in a zone with at least k other users, the entire area is returned and not her precise location. This strategy is based on the k-anonymity concept. However, although these anonymization strategies are well-known, they should not be used alone but complemented by other blurring strategies [24].

Other strategies aim at blurring locations through alteration techniques, which are used to modify the location by adding some noise to the coordinates of the location, typically Gaussian noise. In [2], authors present various spatial approaches including affine transformations, random perturbation as well as aggregation. In [19], a family of geometric data transformation methods (GDTMs) are introduced. The aggregation strategy enables to gather some close locations into a single one. For example, the clustering process can help to reach this goal as seen in [8], where several clustering algorithms are described such as density-joinable cluster, density-time cluster and time-density cluster. In our work, a density-time is used in order to find clusters at the beginning of our process. In doing so, we rely on a blurring strategy, aggregation more specifically.

To end this section, there also exist other strategies consisting in sending fake user locations. Kido et al. introduce this concept by presenting an anonymous communication technique based on the sending of false locations with the true location of the user as demonstrated in [13]. In [5], Bindschaedler and Shokri present a model to generate fake user locations to protect user location privacy. Their model is based on statistical metrics quantifying geographic and semantic aspects of the mobility of users.

2.8.2 Privacy architectures and frameworks

There basically exists two types of frameworks and architectures aiming at protecting user privacy: *decentralized ones*, where components are confined to the mobile device, and *centralized ones*, where components may be the mobile device and on some remote server [11]. In [17], Mokbel et al. introduce a framework called *Casper*, which is entirely centralized. The goal of *Casper* is to allow users to use location-based services without disclosing their location data. This framework consists of two distinct components: a location anonymizer and a privacy-aware query processor. The location anonymizer constantly receives user locations from the mobile device and blurs them using a spatial cloaking technique. It is important to note that the location anonymizer is not on the mobile device but on a remote server. Then, the blurred location is sent to the privacy-aware query processor, which enables to handle location-based requests such as *"Where is the nearest restaurant?"*. This last component is also located on a remote server. In [18], Myles et al. present *LocServ*, a middleware acting as a unifying location service. This location service relies on a remote server that collects user locations from different sources, as well as user privacy requirements, and provides answers to location requests from applications. The main goal of *LocServ* is to protect user locations gathered by various tracking systems. Users must subscribe to the location server *LocServ* and indicate their privacy preferences expressed with rules. Unlike the previous middleware, user anonymity is achieved by using multiple identifiers for one user in *LocServ*. In [9], a decentralized architecture, called *Prive*, is presented. *Prive* enables to preserve the anonymity of users using a decentralized version of k-anonymity. Although the architecture of *Prive* is decentralized, there are a certification server where users need to be registered as well as a centralized pseudonym service. The goal of *Prive* is also to answer to requests like *"Where is the nearest hospital?"*, while protecting user anonymity. In [12], Hong and Landay present *Confab*, a framework for privacy-sensitive ubiquitous computing applications. Its goal is to facilitate the development of privacy-sensitive applications by taking into account user privacy preferences. The architecture is composed of infospaces linked to users or things, e.g., an infospace for a specific user and another for a room. An infospace contains static information (e.g., user name, user address, etc. . .) and dynamic information extracted from different sensors (e.g., temperature). The architecture may be centralized, i.e., an infospace is managed by a remote server, or decentralized, i.e., an infospace is hosted on the user mobile device. As presented in the paper with three use cases, locations may be shared during a specific duration (i.e., time-to-live flag) and according to a pre-defined location levels (i.e., street level, room level, city level). In [7], Fawaz and Shin present *LP-Guardian*, a framework that helps to ensure location privacy. *LP-Guardian* takes into account the tracking context: background or foreground mode. For instance, when locations are caught in background, *LP-Guardian* is able to capture the location object being in the process of creation and blur it in order to protect the user. On the contrary, when the tracking is in foreground, the user may be notified in order to choose an appropriate location sharing option, e.g., hiding the place, revealing it during the application session, as well as revealing it always. In this case, it is not a binary choice as in a standard architecture, described in Figure 2.1, but it is also not a personalized option provided according to the behavior of the user, because the offered choices are not generated on the basis of the user locations. In addition to the blurring option, anonymization may always be enabled or disabled according to the context. Amini et al. offer a system called *Caché*, described in [1], enabling to pre-fetch and store locally location-based content (i.e., weather forecast data, bus schedule data, etc. . .) from various sources (e.g., web services) and for multiple areas. These areas must be pre-defined by the user and are regions of interests of the user. Beresford et al. introduce *MockDroid* in [3], which is a modified version of Android operating system enabling to change personal content shared with an application. It is an interesting way to make users aware that a large amount of personal data is shared with applications or services. Users are able to find a adapted tradeoff between privacy and service functionality they want to use. Finally the last remaining research to present is *FINE*, a framework proposed by Shao et al. in [21]. This framework enables to mainly ensure the confidentiality of location-based service data and user location privacy, and provide a fine-grained access control in the location-based service system. A cloud server is mandatory because it executes location queries and is a bridge between the location-based service provider and the users. By using a specific and adapted encryption technique, the framework ensures the confidentiality of data shared. The user must decide the range of location she wants to query but there is no information regarding how this range is chosen by the user. The context of *FINE* is not the same as our work because the location-based service provider is considered as a remote entity and not as a component of the user mobile device.

When comparing the above architectures and frameworks to ShareZ, *LP-Guardian* is close to our solution in the sense that it also offers a personalized location privacy protection per application. However, its blurring strategy is quite different from ours, which relies on the concept of a privacy tree. Our blurring approach is constantly adapted to the user behavior in order to offer appropriate levels of location privacy protection for each specific location-based service. Furthermore, in the architectures presented above, most of them are centralized and need a remote server to work. Although *Confab* framework enables to share locations according to different location levels that seems to be pre-defined according to the context (e.g., different location levels in a building related to a work place). Our blurring solution is obviously not pre-defined because the privacy tree is updated over time. Unlike *Caché*, our solution automatically computes user's zones of interest in real-time from user raw location data. Moreover, we propose an approach per location-based service that the user wants to use, and not an approach that catches location-based content from different sources. To finish with *MockDroid*, even if this approach is per application, the privacy preferences are limited and also not automatically adapted to the user mobility behavior.

2.9 Conclusion and future work

In this paper, we proposed a novel system-level architecture providing fine-grained control over their location privacy to mobile device users. Our solution consists in a system-level architecture, which includes a service ShareZ, relying on a location access protocol that is strictly local. Our approach also offers a flexible blurring technique that is a privacy tree. Rather than forwarding all user locations to location-based services, ShareZ only shares her zones of interest, according to her privacy preferences. A quantitative location privacy indicator was also introduced and used in order to compare our solution to more traditional blurring approaches. The results indicate that the privacy tree, built from the user zones of interest, is a valuable structure to flexibly protect user privacy, as it enables to meet the trade-off between privacy and accuracy on a per-service and per-user basis. Future work could be focused on a real implementation of this architecture on a real mobile device in order to analyze several performance criteria and measure the tradeoff between location-based functionality and location privacy. Since the privacy tree is already implemented in Objective-C, we could implement the architecture on Apple iOS devices by modifying the operating system layer, which is the most challenging part of this possible future work.

Acknowledgment. This research is partially funded by the Swiss National Science Foundation in the context of Project 146714.

References

[1] Shahriyar Amini, Janne Lindqvist, Jason I. Hong, Maladau Mou, Rahul Raheja, Jialiu Lin, Norman M. Sadeh, and Eran Toch. Caché: caching location-enhanced content to improve user privacy. *Mobile Computing and Communications Review*, 14(3):19–21, 2010.

[2] Marc P. Armstrong, Gerard Rushton, and Dale L. Zimmerman. Geographically masking health data to preserve confidentiality. *Statistics in Medicine*, vol. 18:497–525, April 1999.

[3] Alastair R. Beresford, Andrew Rice, Nicholas Skehin, and Ripduman Sohan. Mockdroid: Trading privacy for application functionality on smartphones. In *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, HotMobile '11, pages 49–54, New York, NY, USA, 2011. ACM.

[4] Alastair R. Beresford and Frank Stajano. Location privacy in pervasive computing. *IEEE Pervasive Computing*, 2(1):46–55, 2003.

[5] Vincent Bindschaedler and Reza Shokri. Privacy through fake yet semantically real traces. *CoRR*, abs/1505.07499, 2015.

[6] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the Crowd: The privacy bounds of human mobility. *Scientific Reports*, 3, March 2013.

[7] Kassem Fawaz and Kang G. Shin. Location privacy protection for smartphone users. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, pages 239–250, New York, NY, USA, 2014. ACM.

[8] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núòez del Prado Cortez. Show me how you move and i will tell you who you are. *Trans. Data Privacy*, 4(2):103–126, August 2011.

[9] Gabriel Ghinita, Panos Kalnis, Spiros Skiadopoulus, Jaffar Joxan, Gabriel Ghinita, Panos Kalnis, and Spiros Skiadopoulos. Prive: Anonymous location-based queries in distributed mobile systems. In *In WWW '07: Proceedings of the 16th International Conference on World Wide Web*, pages 371–380. ACM Press, 2007.

[10] Antonin Guttman. R-trees: A dynamic index structure for spatial searching. *SIGMOD Rec.*, 14(2):47–57, June 1984.

[11] Adrian Holzer, Benoît Garbinato, and François Vessaz. Middleware for location privacy: an overview. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium*, RACS '12, pages 296–303, New York, NY, USA, 2012. ACM.

[12] Jason I. Hong and James A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *MobiSys'04*, pages 177–189. ACM, 2004.

[13] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. An anonymous communication technique using dummies for location-based services. In *Proceedings of the International Conference on Pervasive Services 2005, ICPS '05, Santorini, Greece, July 11-14, 2005*, pages 88–97, 2005.

[14] John Krumm. Inference attacks on location tracks. In Anthony LaMarca, Marc Langheinrich, and Khai N. Truong, editors, *Pervasive*, volume 4480 of *Lecture Notes in Computer Science*, pages 127–143. Springer, 2007.

[15] John Krumm. A survey of computational location privacy. *Personal Ubiquitous Comput.*, 13(6):391–399, August 2009.

[16] Juha K Laurila, Daniel Gatica-Perez, Imad Aad, Olivier Bornet, Trinh-Minh-Tri Do, Olivier Dousse, Julien Eberle, Markus Miettinen, et al. The mobile data challenge: Big data for mobile computing research. In *Pervasive Computing*, 2012.

[17] Mohamed F. Mokbel, Chi-Yin Chow, and Walid G. Aref. The new casper: query processing for location services without compromising privacy. In *VLDB '06*, pages 763–774, 2006.

[18] Ginger Myles, Adrian Friday, and Nigel Davies. Preserving privacy in environments with location-based applications. *IEEE Pervasive Computing*, 2(1):56–64, 2003.

[19] Stanley R. M. Oliveira and Osmar R. Zaïane. Privacy preserving clustering by data transformation. In Alberto H. F. Laender, editor, *SBBD*, pages 304–318. UFAM, 2003.

[20] Nayot Poolsappasit and Indrakshi Ray. Towards achieving personalized privacy for location-based services. *Trans. Data Privacy*, 2(1):77–99, April 2009.

[21] Jun Shao, Rongxing Lu, and Xiaodong Lin. Fine: A fine-grained privacy-preserving location-based service framework for mobile devices. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 244–252, April 2014.

[22] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of Predictability in Human Mobility. *Science*, 327(5968):1018–1021, February 2010.

[23] Yan Wang, Yingying Chen, Fan Ye, Jie Yang, and Hongbo Liu. Towards Understanding the Advertiser's Perspective of Smartphone User Privacy. In *Dis-*

tributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on, pages 288–297. IEEE, June 2015.

[24] Hui Zang and Jean C. Bolot. Anonymization of Location Data Does Not Work: A Large-Scale Measurement Study. In *Proc. of ACM Mobicom*, September 2011.

Chapter 3
A Location Privacy Estimator based on Spatio-temporal Location Uncertainties

Published in the proceedings of
The International Conference on NETworked sYStems (NETYS'17)
Marrakech, Morocco, May 17-19, 2017
Publisher: Springer

Abstract The proliferation of mobile devices and location-based services (LBS) is strongly challenging user privacy. Users disclose a large volume of sensitive information about themselves to LBS. Indeed, such services collect user locations to operate and can thus use them to perform various inference attacks. Several privacy mechanisms and metrics have been proposed in the literature to preserve location privacy and to quantify the level of privacy obtained when these mechanisms are applied on raw locations. Although the use of these metrics is relevant under specific threat models, they cannot anticipate the level of location privacy on the sole basis of the altered location data shared with LBS. Therefore, we propose a location privacy estimator that approximates the level of location privacy based on spatio-temporal uncertainties resulting from location alterations produced when a location privacy preserving mechanism is applied on user raw locations. This estimator also takes into account spatial-temporal user privacy parameters. We also describe the computation of the spatio-temporal uncertainties through the sampling, the Gaussian perturbation as well as the spatial cloaking. Finally, we compare the results of our estimator with those of the success of two localization attacks. The findings show that our estimator provides reasonable or conservative estimates of the location privacy level.

3.1 Introduction

Over the past few years, we have observed a privacy paradigm shift. Following the constant increase of mobile device users and location-based services (LBS), user sensitive data is not only shared with friends and acquaintances, but also with companies, which provide these services. However, users are not always aware of this privacy issue and they often do not have enough information to properly assess risks and benefits of the use of LBS [12]. We are in a privacy paradox as described in [3]. In this paper, Barnes discusses about privacy issues in a context involving teenagers and social networks. A user can reveal a lot of personal information about herself on a social network. This user obviously thinks that her data is adequately protected according to the privacy settings she chooses. She takes care about her privacy and does not want to disclose her private information to people she does not know on this social network. However, her personal data can be sold to third parties or explored for a variety of goals (e.g., profiling, targeted advertising) by the social network itself meaning that there is probably no privacy any more. Analyzing user information on social network is not the only way to obtain personal data about users. For instance, we can easily extract sensitive user information by exploring the metadata of user's photos shared online and performing attacks on them as demonstrated in [21]. There also exist other subtle ways to infer user's personal information, such as analyzing user's locations collected by a LBS. In the context of location privacy, Krumm [14] describes the main computational threats. According to these threats, various inference attacks can be performed by an adversary to reveal user sensitive information, such as home place, gender, tastes and much more as indicated in [8]. In order to deal with a privacy threat, user locations must be protected by using an adapted location privacy preserving mechanism. These mechanisms belong to different types of location alterations, such as location obfuscation, location perturbation, location confusion and location suppression as described in detail in [6]. A large number of location privacy metrics, presented in the literature, can accurately evaluate the level of protection provided by different location privacy preserving mechanisms by taking into account precise threat models as well as specific inference attacks performed by an adversary. To the best of our knowledge, there is no metric that can estimate the level of location privacy on the sole basis of the altered locations sent to a possible adversary. In addition, existing metrics do not take into account spatial and temporal privacy choice of the user.

To address this issue, we propose a spatio-temporal estimator enabling to approximate the level of location privacy and that only takes as input spatial and temporal uncertainties generated when a location privacy preserving mechanism is applied on the raw data in order to alter and protect them. Moreover, the estimator is user-oriented because it takes into consideration privacy choice of the user as parameters or could automatically define them by exploring the mobility behavior of the user. We consider that extracting spatio-temporal uncertainties from location alterations is crucial to estimate the level of location privacy. In the context of a stream of locations, most of location privacy preserving mechanisms, which modify the space dimension of a location, can also have an impact on the time dimension of the location(s) of this stream. Consequently, it is crucial to take into account these two types of uncertainties in the computation of a location privacy estimate. In order to properly evaluate this estimator, we compare its results and those of the performance of two localization attacks according to specific location privacy preserving mechanisms applied on user raw locations. This comparison enables to highlight if our privacy estimator can reasonably estimate the privacy level by only analyzing uncertainties resulting from the alterations produced after the application of a protection mechanism on raw data. For the experiments, we use a Nokia dataset containing real mobility traces of 185 users as precisely described in [16]. We choose three types of location privacy preserving mechanisms presented in [8] and [13]: the sampling, the Gaussian perturbation and the spatial cloaking. We also decide to evaluate the success of the two following localization attacks: the discovery of the most frequently visited places of a user (i.e., user's zones of interest) and the discovery of user's home place. This paper has three main contributions: presenting a new spatio-temporal location privacy estimator, describing the computation of the spatio-temporal uncertainties resulting from location alterations and evaluating our estimator with real user traces.

The paper is structured as follows. We start by describing the definitions and modeling of the main entities of our work in Section 3.2 providing the foundation to introduce the location privacy estimator in Section 3.3. Section 3.3.1 presents how spatial and temporal uncertainties are computed after applying three types of location privacy preserving mechanism on raw locations. In Section 3.4, we present in detail the evaluation as well as the obtained results in the context of localization attacks. Section 3.5 provides an overview of the existing location privacy metrics and highlights the links between these metrics and our estimator. Finally, we summarize the most relevant findings and present the future work in Section 3.6.

3.2 Definitions and modeling

This section describes the definitions and modeling of the main entities required to introduce the location privacy estimator.

3.2.1 User and raw locations

We consider a user who moves in a two dimensional space and owns a mobile device. This device enables to obtain raw locations via an embedded Global Positioning System (GPS) or a WiFi Positioning System (WPS) in order to locate itself. The history of the successive raw locations of the user stored in the device is a sequence $L = \langle loc_1, loc_2, \cdots, loc_n \rangle$, where $loc_i = (\phi, \lambda, t)$ is a 3-item tuple representing a unique location in which $\phi, \lambda \in \mathbb{R}$ are respectively a latitude and a longitude and $t \in \mathbb{N}$ is the time when the location was captured. We use the notation $loc.\phi$, $loc.\lambda$ and $loc.t$ to designate specific parts of loc below. In order to ensure that locations are mostly caught in a regular manner, the duration between two successive locations in L does not exceed a constant Δt_{limit}. Let loc_i and $loc_{i+1} \in L$, $loc_{i+1}.t - loc_i.t \leq \Delta t_{limit}$.

3.2.2 Location privacy preserving mechanism, altered locations and spatio-temporal location uncertainties

A location privacy preserving mechanism is applied on the raw locations of a user in order to protect them and to ensure her location privacy. There exist two main mechanisms to preserve user location privacy: anonymizing the identity of a user and altering the sequence containing user raw locations before sharing them with third-party entities such as LBS. In this paper, we only focus our attention on the second mechanism. Following this, we introduce a function called $protect(L)$ that modifies the sequence of user raw locations passed as a parameter by using a specific location privacy preserving mechanism. This function returns an altered sequence of locations that are sent to a third party entity, which can be seen as an untrusted component. This sequence is called $L_a = \langle loc_{a_1}, loc_{a_2}, \ldots, loc_{a_m} \rangle$. Applying a location privacy preserving mechanism on the sequence containing all raw locations L also generates spatial and temporal location uncertainties, which vary depending on the chosen mechanism. We define a zone z that describes a spatial location uncertainty, expressed by a 3-item tuple $z = (\phi, \lambda, \Delta r)$ where $\phi, \lambda \in \mathbb{R}$ represent a latitude and a longitude respectively and $\Delta r \in \mathbb{R}$ represents the radius of the zone. We also introduce $\Delta t \in \mathbb{N}$ that is a duration representing a temporal location uncertainty. Therefore, a spatio-temporal location uncertainty, called u, is a 3-item tuple including z and Δt as well as the number of raw locations of L affected by an alteration, simply called nb, such as $u = (z, \Delta t, nb)$. In the following sections, we sometimes use the notation $u.z$, $u.\Delta t$ and $u.nb$ to designate specific parts of u. Finally, we introduce a function called $uncertainties(L)$ that returns a sequence containing all the uncertainties computed when the location privacy preserving mechanism was applied on the sequence L. The output of this function is a sequence U containing all uncertainties such as $U = \langle u_1, u_2, \cdots, u_m \rangle$. It is important to note that the size of U can differ from the size of L_a and also the size of L. The computation of uncertainties only depends on the location privacy preserving mechanism applied on the sequence of raw locations and how the mechanism operates. The beginning of the next section will present how spatial and temporal uncertainties are computed in the context of three location privacy preserving mechanisms.

3.3 Location privacy estimator

This section describes both the computation of spatio-temporal uncertainties and the location privacy estimator. The first part is necessary to introduce the estimator, which takes as input the spatio-temporal uncertainties.

3.3.1 Computation of spatio-temporal uncertainties

There exist various location privacy preserving mechanisms that are described in the following papers [2, 7, 8]. Amongst them, we only consider the three following mechanisms such as the sampling, which will be performed in two ways, i.e., according to a time window (TW) and a specific number of locations (LN), the Gaussian perturbation as well as the spatial cloaking. All these mechanisms firstly affect the spatial dimension of a location and can also have an impact on the temporal dimension of a location or a subsequence of locations if we consider the context of a location stream. All new locations mentioned in the explanation below are obviously considered as the altered locations of the location sequence L_a sent to a LBS. During the use of a LBS, the first three strategies are carried out in an offline buffered manner depending on the time window used to obfuscate the raw locations.

3.3.1.1 TW sampling

The sampling according to a time window is a location privacy preserving mechanism enabling to summarize a subsequence of successive locations occurring during a specific period of time into a single new location. In a concrete implementation, we divide the entire user raw location sequence into several location subsequences and compute new locations according to them. The latitude of the new location is simply the mean of all latitudes of the original locations of the subsequence and, in the same manner, the longitude of this new location is the mean of all longitudes of the original locations. Concerning the timestamp of the new location, we consider that it corresponds to the timestamp of the location being in the middle of the subsequence of successive locations, i.e, median value. Consequently, this location privacy preserving mechanism generates both space and time alterations. We introduce $tw \in \mathbb{N}$ being the duration of the time window. We also consider a subsequence $L_{sub_i} \in L$ of successive raw locations such as $L_{sub_i} = \langle loc_1, loc_2, \ldots, loc_j \rangle$. This subsequence is a TW sampling subsequence iff the two following conditions are met:

- $loc_j.t - loc_1.t <= tw$
- $loc_{j+1}.t - loc_1.t > tw$

Then, the new location loc_{a_i} computed from the subsequence L_{sub_i} containing j raw locations is a tuple that includes the following elements:

- $loc_{a_i}.\phi = \frac{1}{j} \times \sum_{i=1}^{j} loc_i.\phi$
- $loc_{a_i}.\lambda = \frac{1}{j} \times \sum_{i=1}^{j} loc_i.\lambda$
- $loc_{a_i}.t = loc_{i/2}.t$

Concerning all spatio-temporal uncertainties u_i produced with the TW sampling, we generate an uncertainty for each new location. The centroid of the spatial alteration $u_i.z$ corresponds to the new location and its radius $u_i.z.\Delta r$ is the distance between the new location and the farthest raw location of the subsequence. The temporal alteration $u_i.\Delta t$ is the duration between the last and the first raw locations of the subsequence. The number $u_i.nb$ of raw locations affected by the mechanism is equal to the size of the subsequence L_{sub_i}. Let a function called $distance(loc_i, loc_j)$, which computes and returns the distance between the two locations passed as parameters. In addition, let a function called $farthest(loc_r, \langle loc_1, loc_2, \ldots, loc_n\rangle)$, which finds and returns the farthest location of the sequence of locations passed as a parameter from a given reference location loc_r. The several items below describe the previous explanations with the subsequence of raw locations L_{sub_i} introduced before.

- $u_i.z.\phi = loc_{a_i}.\phi$
- $u_i.z.\lambda = loc_{a_i}.\lambda$
- $u_i.z.\Delta r = distance(loc_{a_i}, farthest(loc_{a_i}, L_{sub_i}))$
- $u_i.\Delta t = loc_j.t - loc_1.t$

3.3.1.2 LN sampling

The sampling according to a number of successive locations is a location privacy preserving mechanism enabling to summarize a specific number of locations into a single new location. In a concrete implementation, we create several subsequences having the same number of successive locations and we summarize each subsequence into a single one new location. The latitude and the longitude of the new location is computed in the same manner as mentioned above for the TW sampling. The timestamp of this new location also corresponds to the timestamp of the raw location being in the middle of the subsequence of locations. Consequently, both space and time alterations are also generated when this mechanism is applied.

3.3.1.3 Gaussian perturbation

The Gaussian perturbation mechanism modifies each location of the sequence of the user raw locations by bringing spatial noise to its latitude and longitude. The latitude and the longitude of the raw location are changed according to two parameters: a mean and a standard deviation, which are the latitude or the longitude of the raw location and a value that may be expressed in meters respectively. Consequently, this location privacy preserving mechanism only affects the spatial dimension of the original location because the timestamp of each altered location remains unchanged. We introduce Δd_ϕ and $\Delta d_\lambda \in \mathbb{R}$ corresponding to two distances (i.e., spatial noise) randomly generated and added to the latitude and the longitude respectively in order to spatially blur the original location. Each location loc_i, contained in the sequence of user raw locations L, generates a new altered location that is sent to the untrusted component and noted loc_{a_i} as follows:

- $loc_{a_i}.\phi = loc_i.\phi + \Delta d_\phi$
- $loc_{a_i}.\lambda = loc_i.\lambda + \Delta d_\lambda$
- $loc_{a_i}.t = loc_i.t$

In this context, the spatial alteration is computed for each new location such as the centroid of the zone $u_i.z$ is loc_{a_i} and its radius is the distance between the loc_{a_i} and the raw location loc_i. The temporal alteration $u_i.\Delta t$ is equal to 0 because the timestamp of the new location remains the same. The number $u_i.nb$ is equal to 1 because the perturbation only affects one raw location. The spatial and temporal alterations generated are summarized below:

- $u_i.z.\phi = loc_{a_i}.\phi$
- $u_i.z.\lambda = loc_{a_i}.\lambda$
- $u_i.z.\Delta r = distance(loc_{a_i}, loc_i)$
- $u_i.\Delta t = loc_{a_i}.t - loc_{a_i}.t = 0$

3.3.1.4 Spatial cloaking

As presented in [13], Krumm introduces an implementation of the spatial cloaking algorithm that can be applied on a single user's location dataset. In Krumm's paper, an ambiguity is created around a sensitive location (i.e., user's home place in the paper) by computing a specific cloaked region containing the user's home place and deleting all user raw locations being recorded in this region in order to protect the privacy of the user. A cloaked region is a zone defined by a centroid and a radius. The sensitive location is not the center of the computed cloaked region but it is only contained in this region. Consequently, it is more difficult to find the original sensitive location for an adversary in this case. In our implementation, we do the same around all zones of interest found for a user, which are obviously her sensitive areas. By applying this location privacy preserving mechanism on user raw data, we only delete sensitive locations occurring in a cloaked region without altering the previous or successive raw locations that are not located in cloaked regions. Consequently, all locations being in the sequence L_a of altered locations and sent to a LBS will be raw, such as $loc_{a_i} = loc_i$.

Spatial and temporal uncertainties are generated for the deletion of the raw locations being in the cloaked regions. Let C the set containing k computed cloaked regions and $C[i]$ a cloaked region in which the raw locations of the subsequence $L_{sub_i} = \langle loc_1, loc_2, \cdots, loc_j \rangle$ are located. Raw locations are sent to a LBS, consequently, we also compute uncertainties for these raw locations. In the following description, (1) describes the uncertainty of the deletion of raw locations and (2) presents the uncertainty of a raw location. The number $u_i.nb$ of the first uncertainty is equal to the number of raw locations deleted and contained in L_{sub_i} while the number of the second uncertainty corresponds to 1.

- (1) $\langle loc_1, loc_2, \cdots, loc_j \rangle \in C[i]$
  - $u_i.z.\phi = C[i].\phi$
  - $u_i.z.\lambda = C[i].\lambda$
  - $u_i.z.\Delta r = C[i].\Delta r$
  - $u_i.\Delta t = loc_j.t - loc_1.t$
- (2) $loc_i \notin C[i]$
  - $u_i.z.\phi = loc_{a_i}.\phi$
  - $u_i.z.\lambda = loc_{a_i}.\lambda$
  - $u_i.z.\Delta r = distance(loc_{a_i}, loc_{a_i})$
  - $u_i.\Delta t = loc_{a_i}.t - loc_{a_i}.t$

3.3.2 Estimator

The location privacy estimator takes as input all the uncertainties obtained when a location privacy preserving mechanism is applied on the sequence of raw locations L and generates the altered sequence of locations L_a as described in Section 3.2 and 3.3.1. Since an uncertainty has spatial and temporal dimensions, the estimator includes the privacy evaluations of these two dimensions. A top-down approach is chosen to present the estimator. As detailed in Equation 3.1, the final result of this estimator is the sum of each location privacy estimate $Privacy(u_i)$ related to each spatio-temporal uncertainty u_i contained in the sequence U multiplied by the number of raw locations affected by an alteration. Finally, the sum is divided by the total number of raw locations n of L.

$$Privacy_e = \frac{1}{n} \times \sum_{i=1}^{n} (Privacy(u_i) \times u_i.nb) \tag{3.1}$$

The computation of the estimate of the location privacy of a single spatio-temporal uncertainty $Privacy(u_i)$ is described in Equation 3.2. This second equation is the sum of spatial and temporal location privacy estimates of the uncertainty multiplied by their respective factor (i.e., α for the spatial location uncertainty and β for the temporal location uncertainty) and finally divided by the sum of these two factors in order to normalize the final result. These two factors must be chosen according to the importance of the spatial and the temporal dimensions for the user. If a user considers that her spatial privacy is more important than her temporal privacy, α could have more weight than the temporal factor β, knowing that β must be always equal to $1 - \alpha$.

$$Privacy(u_i) = \frac{(\alpha \times P_{space}(u_i.z)) + (\beta \times P_{time}(u_i.\Delta t))}{\alpha + \beta} \tag{3.2}$$

The location privacy estimate of the spatial uncertainty $P_{space}(u_i.z)$ is presented in Equation 3.3, where the minimum between the area of the zone $u_i.z$ and the area of a zone that we consider as a maximum area called z_{max} is divided by this maximum area z_{max}. It means that, when this area is reached, the user cannot lose more privacy because her privacy is fully ensured when this maximum area is reached.

$$\begin{aligned} P_{space}(u_i.z) &= \frac{min(Area(u_i.z), Area(z_{max}))}{Area(z_{max})} \\ &= \frac{min(u_i.z.\Delta r^2, z_{max}.\Delta r^2)}{z_{max}.\Delta r^2} \end{aligned} \tag{3.3}$$

The location privacy estimate of the temporal uncertainty $P_{time}(u_i.\Delta t)$ is presented in Equation 3.4, where Δt_{max} is a time threshold beyond which the user cannot lose more privacy. The equation is therefore the division of the minimum between $u_i.\Delta t$ and Δt_{max} by Δt_{max}.

$$P_{time}(u_i.\Delta t) = \frac{min(u_i.\Delta t, \Delta t_{max})}{\Delta t_{max}} \tag{3.4}$$

The two values, z_{max} and Δt_{max}, should also be chosen by the user who is able to know when she considers that the spatial and temporal dimensions of her privacy are considered as entirely ensured. These two values could also be automatically determined by studying the mobility behavior of the user.

3.4 Experiments and results

In this section, we present the chosen approach to evaluate the reliability of the location privacy estimator in the context of localization attacks. To reach this goal, we observe the correlation between the evolution of the privacy level predicted with the estimator and the evolution of the success of the chosen localization attacks. In the next sections, we first present the dataset used for the experiments, the localization attacks as well as our findings at the end.

3.4.1 Dataset

We select a dataset provided by Nokia that contains real mobility data traces collected in Switzerland (Europe) from October 2009 to March 2011. The collecting process of this campaign is explained in detail in [16]. This dataset consists of real data traces of 185 users including GPS location data, GPS WLAN location data, SMS, calls and several other data. Since the duration of the data collection varies from one user to another, i.e., from less than one day to more than 500 days, we decide to only retain 103 users of this dataset who met the following conditions. A user must have a sequence of raw locations captured during a period of at least 300 days and a Δt_{limit} (defined in Section 3.2) of 600 seconds on an average meaning that the locations are captured in a frequent manner.

3.4.2 Localization attacks

From an adversary viewpoint, an inference attack aims at discovering sensitive information based on user locations. In our context, the adversary is the untrusted component of the mobile device, i.e., a LBS. The data, which is used as input to perform the attack, is the user locations sent to the LBS after applying a location privacy preserving mechanism on raw data. Various threats and inference attacks are presented in [8, 13, 14, 20]. We select two different localization attacks [20] having different goals: discovering zones of interest of a user and discovering user's home place. We describe in detail their goal, the way they operate as well as the quantification of their success in the next sections.

3.4.2.1 Discovering user's zones of interest

This first localization attack is performed by an adversary that wants to highlight all the most frequently visited places, i.e., zones of interest, of a user based on her locations sent from the trusted component, i.e., operating system, to the untrusted component, i.e., LBS. The zone of interest discovery process is entirely based on an algorithm described in this paper [15], with the sole exception that the recent aspect of a zone is not taken into account. In order to calculate the success of this attack, we first compute the reference set of user's zones of interest that is obtained when this attack is performed on user's raw data. Consequently, we decide to quantify the success of this attack as the discovery area percentage of the reference set by comparing it to that obtained with the altered sequence of locations. Firstly, we consider a set Z_r containing all reference zones of interest of the user obtained with the sequence L. Secondly, we introduce Z_b that is a set containing all user's zones of interest computed from the sequence L_a. Finally, we compare Z_r and Z_b to evaluate the success of this inference attack. More specifically, we compute the discovery area mean of all discovery areas linked to the reference zones of interest of the user as described in Equation 3.5. In this Equation, we consider that Z_r contains n reference zones of interest such as $Z_r = \{z_i, z_{i+1}, \cdots, z_n\}$ and Z_b has m zones of interest such as $Z_b = \{z_j, z_{j+1}, \cdots, z_m\}$. We simply use z_i and z_j below to refer a zone of Z_r and a zone of Z_b respectively. The result of the success of this attack is a value between 0 (i.e., no zone is discovered) and 1 (i.e., all zones are discovered) included.

$$success_{IA}(Z_r, Z_b) = \frac{1}{n} \times \sum_{i=1}^{n} discoveredAreaSum(z_i, Z_b) \tag{3.5}$$

Equation 3.6 enables to compute the sum of all discovered areas of a specific reference zone of interest z_i by comparing it with all possible zones of interest of Z_b.

$$discoveredAreaSum(z_i, Z_b) = \sum_{j=1}^{m} discoveredArea(z_i, z_j) \tag{3.6}$$

Then, we compute the discovered area percentage of a reference zone, also scaled from 0 to 1, in Equation 3.7. For this computation, we take into account if there exists a discovered area between a reference zone of interest and a zone of interest of Z_b. Four cases are taken into consideration in order to compute this discovered area percentage. Firstly, if there is no intersection or inclusion between the reference zone and one of the zones of Z_b, the discovered area percentage equals 0. Secondly, in the case where there exists an intersection between the reference zone and one of the zones of Z_b, the discovered area percentage is equal to the area of the intersection divided by the area of the reference zone. Thirdly, in the case where one of the zones of Z_b is fully included in the reference zone, we also compute the discovered area percentage as mentioned previously. And fourthly, if the reference zone is fully included in one of the zones of Z_b, the discovered area percentage corresponds to the area of the intersection divided by the area of the zone of Z_b because the discovery precision is reduced. Since the zone of interest discovery algorithm includes a merging of clusters before the zone of interest discovery, there is no overlap amongst all discovered user's zones of interest. Considering this, we do not need to manage cases where a same specific area of a reference zone is covered by two zones of Z_b. Hence, we can summarize these four cases in two cases only as detailed in the Equation 3.7 below.

$$discoveredArea(z_i, z_j) = \begin{cases} Area(z_i)/Area(z_j), & \text{if } z_i \subset z_j \\ Area(z_i \cap z_j)/Area(z_i), & \text{otherwise} \end{cases} \tag{3.7}$$

3.4.2.2 Discovering user's home place

For this localization attack, an adversary wants to discover the user's home place. Two techniques are used to perform this attack. The first technique is based on the discovery of user's zones of interest with the process explained in the previous section, while the second technique focuses on one heuristic. Regarding the first technique, we use the user's zones of interest because the user's home place is obviously one of them. We start by computing the set containing all user's zones of interest, then we search the home place amongst all of them by highlighting the most likely visited zone of interest of the set during a specific time slice, i.e., from 8:00 PM to 6:00 AM the next day. The second technique is inspired by a heuristic called *last destination*, which is described in [13]. *Last destination* heuristic consists in computing the last destination visited by a user, i.e., at the end of the day. In our implementation, this place is discovered during the time slice starting at 0:00 AM and ending at 4:00 AM. The output of the second technique used is also a zone defined by a centroid and a radius. In order to evaluate these two inference attacks related to the user's home place, we compute a reference user's home place zone z_i with the user raw locations L. Similarly, we also extract z_j, which is the user's home place computed from the altered sequence of locations L_a. We compute the success of these inference attacks by using Equation 3.7 in order to evaluate the discovery percentage between the reference zone z_i and the other computed zone z_j.

3.4.3 Experimental settings and results

In the previous sections, we presented all the key elements used for the experiments. We now describe the experimental settings as well as the main findings.

3.4.3.1 Experimental settings

Regarding the location privacy estimator, we first choose equal factors for the computation of the privacy of the spatial and temporal uncertainty meaning that α and β have the same weight for all users, i.e., both are equal to 0.5. Secondly, we consider that the radius of z_{max} equals a value of 1000 meters indicating that the user considers that her spatial privacy is fully ensured when this radius is reached or exceeded. This value is determined according to the dataset because users mainly move in cities or areas where a sufficient number of individuals are living. In addition, we select a value of 24 hours for Δt_{max}, also meaning that her temporal privacy is entirely ensured if this duration is exceeded. Thirdly, several parameters are selected for each location privacy preserving mechanism. Regarding the TW sampling, we select 68 values for the time window ranging from 5 minutes to 5 days. About the LN sampling, we choose 24 values ranging from 10 locations to 2500 locations per sample. Concerning the Gaussian alteration, 16 standard deviation values are selected ranging from 0.0001 (i.e., about 12 meters) to 0.05 (i.e., 5948 meters approximately). And finally, for the spatial cloaking, we take 6 values ranging from 100 meters to 10000 meters for the radius of the cloaked region. Regarding the localization attacks, we select the following parameters for the user's zone of interest discovery: Δd_{max} equals 60 meters and Δt_{min} is 900 seconds (i.e., 15 minutes). A value of 6 visits is chosen for the *visitThreshold* in order to highlight the frequent user's zones of interest. We found these values by exploring the Nokia dataset. The time slices used for each inference attacks are described in the previous section. To conclude, the chosen location privacy preserving mechanisms are those described in Section 3.3.1.

3.4.3.2 Results

For each selected parameter of each location privacy preserving mechanism, we first compute the mean of the success results of all users for each localization attack. Then, we display the evolution of the privacy level according to the evolution of the mean of the success results of all localization attacks in Figure 3.1 and Figure 3.2. In the four graphs, each dot corresponds to a specific parameter that evolves from the lowest value to the highest value of the range of the parameters of each protection mechanism described in the previous section. These graphs allow us to see the evolution trend between the results of our location privacy estimator and the success of the attacks. Regarding the Gaussian perturbation in Figure 3.2 on the left, the maximum result given by the location privacy estimator is 0.5 because the temporal uncertainty is equal to 0. The best results are obtained with the TW sampling and the LR sampling because the curves show a negative linear correlation between the success of the attacks and the privacy estimator results. Regarding the Gaussian perturbation and the spatial cloaking, we observe exponential decay curves meaning that our privacy estimator is pessimistic. In the context of Gaussian perturbation, the performance of the localization attacks declines relatively quickly because the added spatial noise has a high impact on the detection of the zone's of interest. In the context of spatial cloaking, the performance of the localization attacks declines sharply because all zone's of interest are considered as cloaked regions. Even if Gaussian perturbation and spatial cloaking results present less accurate estimates than those of TW and LR sampling, the obtained privacy estimates are conservative in that they do not give a false sense of location privacy. This means that there is no outlying curve (i.e., exponential curve) or outlier results such as a high success probability of the localization attacks and a high location privacy estimate at the same time.

3.5 Related work

Estimating location privacy is the central aspect of this paper. Therefore, this related work presents a classification of existing privacy metrics found in the literature and the possible links between them and our estimator at the end.

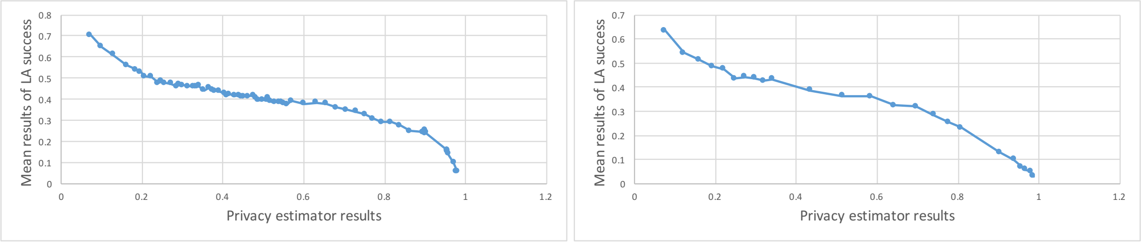


Fig. 3.1: TW (left) and LR (right) sampling results.

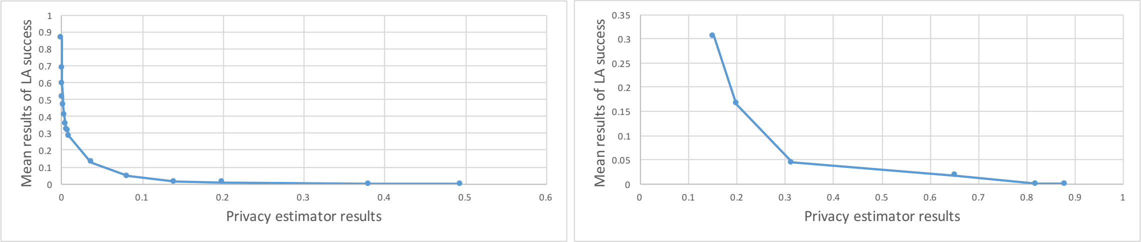


Fig. 3.2: Gaussian perturbation (left) and spatial cloaking (right) results.

3.5.1 Error-based metrics

To begin with the first category, Hoh and Gruteser use two main location privacy metrics to evaluate a path perturbation algorithm they propose in [9]. The first location privacy metric, called *mean location privacy*, computes the accuracy of the estimation of each location contained in a user's path by an adversary (i.e., the expectation of distance error). It takes into account the difference between correct and estimate locations as well as the probability of occurence of the estimate location. In addition to this first metric, they also consider a second metric, called *mean location error* aiming at evaluating the quality of service provided. Basically, this metric helps to compute the location accuracy difference of each user's paths (i.e., between the original and the observed location). In [19], Shokri et al. introduce a new location privacy metric, called *distortion-based metric*. It evaluates the level of distortion of a reconstructed trace of a user. The latter is obtained by applying the reverse of the location preserving mechanism used to generate the observed trace. The metric takes into account the probability of each possible reconstructed trace as well as the sensitivity of the locations in terms of space and time because it may directly have an influence on the user privacy. Shokri et al. also introduce a framework for the analysis of location privacy preserving mechanisms including a metric to evaluate the user's location privacy in [20]. This metric, called *correctness*, enables to quantify the correctness of the attack by computing the expected estimation error of the distance between the true expected result and all the results contained in the estimate distribution, which is the output of an attack. Moreover, other error-based metrics are presented in [6, 17, 18].

3.5.2 Uncertainty-based metrics

In [4], Beresford and Stajano use a metric computing the level of anonymity ensured by a mix-zone by using the entropy. To summarize, an adversary is confronted with a mapping issue including old and new pseudonyms taken by users in a mix-zone. The entropy enables to compute the level of uncertainty in the mapping set and quantify the number of users that we are not able to distinguish from each other. Hoh et al. also introduce a privacy level measure called *mean time to confusion* in [11]. It highlights the tracking time from which an adversary is no longer able to find the next location sample with a sufficient certainty. In [5], Cheng et al. present a framework enabling to control the location uncertainty aiming at preserving user privacy. They also build a model and queries helping to reach this goal and introduce two means of quantifying privacy. The first is the size of uncertainty region due to the fact that the larger the region size, the higher the privacy. The second is based on the location of the user and its link with sensitive regions such as an hospital or other sensitive places related to the user. More formally, it computes the ratio between the area of sensitive regions discovered (i.e., the intersection between the sensitive regions and the area of the uncertainty of a location) and the area of the uncertainty of a location. The higher the ratio, the lower the privacy. Finally, Ardagna et al. present a metric called *relevance* that represents the relative accuracy loss of the location when a location obfuscation is applied on a raw location in [1, 6]. This metric only takes into account the geometric uncertainty generated by a location obfuscation mechanism without defining the adversary's goal and knowledge.

3.5.3 Score-based metrics

Hoh et al. evaluate the degree of privacy protection with two metrics in the context of traffic monitoring in [10]. They consider that an adversary could try to infer the user's home and they evaluate the privacy obtained for different sampling frequencies. The first metric focuses on the effectiveness of the detection by computing the home identification rate (i.e., the number of correct estimated homes out of the total number of correct homes) and the second metric computes the false positive (i.e., the number of incorrect estimated homes out of the total number of estimated homes). In [8], Gambs et al. evaluate the impact of different sanitization mechanisms on different means aiming at reaching the same adversary's attack. More specifically, the attack relates on detecting the user's clusters. In order to evaluate this impact, they use well-known metrics called precision and recall. In their analysis, the precision is the number of correct points of interest divided by the total number of points of interest returned by an attack and the recall is the number of area detected divided by the total number of areas.

To finish, there also exist other metrics such as *k-anonymity* and differential privacy-based metrics as discussed in [6]. In this classification, our location privacy estimator could clearly belong to the *uncertainty-based* category. The existing metrics mainly take into account the spatial dimension to compute the privacy while the temporal dimension is equally important and can also be affected by a location privacy preserving mechanism. To the best of our knowledge, there is no user-oriented location privacy metric using spatio-temporal uncertainties resulting from spatial and temporal alterations applied on user raw locations.

3.6 Conclusion and future work

In this paper, we have presented a location privacy estimator taking into account spatial and temporal uncertainties, generated when a location privacy preserving mechanism is applied on user raw data, as well as user privacy preferences. We also introduce how to generate spatial and temporal uncertainties according to three existing privacy mechanisms. We chose to evaluate it by comparing the results of our estimator and those of the success of two localization attacks. This comparison showed that our estimator provides reasonable or conservative estimates of the location privacy level. Future work could focus on implementing other location privacy preserving mechanisms and other localization attacks in order to have a better overview of the behavior of the estimator. Then, we could also try to automatically adapt z_{max} and Δt_{max} to the mobility behavior of each user. Another interesting work could be to add a weight to each uncertainty according to the degree of importance of the raw location(s) affected by an alteration in order to see if it increases the accuracy of the results of the estimator. And finally, a last challenge could be to adapt the computation of the location privacy with our estimator in realtime during the use of a LBS on a mobile device.

Acknowledgment. This research is partially funded by the Swiss National Science Foundation in the context of Project 146714.

References

[1] Claudio Agostino Ardagna, Marco Cremonini, Ernesto Damiani, S De Capitani Di Vimercati, and Pierangela Samarati. *Location Privacy Protection Through Obfuscation-Based Techniques*, pages 47–60. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[2] Marc P. Armstrong, Gerard Rushton, and Dale L. Zimmerman. Geographically masking health data to preserve confidentiality. *Statistics in Medicine*, vol. 18:497–525, April 1999.

[3] Susan B. Barnes. A privacy paradox: Social networking in the united states. *First Monday*, 11(9), 2006.

[4] Alastair R. Beresford and Frank Stajano. Mix zones: user privacy in location-aware services. In *Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second IEEE Annual Conference on*, pages 127–131, March 2004.

[5] Reynold Cheng, Yu Zhang, Elisa Bertino, and Sunil Prabhakar. Preserving user location privacy in mobile data management infrastructures. In George Danezis and Philippe Golle, editors, *Privacy Enhancing Technologies*, volume 4258 of *Lecture Notes in Computer Science*, pages 393–412. Springer, 2006.

[6] Maria Luisa Damiani. Location privacy models in mobile applications: conceptual view and research directions. *GeoInformatica*, 18(4):819–842, 2014.

[7] Matt Duckham and Lars Kulik. Location privacy and location-aware computing. *Dynamic & mobile GIS: Investigating change in space and time*, pages 34–51, 2006.

[8] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. Show me how you move and i will tell you who you are. *Trans. Data Privacy*, 4(2):103–126, August 2011.

[9] Baik Hoh and Marco Gruteser. Protecting location privacy through path confusion. In *SecureComm*, pages 194–205. IEEE, 2005.

[10] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. Enhancing security and privacy in traffic-monitoring systems. *IEEE Pervasive Computing*, 5(4):38–46, 2006.

[11] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. Preserving privacy in gps traces via uncertainty-aware path cloaking. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS '07, pages 161–171, New York, NY, USA, 2007. ACM.

[12] Flavius Kehr, Tobias Kowatsch, Daniel Wentzel, and Elgar Fleisch. Thinking styles and privacy decisions: Need for cognition, faith into intuition, and the privacy calculus. In *Smart Enterprise Engineering: 12. Internationale Tagung Wirtschaftsinformatik, WI 2015, Osnabrück, Germany, March 4-6, 2015.*, pages 1071–1084, 2015.

[13] John Krumm. Inference attacks on location tracks. In *Proceedings of the 5th International Conference on Pervasive Computing*, PERVASIVE'07, pages 127–143, Berlin, Heidelberg, 2007. Springer-Verlag.

[14] John Krumm. A survey of computational location privacy. *Personal Ubiquitous Comput.*, 13(6):391–399, August 2009.

[15] Vaibhav Kulkarni, Arielle Moro, and Benoît Garbinato. A mobility prediction system leveraging realtime location data streams: Poster. In *Proceedings of the*

22Nd Annual International Conference on Mobile Computing and Networking, MobiCom '16, pages 430–432, New York, NY, USA, 2016. ACM.

[16] Juha K Laurila, Daniel Gatica-Perez, Imad Aad, Olivier Bornet, Trinh-Minh-Tri Do, Olivier Dousse, Julien Eberle, Markus Miettinen, et al. The mobile data challenge: Big data for mobile computing research. In *Pervasive Computing*, 2012.

[17] David Rebollo-Monedero, Javier Parra-Arnau, Claudia Díaz, and Jordi Forné. On the measurement of privacy as an attacker's estimation error. *Int. J. Inf. Sec.*, 12(2):129–149, 2013.

[18] Reza Shokri, Julien Freudiger, and Jean-Pierre Hubaux. A unified framework for location privacy. In *Proceedings of the 9th International Symposium on Privacy Enhancing Technologies (PETS'10)*, pages 203–214. Citeseer, 2010.

[19] Reza Shokri, Julien Freudiger, Murtuza Jadliwala, and Jean-Pierre Hubaux. A distortion-based metric for location privacy. In *Proceedings of the 8th ACM workshop on Privacy in the electronic society*, pages 21–30. ACM, 2009.

[20] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 247–262, Washington, DC, USA, 2011. IEEE Computer Society.

[21] Haitao Xu, Haining Wang, and Angelos Stavrou. Privacy risk assessment on online photos. In Herbert Bos, Fabian Monrose, and Gregory Blanc, editors, *RAID*, volume 9404 of *Lecture Notes in Computer Science*, pages 427–447. Springer, 2015.

Part II
Location Privacy and Mobility Prediction

Chapter 4
MobiDict - A Mobility Prediction System Leveraging Realtime Location Data Streams

Published in the proceedings of
The International Workshop on
GeoStreaming (IWGS'16)
Burlingame, CA, USA, October 31, 2016
Publisher: ACM
Co-first author with Vaibhav Kulkarni

Abstract Mobility prediction is becoming one of the key elements of location-based services. In the near future, it will also facilitate tasks such as resource management, logistics administration and urban planning. To predict human mobility, many techniques have been proposed. However, existing techniques are usually driven by large volumes of data to train user mobility models computed over a long duration and stored in a centralized server. This results in inherently long waiting times before the prediction model kicks in. Over this large training data, small time bounded user movements are shadowed, due to their marginality, thus impacting the granularity of predictions. Transferring highly sensitive location data to third party entities also exposes the user to several privacy risks. To address these issues, we propose MobiDict, a realtime mobility prediction system that is constantly adapting to the user mobility behaviour, by taking into account the movement periodicity and the evolution of frequently visited places. Compared to the existing training approaches, our system utilises less data to generate the evolving mobility models, which in turn lowers the computational complexity and enables implementation on handheld devices, thus preserving privacy. We test our system using mobility traces collected around Lake Geneva region from 168 users and demonstrate the performance of our approach by evaluating MobiDict with six different prediction techniques. We find a satisfactory prediction accuracy as compared to the baseline results obtained with 70% of the user dataset for majority of the users.

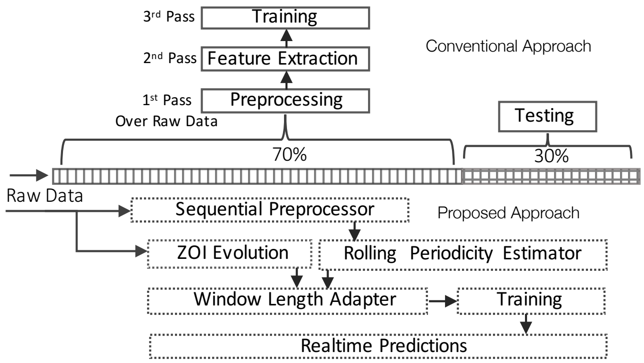


Fig. 4.1: Traditional Prediction Systems *vs.* MobiDict. The process on the top depicts the traditional mobility prediction approach, while the process chain shown at the bottom gives an overview of our technique.

4.1 Introduction

In recent years, we have seen a rapid proliferation in the number of applications offering location-based services. Popular applications such as *Google Now*,[1] collect and utilise sensitive data such as, location history, agenda and contact list, to infer and assist users in everyday activities. Another well-known application, *Moves*[2] enables to automatically identify the transportation mode from collected data and display relevant information on the fly, such as the number of burnt calories. On the similar lines, *Google Maps* is equipped to predict where the user wants to go next based on the location history.[3] As evident from the above examples, mobility prediction is becoming a key paradigm of location-based services.

Problem. The services described above, demand a large volume of data in order to provide relevant mobility predictions. Existing works in this domain utilise more than 70% of the entire dataset, exclusively for the training purpose [1, 10, 21] as depicted in Figure 4.1 under conventional approach. The duration of the datasets, used in the literature usually lasts for more than a year, which amounts for a considerable time, explicitly for model training [19, 34]. This results in a substantial waiting time until the model is able to produce usable predictions in real deployment scenarios.

Another issue associated with learning on a large dataset is the shadowing effect on small user movements that appears insignificant, but affects the granularity of predictions. Existing works attempting to address the above problem link user behaviour with forecasting models, which on the hindsight only results in statistical

[1] Google Now: `https://www.google.com/intl/fr/landing/now/`

[2] Moves: `https://www.moves-app.com`.

[3] Google Maps Predictions: `https://www.searchenginejournal.com`

prediction models without truly capturing the inherent nature associated with user movements [8].

Collecting a substantial quantity of user locations also leads to a privacy issue. A malicious entity can infer sensitive information related to the user, making it relatively easy to discover a particular place by using simple heuristiques [12] and identifying the user [6]. The algorithmic cost of making predictions on a mobile device in a real deployment scenario is relatively high due to the expensive ensemble techniques, combined with complex and extreme learning models, which makes it essential to have a centralized server [14].

Contributions. The fundamental goal of our approach is to restrict the amount of data required for training the mobility models, to small time windows usually lasting for a couple of weeks. Our solution analyzes the substantive user mobility behavioural changes in realtime and incorporates the associated changes to adapt the length of the time window required for training. We explore the evolution of the frequently visited places by the user according to the time and the associated periodicities among those places as a means to quantify user behaviour and couple it with the prediction process to give rise to quick realtime predictions as shown in Figure 4.1 under proposed approach. This process takes place in realtime, over sequential location data that is operational on a mobile device, thus ensuring that no personal location data is transferred to the location-based services. However, in order to utilise these services, only the predicted locations can be shared to maintain the utility/privacy tradeoff space. More specifically, the paper makes the three contributions listed hereafter.

- We propose MobiDict, a mobility prediction system on realtime sequential data in order to forecast user location. This system adapts user mobility model constantly, according to the user behavioural changes. Consequently, utilising considerably less data as compared to the conventional approaches of formulating predictive models and achieving satisfactory prediction accuracy.
- The lower computational complexity, resulting due to the lesser data involved leads to implementation on hand held devices feasible. Thus, eliminating the need to transfer highly sensitive user raw data to third party entities and ensuring user privacy. This enables to avoid the usual long and strenuous training period involved in generating the prediction models, obtaining quicker predictions.
- The reactive zone of interest computation scheme, incorporated in MobiDict, helps to model the mobility behaviour, restricted to small time periods as compared to modelling on long duration data, where the true nature of user behaviour is lost. This enables to make predictions during those small periods with higher accuracies as compared to the conventional approaches.

The rest of the paper is organised as follows. Section 4.2 discusses research efforts similar to ours and Section 4.3 presents our system model and introduces some formal definitions and notations used in the paper. Then, Section 4.4 describes our approaches of quantifying user behaviour. In Section 4.5, we present the different prediction techniques used in the MobiDict system. Section 4.6 presents MobiDict as a whole, showing how the elements presented in the two previous sections fit together to make up the complete system. We discuss the results of a thorough experimental evaluation of our approach in Section 4.7, based on mobility traces collected on our campus by Nokia Research, from 184 users, between October 2009 and March 2011. Finally, Section 4.8 concludes the paper by sketching future research directions around MobiDict.

4.2 Related work

We breakdown the literature review in areas concerning mobility modelling, and mobility prediction.

A domain of works apply sequential mining to extract frequently visited regions with mean travel time, to formulate the mobility models [4, 26]. The above approaches rely on clustering visits to form a visit region. We reviewed several location based clustering works [2, 9, 12, 18, 33]. As opposed to their approach of analysing the entire dataset to cluster the individual regions, we form the models by obtaining the zones in realtime, and characterising the mobility behaviour, dependent on the evolution of the number of zones with time. There exists several studies regarding stream data clustering including real-time analysis (see [5, 16]) but they fail to realistically monitor the evolution of the zones according to time. Other domain of this work falls under modelling the movements as a whole, such as the continuous-time random-walk (CTRW) [25], Levy-flight nature [29] and daily activity analysis and dissimilarities within them as presented in [17]. Although the above models aid in predicting human mobility, the mobility models derived by offline analysis are computationally expensive to be applied to raw location data thus not feasible for making swift online predictions.

Detecting periodicity in time series data is a widely studied problem to predict trends in the data stream. [28] computes the periodicity by analysing the user's visit frequency of places and aggregating total time spent at those places followed by applying Fourier transform to this series. This knowledge is used to predict the users next visit as shown in [3, 24]. However, the analysis are based on the computations on the complete dataset, as opposed to our work in real time location log preprocessing and retrieving non-stationary and non-frequent periodic patterns lasting only for small time intervals. We did not find existing literature to formulate prediction models in realtime based on periodicities, whose instances may be shifted or distorted.

We focus the literature review regarding prediction techniques that first formulate a mobility model and consequently use it to make predictions. More specifically, prediction tasks that address the task of forecasting the next user move based on the users current location. The results of the works, based on this technique [8, 8, 15, 21], show that it is possible to attain accuracies in the range of 60-80%. Several approaches have been used to make the predictions, ranging from Markov based predictors, neural networks, dynamic bayesian schemes, decision trees having several tradeoffs as compared to each other for next place prediction as summarised in [22, 27]. The learning based predictors fall under the category of predictive modelling, association analysis and cluster analysis. The next place predictions derived using the above approaches by having a trained model mapped to 70% of the dataset are presented in the works of [1, 30, 31]. [7] discusses several approaches for learning over sequential data including sliding window methods, conditional random fields and graph transformer networks. Further, Kalman filter based prediction approaches cannot be applied to non-stationary data, involves higher complexity and thus results in higher latency as discussed in [20]. Our approach falls under learning over streaming location data using a recurrent sliding window technique where we adapt the window length for training depending on the mobility behaviour.

4.3 System Model

Hereafter, we introduce our system model, together with formal definitions and notations used in the paper.

User and Locations. We assume a moving user carrying a mobile device whose locations are tracked by Global Positioning System (GPS) and/or Wi-Fi positioning system (WPS). The device regularly receives user's raw location logs as a sequence $L = \langle loc_1, loc_2, \ldots, loc_n \rangle$, where $loc_i = (\phi, \lambda, t)$ is a 3-item tuple representing a location in the format (latitude,longitude,timestamp). In the rest of the paper, we use the notation $loc.\phi$, $loc.\lambda$ and $loc.t$ for the tuple elements.

Zone of Interest. Everyday activities of a user might consist of some location points that she might find useful or spend considerable amount of time. A Zone of Interest (ZOI) is a similar concept that depicts a region encapsulating several of these points. A ZOI is not strongly bounded to any location due to the temporal constraints. It begins when the human activity at a location is initiated and ends when the activity decays. At this stage the ZOI is tied up to the relocated location.

User Mobility Model. Collecting real-life mobility data of users, which is complex, yields mobility traces of individuals. Statistical analysis of these trajectories unfolds hidden patterns to turn this raw data into mobility knowledge. This results in abstracting away from the cluttered data and discover general movement patterns respective to individuals. Simply put, mobility models are these generalisations of movement patterns representing a user.

4.4 Mobility behaviours

4.4.1 Zone of Interest Evolution

This section highlights the complete process of discovering the ZOIs and their evolution which forms an integral part of the user mobility behaviour.

4.4.1.1 ZOI Discovery

The discovery of ZOIs can be divided into three distinct steps. We read the user dataset sequentially so as to simulate the realtime streaming of user locations.

Cluster Discovery. A cluster intuitively contains, locations having common spatial and temporal characteristics. $\Delta d_{max} \in \mathbb{R}$ and $\Delta t_{min} \in \mathbb{N}$ represents a distance in meters and a minimum time threshold respectively. The two following functions are considered: $centroid(\langle loc_1, loc_2, \ldots, loc_n \rangle)$ computing and returning the centroid, which maps the individual locations into the geometrical centroid based on the set distance, and $distance(loc_i, loc_j)$, which computes and returns the Euclidian distance between the two locations loc_i and loc_j.

We consider a subsequence l' of L that contains n successive locations, such that $l' = \langle loc_{s_i}, loc_{s_{i+1}}, \ldots, loc_{e_i} \rangle$. This subsequence l' becomes a cluster iff the following conditions 4.2 and 4.2 are satisfied:[4]

$$\begin{gathered} \forall k \in \{s_{i+1}, \ldots, e_i\}, \\ distance(centroid(loc_{s_i}, \ldots, loc_{k-1}), loc_k) \leq \Delta d_{max} \end{gathered} \tag{4.1}$$

$$loc_{l_{e_i}}.t - loc_{l_{s_i}}.t \geq \Delta t_{min} \tag{4.2}$$

From this cluster, defined as l', we can extract its centroid $centroid = (\phi, \lambda)$ that is the barycenter of all ϕ and λ of the locations contained in l', in which $\phi \in \mathbb{R}$ is a latitude and $\lambda \in \mathbb{R}$ is a longitude. We can also compute the radius of the cluster $\Delta r \in \mathbb{R}$ that is the maximum distance between the centroid of the cluster and a location of the set of locations extracted from l'. We define a cluster as a 4-item tuple $c = (\phi, \lambda, \Delta r, l')$. The notation $c.centroid$ is used to designate the centroid of

[4] This clustering process is inspired by a technique called *DT cluster* and presented in [12].

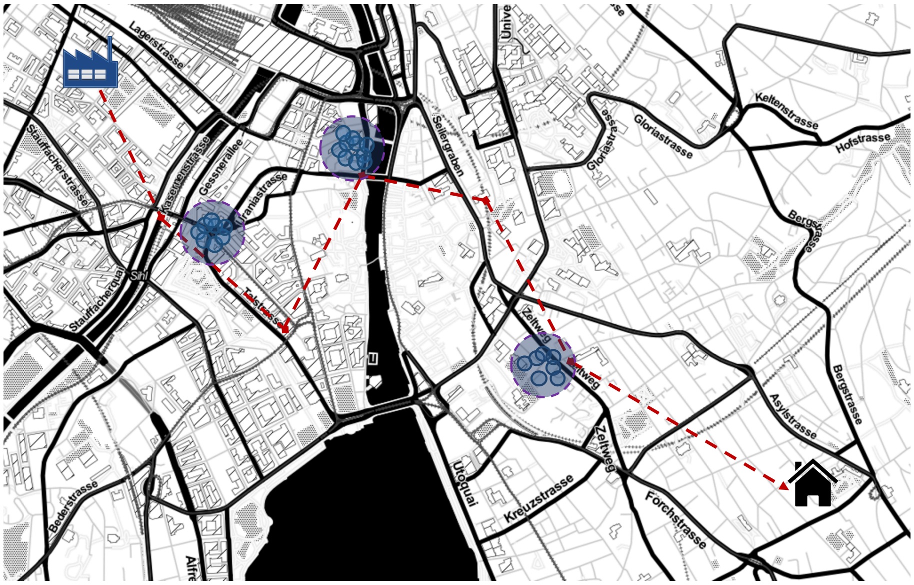


Fig. 4.2: ZOI Construction from Cluster of Location Points.

the cluster c. Hereafter, $C = \{c_1, \ldots, c_i, \ldots, c_m\}$ is the set of m clusters associated with the user, based on her sequence of locations. It is important to note that C must meet the following condition 4.3:

$$\forall c_i, c_j \in C, c_i.l' \cap c_j.l' = \emptyset \tag{4.3}$$

This first part of the clustering process is based on a well-known technique described in [12] and presented as *density-time cluster (DT cluster).*

Conditions 4.2 and 4.2 do not guarantee pairwise disjointness of clusters that is in turn used to form cluster groups as explained further.

Cluster Group. A cluster group includes all the clusters that can be assembled iff an intersection exists between these clusters. Thus, two clusters $c_i, c_j \in C$ are included in the same cluster group g iff the next condition in Equation 4.4 is met:

$$\begin{aligned} &distance(c_i.centroid, c_j.centroid) \\ &\quad - \ (c_i.\mathit{\Delta r} + c_j.\mathit{\Delta r}) < 0 \end{aligned} \tag{4.4}$$

A cluster group is a 4-item tuple $g = (\phi, \lambda, \mathit{\Delta r}, \{c_1, c_2, ...\})$, where $\phi \in \mathbb{R}$, $\lambda \in \mathbb{R}$, $\mathit{\Delta r} \in \mathbb{R}$, $\{c_1, c_2, \ldots\} \in C$ are latitude, longitude, radius and array of clusters constituting g respectively. The centroid of the cluster group is defined by (ϕ, λ), being the mean of all the centroids of the clusters included in g. The following set G contains all the discovered cluster groups, such as $G = \{g_1, g_2, \ldots\}$.

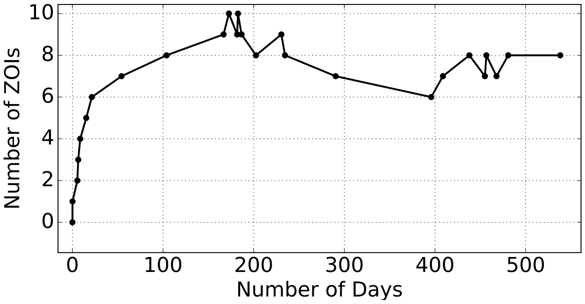


Fig. 4.3: ZOI Evolution Over Time.

ZOI. A ZOI is a frequently and recently visited zone by a user in everyday life. We introduce two constants *visitThreshold* $\in \mathbb{N}$ and *maxTimeDuration* $\in \mathbb{N}$ represent a maximum threshold of visits and a maximum duration threshold between two dates respectively, and a variable called $minVisitNumber \in \mathbb{N}$, which represents a minimum number of visits. Then, *size*(g) is a function that computes and returns the number of clusters of the cluster group g, *meanVisitNumber*(G) is a function computing and returning the mean number of visits amongst the set of all cluster groups G and *timeDuration*(G) is a function that returns the duration between the current date of the system and the last visited date of the cluster group contained in G. *meanVisitNumber*(G) returns values that dynamically change over time according to the mobility behaviour of the user due to the realtime nature of the process. $minVisitNumber$ is equal to the value returned by *meanVisitNumber*(G) until reaching the *visitThreshold*, which is the maximum number of visits that converts a cluster group into a ZOI. A cluster group $g \in G$ is transformed into a ZOI z iff the conditions of Equations 4.5 and 4.6 are satisfied:

$$\begin{aligned} size(g) &\geq minVisitNumber \\ \mid \ \ minVisitNumber &= meanVisitNumber(G) \\ \wedge \ \ minVisitNumber &<= visitThreshold \qquad (4.5) \\ timeDuration(G) &<= maxTimeDuration \qquad (4.6) \end{aligned}$$

A ZOI z is formally, a four item tuple $z = (\phi, \lambda, \varDelta r, g)$, where $\phi \in \mathbb{R}$, $\lambda \in \mathbb{R}$, $\varDelta r \in \mathbb{R}$ and g are the latitude, longitude, radius and the cluster group becoming a ZOI respectively. The tuple (ϕ, λ) is the centroid of z computed from group g. The set Z is finally the set of ZOIs of the user, such that $Z = \{z_1, z_2, \cdots, z_n\}$ as shown in Figure 4.2.

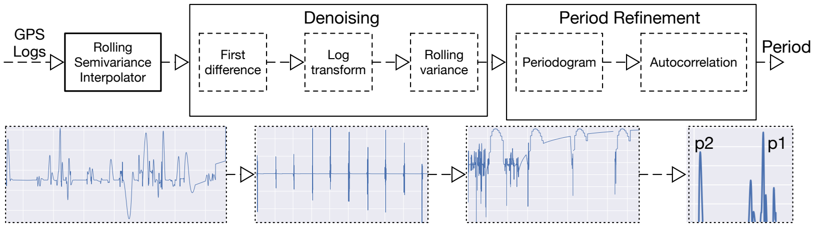


Fig. 4.4: Realtime Periodicity Estimation Chain.

4.4.1.2 ZOI Evolution

A user's ZOIs may change over time and space. Figure 4.3 shows an example of ZOI updates occurring over time for a certain user having location data of more than 500 days. We see a surge of ZOI updates at the beginning, minor variations intermediary and attains a flat tail towards the end. Monitoring this trend of ZOI evolution according to time reflects the changing user behaviours. Thus the number of ZOIS and their evolution can be used to quantify user mobility behaviour.

4.4.2 Periodicity of Movement

Human mobility is characterised by a high degree of periodicity. Detecting these periodic behaviours can assist to generate quick predictions, evading the complex training procedure. However, one of the challenges is to identify periods that do not repeat precisely at the same times, in addition to having multiple interlaced patterns in the non-stationary time series. As a result, standard period estimation techniques such as autocorrelation or Fourier transform cannot be directly applied. We describe the steps involved to accurately detect the movement periodicity below.

Uniform Location Sampling. One of the fundamental drawbacks of the periodicity detection algorithms is the prerequisite that the incoming location stream should be uniformly sampled. However, location logs coming in at non uniform rate is common in communications due to imperfect geolocation sensors or network unavailability to stream logs online. When the sampling is nonuniform, a common technique is to resample/interpolate the signal onto a uniform grid. We use semivariance interpolation on the incoming stream using moving average construction.

In a nutshell, the semivariance conceals the incoming data stream about the spatial variance at a specified distance. We find that Gaussian model provides accurate fitting to the missing data after calculating the semivariance. The semivariance along with Gaussian model allows to model the similarity between points in a filed as a function of changing distance. The semivariance can be mathematically expressed in Equation 4.7 as:

$$\delta_h = \frac{1}{2N_h} \sum_{N_h} (R.\sqrt{(\delta_x.\cos_\theta)^2) + \delta_y^2})^2 \tag{4.7}$$

where δ_h is maximum distance separation among the location logs, N_h are the number of points separated by the distance h. The semivariance is than the sum of the squared difference between these values. To calculate the distance, we utilise equirectangular distance approximation, which is faster as compared to the Harversine formula. In addition, as the distances traversed are usually small, the performance is superior compared to great circle distance approximation.

Dealing with Non Stationary data. Applying signal processing techniques, directly to estimate the user movements and periodicity to non-stationary data puts forth several challenges. The interpolation step is followed by taking the first difference of the streaming interpolated location logs. This step brings forth the trends present in the movement data by exposing the variance for further processing. Next, in order to estimate the magnitude of day-to-day variations, log transform is applied to the series. The rolling variance applied to the logged series, results in a series of constant variance.

Periodicity Estimation. For the final step of the periodicity detection, we compute the rolling autocorrelation over the precessed stream. Next, we calculate the Power Spectral Density (PSD) to get the candidate periods and feed them into the autocorrelation estimator so as to rectify false alarms resulting due to the spectral leakage. The robust autocorrelation routine results in the computation of statistically significant period(s) contained in the non-stationary and noisy location stream. The complete processing chain is shown in Figure 4.4. Through this process, we are able to detect weekly periodic patterns. We focus on estimating short repetitive patterns and detecting larger periods such as holiday vs. non holiday pattern is beyond the scope due to the data limitations we impose.

The above two demeanours, i.e., evolution of ZOIs and movement periodicities, serve as a basis to decipher user mobility behaviours, which play a key role to alter the realtime model formulation and assist in prediction.

4.5 Mobility Predictors

In this section, we describe the different families of prediction techniques used to forecast the next user location. We specify the procedure involved in training these predictors in the context of mobility forecasting. In the plethora of predictor options available, we select the ones that have been already proved successful for spatiotemporal prediction by published works [11, 21]. The starting time at a ZOI and the total spent duration serve as input features to the prediction techniques. The pre-

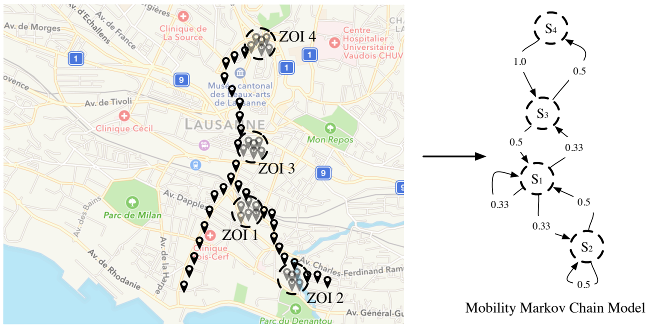


Fig. 4.5: From User's ZOIs to MMC Model.

diction task is then modelled to perform one-step-ahead forecasting on the basis of available data gathered in a time window.

Mobility Markov Chain. A Mobility Markov Chain (MMC) model is described by a state-transition matrix including the user's ZOIs, which are the states, and all the transitions amongst them. These transitions, collected during a training period, feed the model. We assume a set S of n states discovered at a current time t such that $S = \{s_1, s_2, \dots, s_n\}$ knowing that $S = Z$. This set of states is the baseline to build the matrix. Then, by exploring the user's raw locations, we can extract a sequence of vectors V, where one vector contains two successive states, visited by the user such that $V = \langle (s_1, s_2), (s_2, s_2) \dots, (s_1, s_n) \rangle$. The matrix is then filled according to the sequence V by computing the transition probability of each vector included in V. The transition probability to move from s_i to s_j is expressed in Equation 4.8 as follows:

$$p_{s_i,s_j} = P(s_j|s_i) \tag{4.8}$$

Then, the predicted next state is the most likely state s_{next} found on the basis of all computed transition probabilities of the matrix line of the current state s_c as expressed in Equation 4.9:

$$\forall s_i \in S : pred_{s_{next}} = argmax(p_{s_i,s_c}, p_{s_{i+1},s_c}, \dots, p_{s_n,s_c}) \tag{4.9}$$

Figure 4.5 shows the creation of a MMC model of a user as well as the state-transition matrix based on the evolution of the ZOIs of a user. Unlike the 1-order Markov chain, which is described above, the 2-order Mobility Markov Chain is slightly different, as the previous state is also taken into account in the prediction process, which increases the prediction accuracy as presented in [13].

Classification Based Learning. Predicting the next location can be viewed as a classification task, where the training and the discrete output class consist of the currently computed and active ZOIs according to the user behaviour. This set consists of permanent zones and temporary zones that may vanish with time. Thus, every learnt model bounded by a particular time instant consists of ZOIs that may not be active in prediction models formulated at another time instant. We employ an elementary 1-NN based classifier that uses a training point closest to the query point to predict the output label. If $X_i = \{\{x_p\}, \{x_t\}\}$ is the input vector consisting of permanent and temporary zones $\{x_p\}$ and $\{x_t\}$ respectively, the training set consists of $\{(x_1, y_1), (x_2, y_2), ...(x_n, y_n)\}$, where y_i is the next zone, traversed according to the time series sequence. Thus, the task here is to determine y_{new} for x_{new} that is performed by finding the closest point x_j to x_{new}, w.r.t. the euclidean distance.

Artificial Neural Network. In the ANN model, the training patterns are highly dependent on the training window length to model the forecasting as predictive regression problem. Each pattern will consist of the number of ZOI movements recorded in the training window. $pattern1 = x_1, x_2, ...x_n$, $pattern2 = x_2, x_3, ...x_{n+1}$ correspond to the number of input nodes, where each node represents a pattern collected for a day, thus training the model using the sliding time window approach. The model consists of one output node and the number of nodes in the hidden layer is determined empirically. The 3 layered feed-forward neural network with back propagation can be represented in Equation 4.10 as below:

$$y_i = f(g((\sum_j w_{ij}.in_j) - \theta_j)) \tag{4.10}$$

where w_{ij} is the weight vector, θ is the bias, in_j is the input layer representing the movement patterns for a day.

Recurrent Neural Networks. Recurrent neural networks have a memory layer that is advantageous to model long term time series data, by assisting the inputs to be correlated with input/output pairs, which even lie beyond the current window length. Similar to the previous description, we use a three layered architecture where the hidden layer is recurrently connected to itself. The network can be represented in Equation 4.11 as below:

$$y^i = w_2.\sigma(w_1.x^i + w_r h^{i-1}) \tag{4.11}$$

where σ is a non-linear transfer function, here, we use sigmoid function. w_1, w_2 are the connecting weights and w_r are the recurrent weights.

Fourier Extrapolation. We also test MobiDict with Fourier extrapolation, which is capable of deconstructing the time series as a polynomial base, with bounded randomness and a cyclic component. The frequency domain, due to its nature, transforms the time bounded user visits in time domain into fixed cycles. The extrapolation yields a de-noised copy of the movements observed in the history. Thus performing prediction over a time window of t time units. Here, the high frequency components are used to estimate movements over regular ZOIs and the low

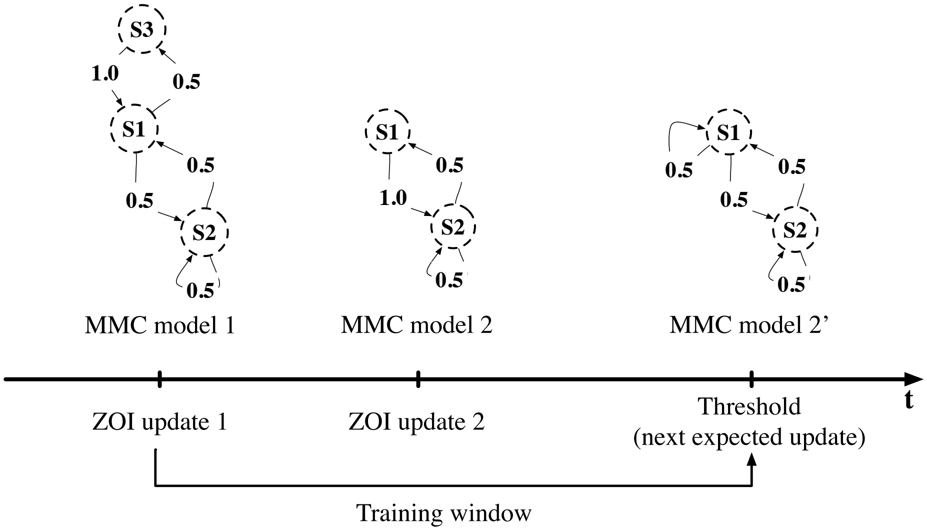


Fig. 4.6: MMC Training Window.

frequency components, to predict irregular user movements restricted to small time bounds.

We implement the above machine learning prediction techniques using PyBrain [32] and the MMC models are generated within a Cocoa application where the entire realtime process is implemented.

4.6 The MobiDict System

In this section, we present the MobiDict prediction system design and illustrate how the mobility behaviours are coupled with the predictors to produce the next location prediction in realtime. The overview of our approach is presented in Figure 4.1, which depicts the fundamental elements involved in our system. As described in Section 4.4, we present two approaches to quantify user behaviour. Due to the nature of MMC, movement periodicity cannot be directly integrated into the model, thus we base MMC only on the ZOI evolution aspect. However, in case of the other machine learning techniques, we involve the periodicity associated with the movement within the evolving ZOIs. We perform a systematic evaluation of MobiDict by testing it with all the described prediction approaches to analyse the prediction accuracies. We now describe how the MMC and the machine learning based system individually integrate the respective mobility behaviours to produce next place predictions. The common goal being, formulation of a robust predictive system on streaming location data.

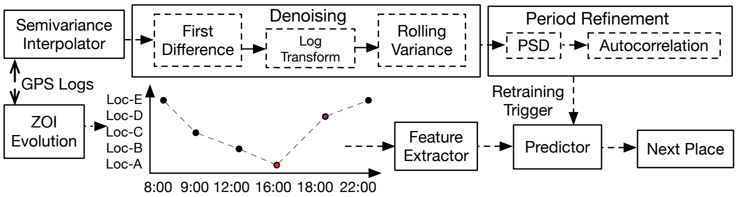


Fig. 4.7: Realtime Periodicity Aware Training Process.

4.6.1 MMC-based System

A Mobility Markov Chain model only depends on the states and the transition probabilities amongst them. This property bears similarity with the evolving ZOIs of the user over time representing the behaviour. Therefore, to implement MobiDict, we combine the evolution of ZOIs with the creation of the user's MMC model. Figure 4.5 presents an intuitive description of what is a ZOI significant update, which basically triggers the adaptation of the user model every time there is a new significant update. This preserves the relevance of the user mobility model.

Figure 4.6 shows the successive training windows, the ZOI updates and the updates of the mobility model. The training window is initiated when there is a significant ZOI update and ends when a certain threshold is exceeded. An update is considered as significant when either a new ZOI is added to the ZOI set or removed from it under the assumption that this set contains more than one ZOI. At each update, the user's MMC model is rebuilt according to the entire state sequence S that is updated in realtime by taking into account user's raw locations. As seen in Figure 4.3, many updates are sometimes accumulated, in such cases, the mean time between two updates is used to compute the threshold of the training window. The next expected update is triggered by adding the mean time between all the past updates u_{mean}. The next threshold t_{next} can be formally expressed in Equation 4.12 as below, in which $date_c$ is the current date of the system:

$$t_{next} = date_c + u_{mean} + \frac{u_{mean}}{2} \tag{4.12}$$

If this threshold is exceeded without having detected the expected significant update, the training window is interrupted. At the end of the training window, the MMC model is also updated in order to take into account the entire state sequence collected during the window as well as the one formulated during previous training windows.

4.6.2 Machine Learning-based System

The system should take into consideration the recent movement histories and the associated periodicities in order to produce an updatable mobility model. The problem can be formulated as a non-stationary time series prediction, where the model needs to be retrained according to variations in the incoming data stream. In our case are the user movements and the variations are linked to changing periodicities. We empirically determine that the model accuracy is affected for an autocorrelation index change of 0.2 and greater. This serves as a trigger for periodic and incremental model retraining, where the batch size consists of movement histories, with the changed periodicity bounds.

We first describe the realtime processing chain, as shown in Figure 4.4 to estimate the changing periodicities. As described in Section 4.4, we perform Fourier analysis that expresses the function, as summation of individual periodic elements. Further, we compute the power spectral density to find the strength at each frequency, and only the dominant frequency components are selected. The periodogram highlights the periodicities lasting for short and medium terms, on the other hand, autocorrelation is suitable for large period detection. We combine the approaches so as to filter out harmonics and get refined candidate periods. This can be formally expressed in Equation 4.13 as below:

$$P(\frac{C_k}{N}) = \left\| \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n).e^{\frac{-j.2\pi.c.n}{N}} \right\|^2, c = 0, 1 ... \frac{N-1}{2} \tag{4.13}$$

where C_k are the strength encodings at a given frequency k, $x(n)$ are the spectral coefficients associated with the sinusoids and N is the total number of strength encodings at a particular frequency.

We track the periodicity continually, following the above approach. Regarding the training phase, as depicted in Figure 4.7, the ZOI evolution is tracked to form a feature vector representing the movements across them. The other features consist of the starting time and stay time at a particular zone. The extracted feature vectors are fed to the predictors described in Section 4.5. The periodicity feature is tracked to monitor if it changes by 0.2. At this point, the training reinitiates to reform the mobility model, taking the new periodicities, thereby adapting to current behaviour of the user.

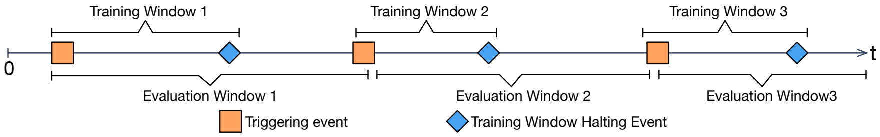


Fig. 4.8: Realtime Evaluation Scheme.

4.7 Experimental Evaluation

In this section, we demonstrate the experimental results of our approach based on the Nokia data set [19] consisting of mobility traces, collected from 184 users in Switzerland from October 2009 to March 2011. The participants consisted of university students and employees with a mean duration of 14 months comprising of more than 10 million location points. Amongst the users of this dataset, we only select 168 of them having a dataset duration of at least 30 days.

4.7.1 Experimental settings

Here, we describe the selection of users from the dataset, as well as the choices made, behind the algorithmic parameters. In order to obtain the ZOIs of a user, we set a value of 60 meters for Δd_{max}, 900 seconds for Δt_{min} to cluster the individual locations with respect to space and time. The $visitThreshold$ parameter is set to 6 visits and the $maxTimeDuration$ of 3 months. In order to determine the above parameters, we analyse the complete dataset to compute the average of the mean number of visits of all cluster groups of each user per month. This choice was based on selecting users having a dataset duration of at least 30 days. In order to simulate realtime incoming data, we read the data-points sequentially according to the logged timestamps.

4.7.2 Real-time Evaluation Scheme

Figure 4.8 describes the evaluation approach, followed to compute the prediction accuracy over time for each user. This scheme shows the successive training windows, as well as the consecutive evaluation windows. In the case of the 1-order and 2-order MMC, this trigger is a significant update about the set of the user's ZOIs, while in the case of the other learning based approaches, we rely on a significant change in the autocorrelation index, representing the user periodicity. It is important to note that all the information collected during the previous training windows is also taken

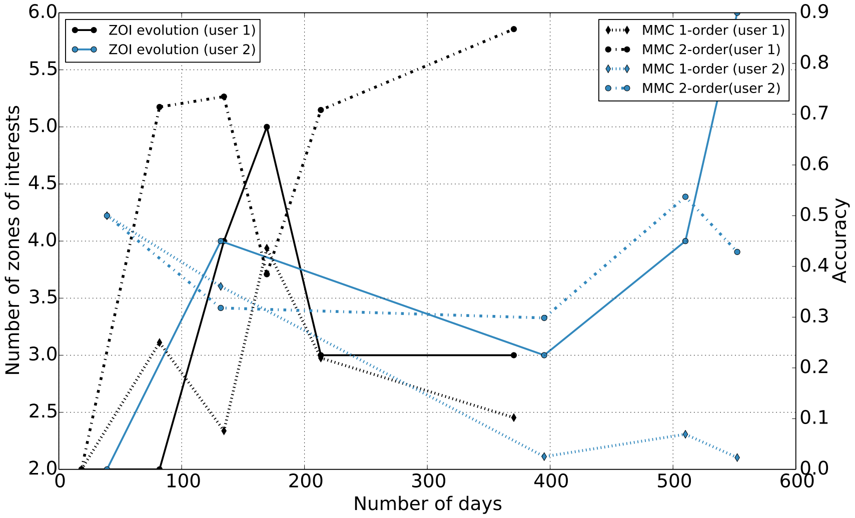


Fig. 4.9: Evolution of ZOIs and Prediction Accuracy Over Time of 2 Users According to 1-order and 2-order MMC.

into account for the next training windows. The evaluation window commences at the very first trigger, which is when the first two ZOIs of a user are computed. During a training window, the model analyses the user's movements to construct a user specific mobility model according to the techniques described in Section 4.5. The MobiDict system is evaluated with respect to each family of predictors. At the beginning of the every new training window, the prediction accuracy result is computed. As the evaluation metric, we consider the prediction accuracy, which is the fraction of samples for which the model successfully predicts the next location during the evaluation window.

4.7.3 Results and Discussion

We evaluate the performance of MobiDict by comparing it against the accuracy obtained by using the conventional approach of formulating a model, trained on 70% of the dataset and evaluated on the rest. The resulting accuracy that we use for baseline comparison for all the predictor families is shown in Table 4.1. We also compare our baseline results with the results obtained by existing works on the same dataset and achieve similar accuracies with the same feature selection techniques.

Figure 4.9 depicts the evolution of the user's ZOIs and the prediction accuracy computed with the 1-order and 2-order MMC prediction technique over time. We consider two different users to deeper describe the results obtained. They are con-

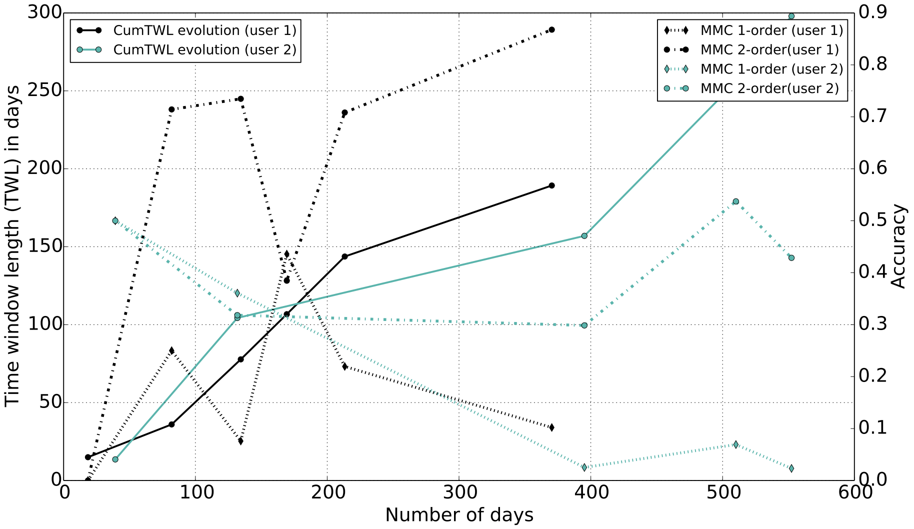


Fig. 4.10: Evolution of Cumulative Time Window Length and Prediction Accuracy Over Time of 2 users According to 1-order and 2-order MMC.

tained in the Nokia dataset having different dataset durations, i.e., more than 350 days for the first user and more than 500 days for the second user.

We obtained higher prediction accuracies with 2-order MMC as compared to 1-order MMC for majority of the users as also depicted in case of these two users. This is mainly due to the fact that, 2-order MMC takes into account the current user's state and the previous state to search the next state in the model, improving the quality of the predictions. We also observe that, when the number of ZOIs has a sudden shift, the accuracy does not necessarily decrease with this drastic variation. For instance, at the end of the evolution of the number of ZOIs of user 2 (i.e., from point 4 to point 5), there is an increase of two zones, however, the prediction accuracy is not significantly affected for both the MMC. With the decrease of two zones (i.e., from point 3 to point 4 for user 1), the accuracy of 2-order MMC increases, while that of the 1-order decreases. Here, we assume that some variations may sometimes require longer training periods to obtain relevant MMC model according to the changes, as the predictions are strongly linked to the transition probabilities contained within them.

In Figure 4.10, cumulative training window lengths are depicted according to the accuracy and the evolution of user movements according to time. We see, a clear trend in the number of days taken to compute the predictions at a specific time. With the considered two users, we observe that, there is no absolute requirement to use a large amount of data to obtain satisfactory prediction accuracies with the MMC prediction technique, because, with less than 100 days, we can obtain accuracy of more than 0.5. Regarding the entire dataset analyses, 34% of users

Technique	Accuracy (%)
1-order MMC	57.19
2-order MMC	61.66
1-NN	59.28
ANN	60.85
RNN	72.79
Fourier ext.	63.87

Table 4.1: Baseline Results.

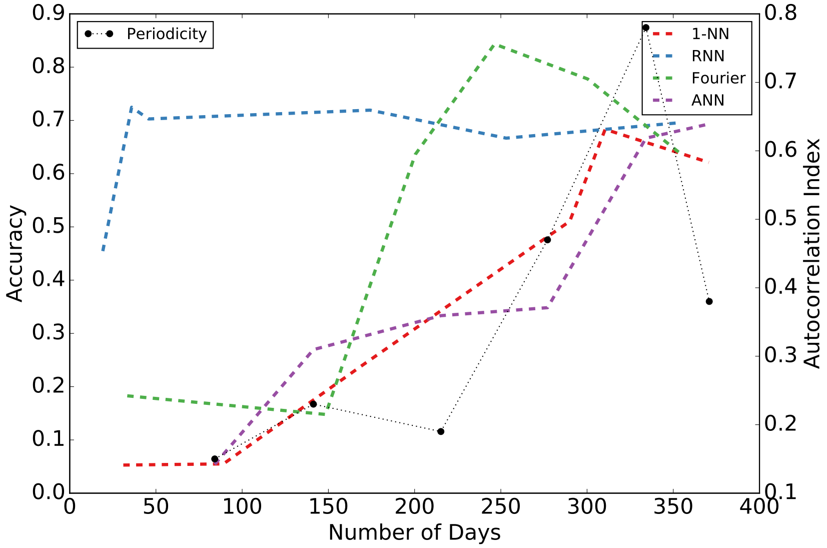


Fig. 4.11: Variation of Accuracy with Time and Movement Periodicity.

reach a satisfactory accuracy with less than 100 days. We also assume that this is closely linked to the quality of the information, i.e., transitions between 2 or more states, included into the model during the training windows. In addition, it is also important to note that, in realtime and for the MMC techniques, we use raw data without any refinement, which could affect the quality of the user's mobility model.

Next, we analyse the effect of movement periodicities, on the accuracies of the learning based predictor families. As shown in Figure 4.11, we see a clear correlation between the periodicity and the accuracy of classification, neural networks and the Fourier based approach. However, we also see that, recurrent neural network has no visible impact of user periodicities except for the minor variations. We observe this trend for majority of the users across the dataset. The main reason being, RNN's blend the input vector at the current state (i.e., the movement histories) with the previously learnt state vector to yield a new state. Thereby, taking the entire history

	Percentage of days			
Prediction technique	<=20%	>20% & <=60%	>60%	Number of satisfactory users
1-NN	168	0	0	101
ANN	103	65	0	129
RNN	37	131	0	149
Fourier ext.	65	103	0	112
1-order MMC	137	17	14	93
2-order MMC	62	41	65	142

Table 4.2: Dataset Analysis.

into account before making a prediction, thus effectively combining high level direction with low level modelling. This results in high accuracy, which is maintained stable with time. On the other hand, the classification and neural network based approach weigh the current state higher than the past depicting very high correlation with periodicity. As, with respect to Fourier extrapolation, since the individual frequency components are contribute to forecasting, the higher the periodicity the better is the accuracy.

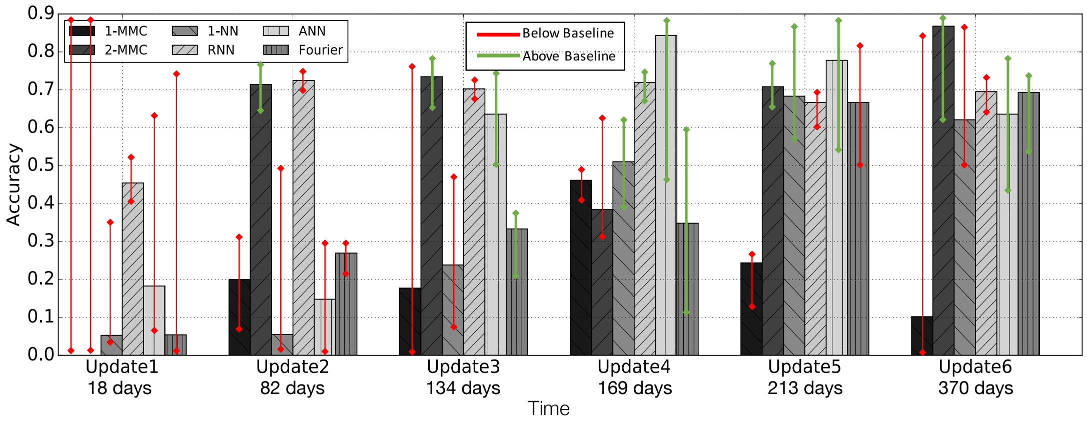


Fig. 4.12: Comparison of MobiDict Accuracy for Individual Predictors against Baseline Accuracies (example of one user).

Next, we evaluate the running accuracy difference between MobiDict for all the predictors against the baseline accuracy at each training model update as shown in Figure 4.12. As we see, the accuracies are in general lower than the baseline accuracies however, in most of the cases, accuracies are greater than 50%. After update 3, the accuracies of 2-MMC, ANN, RNN and Fourier based predictors are often higher as compared to the baselines. Regarding 1-order MMC technique, although the baseline result is not very high compared to the other techniques (i.e., 57.19%), the prediction accuracy results of the 1-order MMC model are mainly far from it. In addition, we note that the 2-order MMC accuracy results are fairly satisfactory remaining higher than the baselines for most of the time. The high baseline accuracies may also result due to the overfitting of the model when looking directly at the 70% of the dataset as compared to realtime training. Realtime training and prediction involve higher stochasticity in the time bounded noisy data that is not the case when formulating a prediction model over the complete dataset. We further evaluate the total number of users in the entire dataset providing accuracy levels higher than 50% as summarised in Table 4.2. We indicate the number of users having prediction accuracy greater than 50% in terms of total percentage of days. We observe that RNN yields the maximum number of satisfactory users whose accuracy is greater than 50% (and lower than 60%), making it an ideal predictor to be integrated in MobiDict.

Further, we approach the problem concerning the computational complexity of learning approaches by analysing the cost involved at the training time. This complexity is directly linked to a quadratic equation that involves inverting a kernel matrix having a complexity of order n^3, where n is the size of the training data [23]. The training time to arrive at an optimal solution depends on the technique used, but generally has the order of n^2. Thus, the baseline complexity is $0.7 * D$ where D is the total number of data-points collected. Therefore, the complexity in our case is $N_{up} * n^3$ where, N_{up} is the total number of updates. Further, n in our system represents the data-points included in the individual training windows, i.e., $n = t_{n_1} + t_{n_2} \ldots + t_{n_N}$, since we account for the user behaviour, which is constant for some time periods. Consequently, the total number of data-points will be lower than D, thereby having a lower complexity. The same goes for the training time.

4.8 Conclusion

With the growing ubiquity of location-aware mobile devices, the ability to analyse and predict mobility on a large scale is becoming possible, opening new opportunities but also posing new challenges. Furthermore, with mobile devices becoming more powerful every day, it becomes possible to compute mobility predictions locally, i.e., without resorting to backend servers. Yet traditional approaches to mobility prediction rely on processing large datasets on powerful backend servers. This makes mobility prediction quite tedious and slow. In addition, such centralized approaches come with a major location privacy concern, threatening the success of widespread adoption of location-based services in the coming days. This enforces a real need to restrict computations involving sensitive user data on a local mobile device.

To address these issues, we introduce MobiDict, a realtime mobility prediction system, to provide swift next place predictions. Our approach couples the prediction system with dynamic user mobility behaviours to restrict the data required for model training to short durations as opposed to conventional training approaches. This achieves accuracies exceeding 50% for about 40% of the users contained in the dataset for 2-MMC and RNN predictors. We also examine periods where our system accuracy, even exceeds the baselines. Thus exhibiting that large amount of training data is not an absolute requirement to produce viable next place predictions. We also evaluate the computational cost associated with our approach and theoretically validate the feasibility to operate on a mobile device.

We observe that certain family of predictors are more suited for particular mobility behaviours. Our future work will be an attempt to have an ensemble approach in the system to select a suitable predictor in realtime according to behavioural changes, to attain higher accuracies. We will also focus on quantifying the computational cost of the approach on an actual mobile device to confirm our hypothesis. Another area will be to optimise the process so as to have fewer number of model updates that will intern contribute to the cost.

Acknowledgment. This research is partially supported by the Swiss National Science Foundation grant 146714 and 157160.

References

[1] Theodoros Anagnostopoulos, Christos Anagnostopoulos, and Hadjiefthymiades. Mobility prediction based on machine learning. In *IEEE 12th International Conference on Mobile Data Management*, pages 27–30, June 2011.

[2] Daniel Ashbrook and Thad Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, pages 275–286, 2003.

[3] Huiping Cao, Nikos Mamoulis, and David W. Cheung. Discovery of periodic patterns in spatiotemporal sequences. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):453–467, April 2007.

[4] Xihui Chen, Jun Pang, and Ran Xue. Constructing and comparing user mobility profiles for location-based services. SAC, pages 261–266, 2013.

[5] Yixin Chen and Li Tu. Density-based clustering for real-time stream data. KDD '07, pages 133–142, 2007.

[6] Yves-Alexandre de Montjoye, César A. Hidalgo, Michel Verleysen, and Vincent D. Blondel. Unique in the Crowd: The privacy bounds of human mobility. *Scientific Reports*, 3, March 2013.

[7] Thomas G. Dietterich. Machine learning for sequential data: A review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pages 15–30. Springer-Verlag, 2002.

[8] Trinh Minh Tri Do and Daniel Gatica-Perez. Contextual conditional models for smartphone-based human mobility prediction. UbiComp '12, pages 163–172, 2012.

[9] Emre Eftelioglu, Xun Tang, and Shashi Shekhar. Geographically robust hotspot detection: A summary of results. In *ICDMW*, pages 1447–1456, 2015.

[10] Vincent Etter, Mohamed Kafsi, and Ehsan Kazemi. Been There, Done That: What your Mobility Traces Reveal about your Behavior. *the Procedings of Mobile Data Challenge by Nokia Workshop at the Tenth International Conference on Pervasive Computing*, June 2012.

[11] Vincent Etter, Mohamed Kafsi, Ehsan Kazemi, Matthias Grossglauser, and Patrick Thiran. Where to go from here? mobility prediction from instantaneous information. *Pervasive and Mobile Computing*, 9(6):784 – 797, 2013. Mobile Data Challenge.

[12] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. Show me how you move and i will tell you who you are. *Trans. Data Privacy*, pages 103–126, 2011.

[13] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. Next place prediction using mobility markov chains. MPM '12, pages 3:1–3:6. ACM, 2012.

[14] Lahouari Ghouti. Mobility prediction using fully-complex extreme learning machines. In *ESANN*, 2014.

[15] Győző Gidófalvi and Fang Dong. When and where next: Individual mobility prediction. MobiGIS '12, pages 57–64, 2012.

[16] Michael Hahsler and Margaret H. Dunham. Temporal structure learning for clustering massive data streams in real-time. In *SDM*, pages 664–675, 2011.

[17] Shan Jiang, Joseph Ferreira, and Marta C González. Clustering daily patterns of human activities in the city. *Data Mining and Knowledge Discovery*, 25(3):478–510, 2012.

[18] Jong Hee Kang, William Welbourne, Benjamin Stewart, and Gaetano Borriello. Extracting places from traces of locations. *Mobile Computing and Communications Review*, 9:58–68, 2004.

[19] Juha K Laurila, Daniel Gatica-Perez, Imad Aad, Olivier Bornet, Trinh-Minh-Tri Do, Olivier Dousse, Julien Eberle, Markus Miettinen, et al. The mobile data challenge: Big data for mobile computing research. In *Pervasive Computing*, 2012.

[20] Joseph J. LaViola. Double exponential smoothing: An alternative to kalman filter-based predictive tracking. EGVE '2003, pages 199–206.

[21] Tran Le Hung, McDowell Michele, Catasta an Lucas Kelsey, and Aberer Karl. Next place prediction using mobile data. Mobile Data Challenge by Nokia, 2012.

[22] Sunyoung Lee and Kun Chang Lee. Context-prediction performance by a dynamic bayesian network: Emphasis on location prediction in ubiquitous decision support environment. *Expert Syst. Appl.*, pages 4908–4914, 2012.

[23] Dan Levi and Shimon Ullman. Learning model complexity in an online environment. In *Computer and Robot Vision, 2009. CRV '09. Canadian Conference on*, pages 260–267, 2009.

[24] Zhenhui Li and Jiawei Han. *Mining Periodicity from Dynamic and Incomplete Spatiotemporal Data*, pages 41–81. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[25] Miao Lin, Wen-Jing Hsu, and Zhuo Qi Lee. Modeling high predictability and scaling laws of human mobility. In *IEEE 14th International Conference on Mobile Data Management*, volume 2, pages 125–130, 2013.

[26] Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. Wherenext: A location predictor on trajectory pattern mining. KDD, 2009, pages 637–646, 2009.

[27] Jan Petzold, Faruk Bagci, Wolfgang Trumler, and Theo Ungerer. Comparison of different methods for next location prediction, 2006. pages 909–918. Springer, 2006.

[28] Bhaskar Prabhala, Jingjing Wang, Budhaditya Deb, Thomas La Porta, and Jiawei Han. Leveraging periodicity in human mobility for next place prediction. In *WCNC*, pages 2665–2670, 2014.

[29] Injong Rhee, Minsu Shin, Seongik Hong, Kyunghan Lee, Seong Joon Kim, and Song Chong. On the levy-walk nature of human mobility. *IEEE/ACM Transactions on Networking*, 19:630–643, 2011.

[30] Naveen Kumar Saini and Aditya Trivedi. Refined cluster based mobility prediction with weighted algorithm. In *CICN*, pages 350–354, Nov 2010.

[31] Nancy Samaan and Ahmed Karmouch. A mobility prediction architecture based on contextual knowledge and spatial conceptual maps. *IEEE Transactions on Mobile Computing*, 4(6):537–551, November 2005.

[32] Tom Schaul, Justin Bayer, Daan Wierstra, Yi Sun, Martin Felder, Frank Sehnke, Thomas Rückstieß, and Jürgen Schmidhuber. PyBrain. *Journal of Machine Learning Research*, pages 743–746, 2010.

[33] Yihong Yuan and Martin Raubal. A framework for spatio-temporal clustering from mobile phone data. In *AGILE*, pages 22–26, 2012.

[34] Yu Zheng, Xing Xie, and Wei-Ying Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33:32–39, 2010.

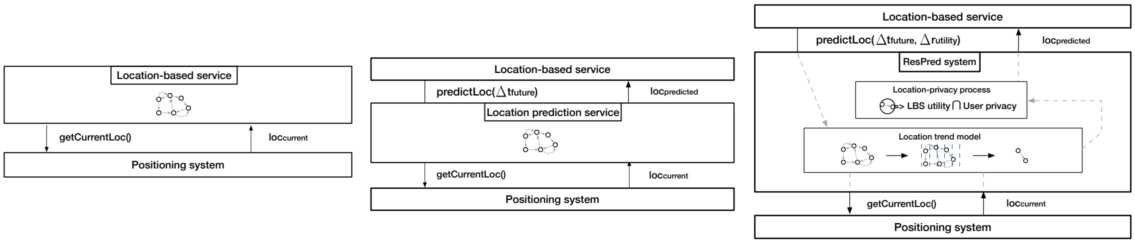
Chapter 5
ResPred: A Privacy Preserving Location Prediction System Ensuring Location-based Service Utility

Published in the proceedings of
The International Conference on
Geographical Information Systems Theory,
Applications and Management (GISTAM'18)
Madeira, Portugal, March 17-19, 2018
Publisher: SCITEPRESS
Best Student Paper Award

Abstract Location prediction and location privacy has retained a lot of attention this recent years. Predicting locations is the next step of Location-Based Services (LBS) because it provides information not only based on where you are but where you will be. However, obtaining information from LBS has a price for the user because she must share all her locations with the service that builds a predictive model, resulting in a loss of privacy. In this paper we propose *ResPred*, a system that allows LBS to request location prediction about the user. The system includes a location prediction component containing a statistical location trend model and a location privacy component aiming at blurring the predicted locations by finding an appropriate tradeoff between LBS utility and user privacy, the latter being expressed as a maximum percentage of utility loss. We evaluate *ResPred* from a utility/privacy perspective by comparing our privacy mechanism with existing techniques by using real user locations. The location privacy is evaluated with an entropy-based confusion metric of an adversary during a location inference attack. The results show that our mechanism provides the best utility/privacy tradeoff and a location prediction accuracy of 60% in average for our model.

5.1 Introduction

In recent years, predicting future locations of users has become an attractive topic for both the research community and companies. Location prediction can boost the creation of new Location-Based Services (LBS) in order to help users in their daily activities. For example, a LBS could send personalized information to users, such as the menu of different restaurants the users could like in the vicinity of a location in which they will probably be at a specific time, e.g., Monday between 11:30 am



(a) First architecture

(b) Second architecture

(c) Third architecture including ResPred system

Fig. 5.1: Problem/contribution overview through three different system architectures

and 12:00 pm. In order to obtain future locations of a user, a LBS needs to build a predictive model containing spatial and temporal information. However, this leads to a first location privacy issue because the user must send all her raw locations to a third-party entity as described in Figure 5.1 (a). In this architecture, the LBS, which can be a possible adversary, is installed on the mobile device of the user and gathers all user locations. To preserve location privacy, the idea is to create a location predictive model in a trusted component that can be stored at the operating system level of the mobile device. In this context, the trusted component itself will provide the future locations of the user to the LBS as depicted in Figure 5.1 (b). Even after a large number of requests performed by the LBS, it should not be able to reconstruct the entire predictive model of the user but may have a good partial view of her model. As a result, this is a undeniable second location privacy issue. It has been demonstrated in the literature that sharing accurate locations has a real cost for a user because a potential adversary cannot only discover a lot of sensitive information related to the user but also identify her by just performing simple location attacks as described by Krumm in [13]. In addition, the authors of [19] show that a few number of user's locations only might highly compromise the location privacy of a user.

Because of the availability of different positioning systems on mobile devices, LBS are very convenient for daily activities. Consequently, users cannot completely avoid using LBS. However, users must know that it is fundamental to preserve their privacy when they are using LBS. Currently, users can only enable or disable the access to locations for specific applications and sometimes reduce the precision of the locations obtained with a positioning system. These options depend on the operating system itself. These simple choices are not adapted to the context of our work because we want to preserve the location privacy of the user at a higher level, which is a location prediction level. In order to protect raw locations of a user, some existing Location Privacy Preserving Mechanisms (LPPMs) can be applied, such as spatial perturbation, spatial cloaking, sending dummy locations as well as spatial rounding, as discussed in [1, 6, 9, 12, 13]. Nevertheless, these mechanisms may quickly decrease the utility level of a LBS as the level of protection increases, up to the point when the LBS becomes unusable.

In this paper, we present a privacy preserving location prediction system called *ResPred*, *res* and *pred* mean respect (i.e., respect the privacy of users) and prediction respectively. This system allows LBS to request future location of users. For instance, a LBS can display information containing future public transportation departures located in the vicinity of the predicted location returned by *ResPred* on the mobile device of the user in advance. Figure 5.1 (c) presents the *ResPred* system that contains two components: one component focuses on the location prediction and the second on the location privacy. We assume that the *ResPred* system is created at the operating system level of the mobile device and that the *ResPred* system and the positioning system are trusted. The system includes a location prediction component based on a statistical location trend model and a location privacy component helping to blur the predicted locations by finding an appropriate tradeoff between the LBS utility and the user privacy preference expressed as a maximum percentage of utility loss. We also assume that the LBS is untrusted, which indicates that it is a possible adversary. As depicted in Figure 5.1 (c), the LBS requests the future location of the user by indicating a time duration between the current time and the time of the desired predicted location and the system returns a predicted location that will be found by exploring the location trend model and protected by our LPPM. The predicted location is more specifically transformed according to the required utility level of the LBS and the maximum utility level that the user is willing to sacrifice in order to protect her location privacy. We evaluate our system from a utility/privacy perspective, which is the crucial aspect of our approach. In addition, we compute the location prediction accuracy of the location trend model. We chose real mobility traces coming from two datasets, the *PrivaMov* dataset described in [3] and a private dataset collected by a researcher in Switzerland. The first part of the utility/privacy evaluation consists in assessing the utility level of our LPPM and two other well-known mechanisms described in the literature, namely the rounding and the Gaussian perturbation. The second part of the utility/privacy evaluation focuses on the measurement of the confusion level of an adversary performing a location attack on the received predicted locations from the *ResPred* system. The metric used to evaluate this confusion level is based on the well-known Shannon entropy. The results show that our location privacy preserving mechanism provides the best utility/privacy tradeoff compared to the other evaluated mechanisms as well as a good location prediction accuracy for the analyzed users. The contributions of this paper are listed below.

- We describe a system, called *ResPred*, allowing LBS to request future location of a user.
- We present a statistical model containing location trends of a user per time slice, helping to extract short, mid and long-term predicted locations.
- We describe a LPPM enabling to reach an appropriate utility/privacy tradeoff.

- We use real user locations to assess the performance of our system and, more specifically, its two components.

The paper is organized as follows: in Section 5.2 we begin with the description of the system model containing the formal definitions used in the paper. Section 5.3 presents the problem addressed in this paper, while the *ResPred* system is described in Section 5.4. Then, we present the evaluation of the system from a utility/privacy perspective in Section 5.5. In addition, we also evaluate the location prediction accuracy of the location trend model of the *ResPred* system. We detail the closest work to the two main subjects of this paper in Section 5.6, which are the location privacy as well as the location prediction. Finally, we highlight the most important findings of the paper and discuss future work in Section 5.7.

5.2 System model

This section focuses on describing the key definitions used to present our system. In order to facilitate the analysis of locations of a user, the time is discretized. We also introduce Regions Of Interest (ROIs) on which the location predictive model is based. Finally, we present the threat model that describes the context used to evaluate the location privacy.

5.2.1 User and locations

We consider that a user moves on a geodesic space and owns a mobile device that is able to detect her locations as well as when they are captured via a positioning system, e.g., GPS, WiFi or radio cells. A location is described as a triplet $loc = (\phi, \lambda, t)$ where ϕ and λ are the latitude and longitude of the location in the geodesic space, and t is the time when the location was obtained from the positioning system. Locations are formally represented as a sequence $L = \langle loc_1, loc_2, \cdots, loc_n \rangle$. A subsequence of successive location of L is described as follows $l_{sub_i} = \langle loc_1, loc_2, \cdots, loc_m \rangle$ in which the first location of this subsequence is noted $l_{sub_i}.loc_{first}$ and the last location is $l_{sub_i}.loc_{last}$. We can express the latitude, longitude and time of a location loc_i by directly writing $loc_i.\phi$, $loc_i.\lambda$ and $loc_i.t$ respectively.

5.2.2 Temporal discretization

In order to discretize time, we compute n slices generated according to the chosen temporal granularity and time span, e.g., every 20 minutes during one week. A time slice is a triplet defined as follows $ts = (t_{starting}, t_{ending}, index)$ where $t_{starting}$ (Monday - 7:00 am) and t_{ending} (Monday - 7:20 am) represent the starting time and ending time of the time slice and *index* is its unique identifier ranging between 1 and n (n represents the total number of computed time slices). For instance, if we generate all time slices having a duration of 20 minutes during a period of 1 week, we will obtain 504 time slices. All the possible time slices are represented as a sequence called *timeslices*, such that $timeslices = \langle ts_1, ts_2, \cdots, ts_n \rangle$. In addition, we introduce a function called $convert(\langle loc_1, loc_2, \cdots, loc_m \rangle)$ translating a sequence of one or several successive locations into a sequence of one or several successive time slices called $timesliceTab$, m being the total number of location(s) to convert. This sequence is described as follows: $timesliceTab = \langle ts_1, ts_2, \cdots, ts_n \rangle$ in which n is the total number of successive time slices.

5.2.3 Regions of interest

A region of interest (ROI) is defined as a circular area visited by a user during a certain period of time, which is a quadruplet of the form $roi = (\phi, \lambda, \varDelta r, visits)$. Items ϕ and λ are the coordinates of the center of the ROI in a geodesic space. $\varDelta r$ is the radius of the ROI and *visits* is a sequence of subsequences of L such that $visits = \langle l_{sub_1}, l_{sub_2}, \cdots, l_{sub_m} \rangle$ in which each subsequence of successive locations is contained in L such that $\forall l_{sub_i} \in visits, l_{sub_i} \subset L$ and $l_{sub_i}.loc_{last}.t < l_{sub_{i+1}}.loc_{first}.t$. Each visit of a ROI has a duration equal or greater than a threshold, called $\varDelta t_{min}$, such as $\forall l_{sub_i} \in visits, l_{sub_i}.loc_m.t - l_{sub_i}.loc_1.t \geq \varDelta t_{min}$. In addition, all locations of the visits are contained in the ROI spatially described by the first three items of it, i.e., latitude, longitude and radius. The set containing all ROIs of a user is noted as follows: $rois = \{roi_1, roi_2, \cdots, roi_n\}$. The last and important characteristic of the ROI is that there is no spatial intersection between ROIs. This means that, if two ROI candidates intersect during the discovery process of ROIs, they will be merged and a new ROI is created from these two ROI candidates.

5.2.4 Threat model

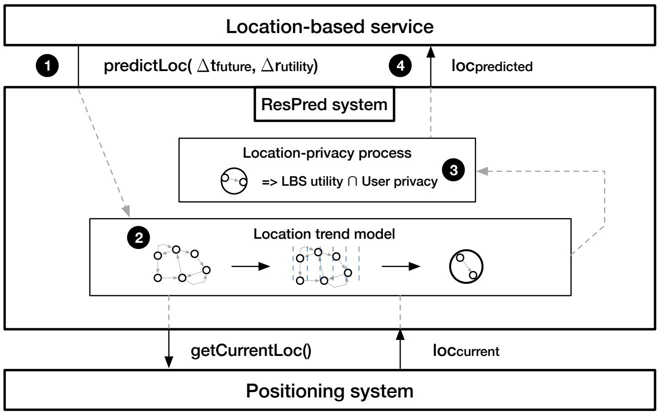
We consider a threat model that takes into account a honest but curious adversary in the form of a LBS using *ResPred*. The LBS will try to infer future locations of the user based on a location history gathered by requesting *ResPred*. This location history contains all predicted locations sent by *ResPred* and consists in its unique background knowledge on which the location attack will be performed. This history is not complete because we consider that the LBS will not request *ResPred* constantly but a limited number of times in a random manner during a certain time slice or by following the usual use of the LBS by the user, e.g., everyday at the end of the afternoon. The honest but curious behavior of the LBS also means that it will not try to break the sharing protocol or obtain the location predictive model of the system *ResPred*. In addition, we consider that the LBS always gives adapted parameters to its service to the *ResPred* system, more specifically the values of the parameters Δt_{future} and $\Delta r_{utility}$ as depicted in Figure 5.1 (c) or Figure 5.2.

5.3 Problem statement

Considering that a LBS wants to estimate the future location of a user, it needs to create a predictive model of the user. In order to reach this goal, the LBS will constantly collect locations of the user to update her model as shown in Figure 5.1 (a). However, the location privacy of user is entirely compromised because all her raw locations are regularly shared with the LBS. This means that all sensitive information related to the user is given to a third-party entity. For example, the LBS can discover the following sensitive information related to the user from her raw locations: her home and work places but also her likes and dislikes about religion and/or politics.

The first solution is to delegate the creation of the predictive model to a service at the operating system level that we consider as trusted, as shown in Figure 5.1 (b). In this context, the only service that has access to the raw locations of the user coming from the positioning system is the dedicated service. The latter provides predicted locations to the LBS that needs them to operate properly. Although the location privacy of the user is increased in this context, there is still a location privacy issue about the predicted locations shared with the LBS. Because of all the predicted locations gathered by the LBS, the latter can always infer precise location habits of the user, especially when it is requesting the trusted service for the same future time every day for instance.

Consequently, the challenge is to protect as much as possible the location privacy of the user in the context of the sharing of her predicted locations with a LBS. Although there exist various LPPMs in the literature, they do not necessarily meet the utility requirement of a LBS. This means that they can easily compromise the proper functioning of the LBS until reaching the point it becomes unusable for the user. For example, the location information provided by the LBS can be inaccurate or simply erroneous because the precision of the prediction has been made too low by the LPPM. As a result, the user might stop using the LBS. As discussed in the introduction, our approach consists in building a system, including a location predictive model as well as an adapted LPPM, that takes into account the utility requirement of the LBS and the utility/privacy tradeoff expressed by the user as indicated in Figure 5.1 (c) or Figure 5.2.



2 $tmpLoc_{predicted} = predictLoc(loc_{current}, \Delta t_{future})$

3 $loc_{predicted} = protect(tmpLoc_{predicted}, \Delta p_{maxUtilityLoss}, \Delta r_{utility})$

Fig. 5.2: ResPred system overview

5.4 System overview

As described in Figure 5.2, *ResPred* contains two components. The first component focuses on location prediction, while the second component relates to location privacy. Consequently, the first component is responsible for the prediction of the future location of the user and includes her predictive location model, called *location trend model.* The second component aims at protecting the predicted location computed by the first component and uses a LPPM called *utility privacy tradeoff LPPM.*

A request of a LBS consists in asking where a user will be in the future. As described in Equation 5.1, the LBS requests the future location by specifying the time duration expressed by Δt_{future} in seconds from the current time, e.g., 7200 seconds (2 hours) from now. The LBS also indicates its required utility $\Delta r_{utility}$ that allows it to operate properly. For instance, if a LBS must call a taxi for a user in advance, the LBS will indicate an utility of a short distance in meters, such as 500 meters. A long distance could compromise the use of the taxi service itself and the related LBS because it could display inaccurate information to the user. The returned value is a location expressed by a pair $loc_{predicted} = (\phi, \lambda)$.

$$loc_{predicted} = predictLoc(\Delta t_{future}, \Delta r_{utility}) \tag{5.1}$$

To summarize, *ResPred* will answer the following question: *Knowing the user will need some location-based information within $\Delta r_{utility}$ meters in Δt_{future} seconds from now, where will be the user?*

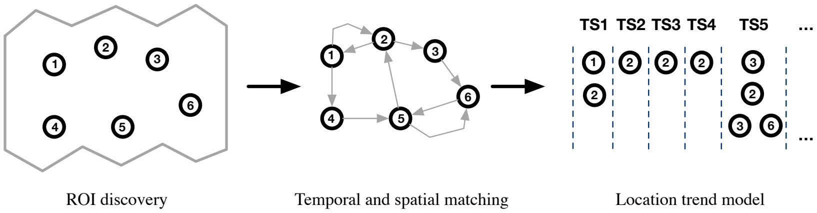


Fig. 5.3: From ROIs to location trend model

5.4.1 Location prediction component

The location prediction component contains a predictive model that represents the location trends of a user organized per time slice. As mentioned in Section 5.2, time is discretized into time slices during a given period of time, such as 504 time slices during one week (i.e., the duration of one time slice is 20 minutes). A location trend model is an array in which each cell contains all the possible ROIs or successive ROIs visited during a specific time slice. Figure 5.3 describes the creation process of the location trend model. Firstly, the *ROI discovery* process enables to discover all the ROIs of a user by analyzing the raw locations of the user. Secondly, all raw locations are marked with a specific ROI and a specific time slice as specified in the *Temporal and spatial matching* step. This step helps to pre-process the locations for the creation of the location trend model. Finally, we discover the structure of the location trend model in which we collect all the ROIs or successive ROIs visited during each time slice. Since the location trend model is a statistical model, each visited ROI or successive visited ROIs stored for a given time slice have a visit counter. This enables to highlight the location habits of the user per time slice, i.e., the ROIs or successive ROIs that are the most visited by the user during a time slice. In addition, this allows the component to find the predicted locations to answer the LBS requests.

As depicted in Figure 5.2, the location trend model will have to solve the following request expressed in Equation 5.2 and return a temporary predicted location $tmpLoc_{predicted}$. The latter is not the final predicted location sent to the LBS at the end of the process because $tmpLoc_{predicted}$ must be protected by the LPPM of the location privacy component.

$$tmpLoc_{predicted} = predictLoc(loc_{current}, \mathit{\Delta} t_{future}) \tag{5.2}$$

In order to find the $tmpLoc_{predicted}$, the location prediction component starts by searching the target time slice corresponding to the time slice that includes the future time computed by adding the $\mathit{\Delta} t_{future}$ duration to the current timestamp, i.e., $loc_{current}.t$. After having found this target time slice, the location trend model is analyzed to find the location trends corresponding to the target time slice expressed as ROI(s). The $tmpLoc_{predicted}$ is a triplet such as $tmpLoc_{predicted} = (\phi, \lambda, \mathit{\Delta} r)$. Item

Δr is a radius that is the accuracy of the temporary predicted location. There are two cases now to compute the items of the $tmpLoc_{predicted}$. Firstly, if the analysis highlights that the most likely visited location in the target time slice corresponds to one ROI, the temporary predicted location has the same latitude, longitude and radius as those of the ROI. Secondly, if the analysis shows that the most likely visited locations are two or several successive ROIs, the component merges all the ROIs into one single ROI and computes a new latitude, a new longitude and a new radius, which correspond to the items of the $tmpLoc_{predicted}$. In addition, it is important to note three specific location prediction scenarios that can occur during the prediction process. The best scenario is that the component finds the most likely ROI or successive ROIs to compute the $tmpLoc_{predicted}$ by exploring the location trends of the target time slice. Secondly, it can happen that all ROIs or successive ROIs have the same visit counter value. In this context, the last visited ROI or successive ROIs are used to compute the $tmpLoc_{predicted}$. Finally, it is also possible that there is no ROI or successive ROIs recorded for the target time slice. For this unique and specific problem, the component explores previous time slices until finding a visited ROI or successive ROIs to compute the $tmpLoc_{predicted}$.

5.4.2 Location privacy component

The goal of the location privacy component is to protect as much as possible the temporary predicted location found by the location prediction component. The LPPM that will be applied on the $tmpLoc_{predicted}$ depends on two aspects: the LBS utility $\Delta r_{utility}$ given by the LBS and the user privacy preference given by the user expressed as a maximum utility loss percentage $\Delta p_{maxUtilityLoss}$. This means that the LBS can provide useful and relevant information in a radius, which is the LBS utility in meters, around a reference location. Beyond this distance, there is no guarantee that the LBS is able to operate properly or to provide a reliable information to the user. For example, if the LBS is an application of a taxi company and asks a predicted location, at the end of day when the user usually requests the LBS for a taxi, in order to anticipate the user's request, the LBS will indicate a close utility in meters in order to not be far from the user in a future time. The maximum utility loss is expressed as a percentage that clearly indicates the maximum utility that the user is willing to sacrifice in order to protect her location privacy. Consequently, its value is a percentage ranged between 0 included and 1 not included. 0 is included and means that the user simply does not want to lose any LBS utility. 1 is not included because this would mean that the LBS cannot work properly if this value is reached. Equation 5.3 describes the request handled by the component including the LBS utility $\Delta r_{utility}$ and the maximum utility loss percentage $\Delta p_{maxUtilityLoss}$.

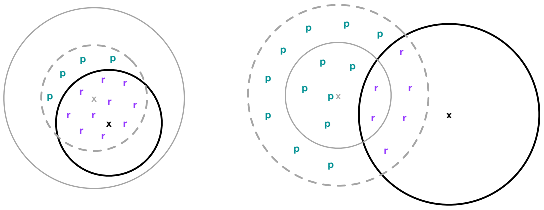


Fig. 5.4: Computing new coordinates when the radius of the reference zone is adjusted, i.e., greater or smaller than the radius of $tmpLoc_{predicted}$

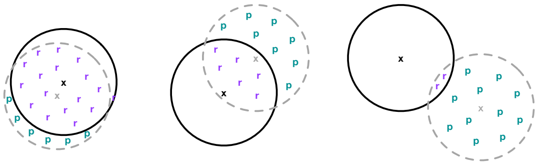


Fig. 5.5: Three possible random generations of new coordinates (position x in black) according to a high maximum utility loss percentage

$$loc_{predicted} = protect(tmpLoc_{predicted}, \Delta p_{maxUtilityLoss}, \Delta r_{utility}) \qquad (5.3)$$

The location privacy preserving mechanism works in the following manner. The component firstly creates a reference zone $zone_{ref}$ that has a latitude and a longitude corresponding to those of the $tmpLoc_{predicted}$ and a radius equals to the LBS utility $\Delta r_{utility}$.

The goal of the component is now to change the latitude and the longitude of the $tmpLoc_{predicted}$ by computing new coordinates. The component will create a new zone, called $zone_{new}$, which is a zone having the new generated latitude and longitude as coordinates and a radius equals to the LBS utility $\Delta r_{utility}$. In order to compute these new coordinates, the component firstly generates a random angle that indicates the direction of the new coordinates. Then, a latitude and a longitude are generated randomly in the direction of the angle between 0 and a threshold value corresponding to the case where there cannot have any intersection between $zone_{ref}$ and $zone_{new}$, i.e., $2 \times \Delta r_{utility}$. Now the component must check if the protected percentage of the $zone_{ref}$ is not greater than the maximum utility loss percentage indicated by the user, i.e., $p_{maxUtilityLoss}$. In order to check this condition, the component computes the area of the intersection between the reference zone $zone_{ref}$ and the new zone $zone_{new}$. The area of this intersection is divided by the area of the $zone_{ref}$ in order to obtain a revealed percentage $p_{revealed}$, which is shared with the LBS. Finally, the component computes the protected percentage that is equal to: $p_{protected} = 1 - p_{revealed}$. The new coordinates are validated only if $p_{protected}$ is lower or equal to the maximum utility loss percentage given by the user. If it is not the case, new coordinates are generated until meeting this condition. When this condition is met, $loc_{predicted}$ is created with a latitude and a longitude corresponding to the new coordinates and is sent to the

LBS. Therefore, there is a clear link between the utility that the user is willing to lose and her location privacy because the greater the $p_{maxUtilityLoss}$, the better the user protects her location privacy. Equation 5.4 summarizes the checking of this condition. The function *area* enables to compute the area of the elements passed as parameters.

$$1 - \frac{area(zone_{ref} \cap zone_{new})}{area(zone_{ref})} \leq \Delta p_{maxUtilityLoss} \tag{5.4}$$

This means that the location privacy component tries to find an appropriate tradeoff between LBS utility and the location privacy preference chosen by the user. In order to illustrate the process, Figure 5.4 depicts the impact of the change of the radius of the reference $zone_{ref}$ in the case where the $tmpLoc_{predicted}$ has a radius greater than $\Delta r_{utility}$ and in the case where $tmpLoc_{predicted}$ has a radius smaller than $\Delta r_{utility}$. The $tmpLoc_{predicted}$ is the gray circle, the $zone_{ref}$ is the gray circle with the dotted lines and the dark circle is the $zone_{new}$. The center of the $zone_{new}$ corresponds to the location that is sent to the LBS by *ResPred*. The letters r indicate the zone that is revealed to the LBS, while the letters p describe the zone that is protected. In addition, Figure 5.5 depicts three possible random generations of new coordinates according to a maximum utility loss percentage that is really high. The resulting value of the process of this component is the predicted location $loc_{predicted}$, which is also returned to the LBS as described in Equation 5.3 and in Figure 5.2.

5.5 Evaluation

The main goal of the evaluation is to assess our system from a utility and a location privacy perspective. In order to reach this goal, we ran several experiments taking into account different LBS scenarios and different LPPMs including our mechanism and existing ones. In addition, we also compute the location prediction accuracy of the location trend model.

5.5.1 Dataset

We chose real user locations of two datasets: *PrivaMov* dataset described in [3] and a very detailed dataset of one user. From these two datasets, we extracted locations of users that were captured via different positioning system such as GPS, radio cells as well as WiFi. We performed an analysis in order to select the best users of *PrivaMov* for our evaluation. This selection was based on the quality of the user datasets. This quality was assessed by computing the percentage of hours during one day having at least one location, called *daily percentage* later. In order to properly fill the location trend model, we need very rich user datasets without important gaps in terms of days. More specifically, a user is selected if the average of all her daily percentages is greater or equal to 0.4, if her dataset duration in terms of days was greater or equal to 30 days and lower than 250 days and if all weekdays (from Monday to Sunday) have at least one daily percentage. Seven users only of the *PrivaMov* dataset met all these conditions. Consequently, we evaluated eight users in total, i.e., seven users from *PrivaMov* and one user from the private dataset. The average of the duration of all evaluated user datasets is 115 days and the average of the number of locations of all evaluated user datasets is 7'580'391.

5.5.2 LBS scenarios

We decided to define two LBS scenarios for the evaluation. The first scenario is a public transportation LBS that provides next departure information of bus, metro and train in advance. The information is displayed on the mobile of the user just before the usual checking of the public transportation departures by the user. The second scenario is a taxi LBS that calls a taxi in advance for the user by following the usual use of the service by the user. We assume that the LBS knows the usage habits of the service by the user but it does not obviously know the location of the user in the future. That is why these two LBS must use our *ResPred* system to obtain it. As depicted in Figure 5.2, an LBS can request a user's location in the future, e.g., 2 hours from now. For each scenario, we chose an array of target time slices for which the predicted locations must be computed. The parameters of these two LBS scenarios are defined in Section 5.5.4.

5.5.3 Existing location privacy preserving mechanisms

In order to properly assess the LPPM of our system, we selected two existing LPPMs from the literature. We compare them with our mechanism from the utility/privacy perspective whose metrics are detailed in Section 5.5.6. We chose the spatial rounding presented in [1, 13] as well as the Gaussian perturbation described in [2]. The spatial rounding works with a grid that discretizes the space in which the user is moving. The mechanism transforms raw coordinates of a location into new coordinates corresponding to the nearest vertex of the square or rectangle that is a grid's cell in which the raw location is. The spatial Gaussian perturbation is a mechanism that adds spatial noise to the latitude and the longitude of a raw location according to a certain mean and a standard deviation. All these parameters are presented in the next section.

Parameter/LBS scenario	Public transportation LBS scenario	Taxi LBS scenario
LBS utility distance	1000 meters	500 meters
Number of target time slices	10	4
Number of prediction requests	100	100
Time slice distribution of prediction requests	Randomly or evenly distributed	

Table 5.1: List of parameters used for each LBS scenario

5.5.4 Experimental settings

The experimental settings of the utility/privacy evaluation as well as location prediction accuracy evaluation are detailed in this section.

5.5.4.1 ROI discovery

In order to discover the ROIs of a user, we use a specific part of a discovery process of Zones of Interest (ZOIs) described in [14]. The Δd_{max} is equal to 60 meters and Δt_{min} has a value of 10 minutes. We follow the creation process of clusters, which are called ROIs in this paper, without creating any cluster groups or ZOIs similarly to [14]. After discovered all clusters, we merge them if an intersection occurs between two clusters and we repeat this merging process until reaching a stable cluster set in which there is no more intersection. We do not filter out the ROIs that are not frequently and/or not recently visited because we want to keep a high number of ROIs describing the mobility of the user, in order to properly fill the location trend model.

5.5.4.2 Location trend model

The location trend model is created with time slices having a duration of 20 minutes during a period of one week, resulting in 504 time slices for one week. We chose this time slice duration by exploring the entropy level of each time slice cell of the location trend model and finding that it was the best time slice duration for the location prediction goal. We used the entropy level because it highlights the level of uncertainty to find one single ROI for a time slice cell.

5.5.4.3 LBS scenarios

A scenario corresponds to a specific type of LBS as described in Table 5.1: the public transportation LBS and the taxi LBS. Firstly, each LBS has its own utility distance and a specific number of target time slices. For example, the target time slices of the public transportation LBS are in the morning, i.e., from 7:00 to 7:20 am, and at the end of the afternoon, i.e., from 5:00 to 5:20 pm, every working day. Regarding the taxi LBS, the target time slice are in the evening Thursday from 10:00 to 10:20 pm, Friday from 11:00 to 11:20 pm, and Saturday from 4:00 to 4:20 pm and from 11:00 to 11:20 pm. We distribute the total number of location prediction requests, called *frequency* in Table 5.1, which are 100 in total, per target time slice for each LBS scenario. The distribution of the total number of location prediction requests can be equal for all the target time slices, i.e., 10 (100 divided by 10) for each target time slice. Or the process can also randomly distribute the 100 predicted location requests per target time slice meaning that some time slices can have more predicted locations than others. These scenarios are the same for all evaluated users.

5.5.4.4 Location privacy preserving mechanisms

As mentioned previously, we chose two LPPMs: the grid-based rounding and the Gaussian perturbation in addition to our proposed LPPM. For the *utility privacy tradeoff LPPM* included in *ResPred*, we selected four values for $\Delta p_{maxUtilityLoss}$: 0.2, 0.4, 0.6 and 0.8. Regarding the grid-based rounding, we decided to have a difference of 0.005, 0.05 and 0.5 between two successive latitudes or longitudes to create each cell of the grid. The values of the grid-based parameters are ranged from approximately 380 to 38000 meters. Finally, we chose 4 standard deviations that are 0.0005, 0.005, 0.05 and 0.5 for the Gaussian perturbation, the mean being the latitude or the longitude of the raw location. The values of the Gaussian perturbation parameters are ranged from approximately 55 to 66500 meters.

5.5.5 Location prediction accuracy

We decide to evaluate the location prediction accuracy of the location trend model by performing the following steps. Firstly, the dataset of each user must be divided into two datasets according to the total number of locations: a training set of 60% and a test set of 40%. We discover the ROIs and we create the location trend model of a user with her training dataset. Secondly, the evaluation process is the following: we start by selecting 200 unique locations in the test set. For each selected location of the test set, we convert its timestamp into a target time slice. Then, we find the most likely visited ROI or successive ROIs of the target time slice by exploring the location trend model, which is the same process explained in Section 5.4.1. We consider that the prediction is correct for this location only if the latter is contained in the ROI or the merged ROI computed from the successive found ROIs. If there is no ROI for the target time slice, we simply do not take the prediction into account. At the end, we compute a ratio that is the number of correct predictions out of the number of predictions that returned a value after having explored the model.

5.5.6 Utility/privacy metrics

The metrics presented in this section enables to evaluate the utility as well as the location privacy of all predicted locations of a user shared with a LBS and, consequently, to highlight the LPPM that gives the best utility/privacy tradeoff.

5.5.6.1 Utility metric

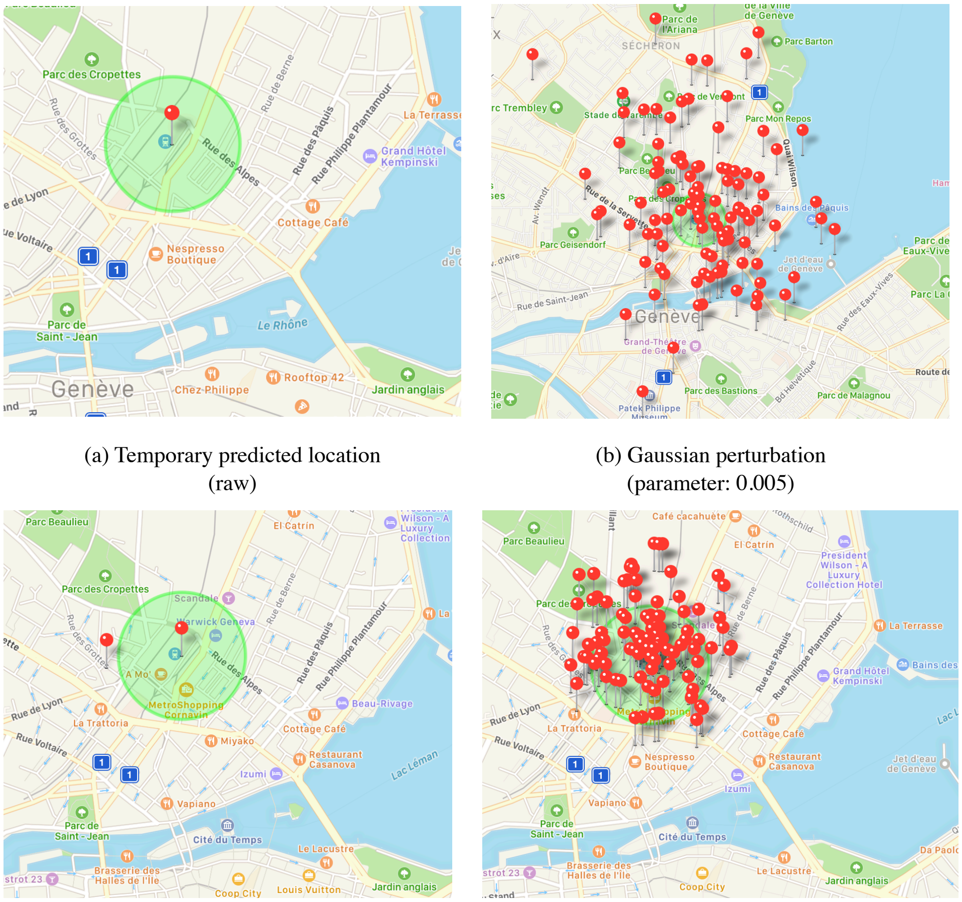
The utility metric allows us to evaluate if a predicted location sent to the LBS meets the utility requirement of the LBS given at the beginning of the process by the LBS itself. We define a reference zone $zone_{ref}$ that has a center corresponding to the center of the $tmpLoc_{predicted}$ and a radius that has the value of $\varDelta r_{utility}$. We also create a zone to check $zone_{toCheck}$ having a center that is the coordinates of the predicted zone $loc_{predicted}$ and a radius equals to the value of $\varDelta r_{utility}$. The utility is validated if there is an intersection between the $zone_{ref}$ and the zone to check $zone_{toCheck}$. Equation 5.5 describes the two possible results of the utility metric.

$$res_{utility} = \begin{cases} 0, & \text{if } zone_{ref} \cap zone_{toCheck} = \emptyset \\ 1, & otherwise \end{cases} \tag{5.5}$$

Then we compute the utility average of a target time slice by dividing the number of predicted locations that meet the utility condition by the total number of predicted locations sent to the LBS for this target time slice. Finally, we calculate the average of the utility results obtained for all the target time slices in order to obtain the utility result of the scenario.

5.5.6.2 Location privacy metric

The location privacy metric corresponds to a metric that evaluates the degree of confusion of an adversary, the LBS in our case, during a location attack on the predicted locations received from *ResPred*. The metric is based on the Shannon entropy that can compute a level of uncertainty as described in [16]. As mentioned in Section 5.2.4, the location attack performed by an adversary consists in trying to discover one location amongst all of the predicted locations sent by *ResPred* for a specific target time slice considering that the adversary knows how the time is discretized in our location trend model. The goal of a LPPM is to confuse the adversary in order to reduce its probability of finding one single location for a target time slice. In order to compute the location privacy, we will create a grid that discretizes the space and compute the density proportion $p_{density}$ of each visited cell of this grid. The density proportion $p_{density}$ is the number of predicted locations out of the total number of predicted locations visited per visited cell of the grid during



(a) Temporary predicted location (raw)

(b) Gaussian perturbation (parameter: 0.005)

(c) Rounding mechanism (rounded to 2 decimals)

(d) Utility privacy tradeoff LPPM (parameter: 0.9)

Fig. 5.6: Visual description of the impact of the different LPPMs on a temporary predicted location

the target time slice. Each cell of the grid is a rectangle of approximately 100 meters per 180 on an average, i.e., a difference of 0.001 between two successive latitudes or longitudes. Equation 5.6 describes the computation of the location privacy for a specific target time slice in which i is the index of the i^{th} visited cell by the user, n is the total number of visited cells by the user during the target time slice. A low entropy result means a low confusion of the adversary, while a high entropy result means a high uncertainty.

$$res_{locationPrivacy} = -\sum_{i=1}^{n} p_{density_i} \log_2 p_{density_i} \tag{5.6}$$

Finally, we compute an average result for each scenario in the same way as for the utility metric (described at the end of the previous section).

LPPM/Result	Utility result	Location privacy result
Utility privacy tradeoff LPPM	**1.0**	**2.81**
Grid-based rounding	0.62	0
Gaussian perturbation	0.50	2.78

Table 5.2: Utility / location privacy results

5.5.7 Results

The average of the location prediction accuracy of the location trend model for all evaluated users is equal to 60%. In addition, we obtain a minimum and a maximum location prediction accuracy of 16% and 90% respectively.

Regarding the utility/privacy tradeoff evaluation, we firstly compute the average of the utility results of all LBS scenarios per user and, secondly, we calculate average utility results of all users. We do exactly the same for the location privacy results. The results are summarized in Table 5.2. We can clearly see that our LPPM, i.e, the utility/privacy tradeoff LPPM, has the best utility/privacy tradeoff because the utility result and the location privacy result reach the highest values. This means that our LPPM meets the LBS utility requirements and is also able to protect the location privacy of the user according to her privacy preference. Although the Gaussian perturbation has also a high location privacy result, a reasonable utility result is not reached. The location privacy result of the grid-based rounding is equal to 0 indicating that the adversary has no confusion because the modified locations, i.e., predicted locations, are always the same for a target time slice. The Gaussian perturbation has the advantage of blurring the location via a single parameter expressing a distance, while the grid-based mechanism requires the creation of a grid that can take a substantial time and its exploration before being able to blur a location. Although our mechanism must check a location privacy condition, it computes the new coordinates within a reasonable time.

Finally, we can see the blurring impact of the different LPPMs on a temporary predicted location in Figure 5.6. In Figure 5.6 (a), we can see the center as well as the radius of a temporary predicted location, both depicted with a marker and a circle. In Figure 5.6 (b) and (d), 100 new locations, depicted with new markers, are created according to the corresponding LPPM. Regarding the rounding, the coordinates have only been rounded to two decimals in the figure but in the context of the evaluation with a spatial grid, we would have obtained 100 times the same location because the structure of the grid is fixed and the nearest location is always the same for a single location to blur.

5.6 Related work

The related work below tackles the two main subjects of the paper that are the following: the description of existing LPPMs as well as the different predictive models presented in the literature that are used to compute future user locations.

5.6.1 Location privacy preserving mechanisms

In a location prediction context, we consider that we need to protect the predicted location that is sent to a LBS as mentioned in Section 5.3. To reach this goal, there exist various mechanisms to protect the predicted location, such as applying a spatial perturbation [1, 2, 6], using a spatial cloaking mechanism [9], sending dummy locations [12] or using a rounding mechanism [1, 13].

Applying a spatial perturbation enables to spatially modify a location as mentioned by several authors in [2, 6]. As described in these papers, we can add spatial noise to the coordinates of a location. However, the more noise is added to the location sent to the LBS increases, the more the LBS utility decreases in our context because a LBS may provide information that is not related to the raw predicted location, depending on the level of protection. In the case of the spatial cloaking presented by Gruteser and Grunwald in [9], the predicted location should only be sent if the user is considered as *k-anonymous*, meaning that the user cannot be distinguishable from at least $k-1$ other users. This technique is unfortunately not realistic in our context and not easy to implement especially in the case where the mobility models of users are not centralized or shared in a common server. As detailed in [12], sending dummy locations is interesting in order to add noise if and only if multiple predicted locations can be sent to a LBS. However in our system, it is impossible to use this LPPM because only one predicted location must be sent to a LBS as an answer to a predictive request supported by *ResPred*. Utilizing a rounding mechanism, as described in [1, 13], can be considered because the predicted location is changed into a new location corresponding to a nearest reference point. If we consider that space is discretized and described with multiple reference points (the vertices of each cell of a grid for instance), the mechanism consists in modifying a location into a new location corresponding to the nearest reference vertex of the cell in which the location is as indicated in the papers cited previously. Cryptography techniques could be also used to protect locations sent to third parties as mentioned in [10] but our work is not focused on this kind of privacy/security strategies. To summarize and according to the best of our knowledge, there is no LPPM that can find an appropriate tradeoff between the utility and the privacy in a location prediction context. For our utility/privacy evaluation, we chose the closest

LPPMs to our work, that are the rounding and the spatial perturbation as detailed in the previous section.

5.6.2 Location prediction requests and models

As detailed in the complete survey in [10] or in [5], various techniques exist to predict future locations of users. In the literature, there exist different location predictive models for different types of location prediction requests, such as predicting a future location based on a time duration [11, 15], predicting the next location that will probably be reached by a user [7, 8, 18], etc. Some location prediction-based papers focus on other location-based predictive requests, such as the prediction of the staying time in a particular ROI or when the user will reach or leave a ROI [8], the prediction of the number of users reaching a specific zone [4] and much more. Other remaining works are focused on range queries that enable to identify if one or multiple user(s) will be in a specific area during a specific time window. In [17], the authors describe a way to prune an order-k Markov chain model in order to efficiently compute long-term predictive range queries.

The main focus of our paper, in terms of prediction, is to comnpute a future location of a user based on a time duration from the current time. In the literature, it is shown that some predictive models can work better for near location predictions and others are more suited for distant location predictions. In [11], the authors present a hybrid prediction model for moving objects. For near location predictions, their model uses motion functions, while for distant location predictions, their model computes the predicted location based on trajectory patterns. The structure in which they store the trajectory patterns of a user is a trajectory pattern tree. However, they do not evaluate their model with real mobility traces. Their predictive model is close to our location trend model because they use the notion of patterns based on spatial clusters to fill their model. Nevertheless, the structure of their final model is clearly not the same as ours because they create a trajectory pattern tree. Sadilek and Krumm propose a method to predict long-term human mobility in [15] up to several days in the future. Their method, which can highlight strong pattern of users, uses a projected eigendays model that is carefully created by analyzing the periodicity of the mobility of a user as well as other mobility features. This work highlights that it is crucial to extract strong patterns for long-term predictions. The location trend model we propose in the *ResPred* system is close to the model presented by Sadilek and Krumm. However, our model is different in that it is based on ROIs and not on raw locations and takes less features into account.

5.7 Conclusion

In this paper, we presented a system called *ResPred* that enables to compute predicted locations of a user for LBS. This system contains two components. The first component focuses on location prediction by including a predictive model based on location trends expressed as ROI(s). The second component aims at protecting the location privacy of the user by finding an appropriate tradeoff between a utility specified by the LBS and a location privacy preference indicated by the user that is expressed as a maximum utility loss percentage. The results clearly show that our LPPM provides the best utility/location privacy tradeoff compared to two other existing LPPMs. In addition, the location trend model is promising if we look at the location prediction accuracy results, especially in the context of location prediction according to a certain time duration in the future. Future work will consist in extending the evaluation to more users by finding a dataset having rich user datasets, which is a real need for the research community. We will also design other inference attacks in order to evaluate the location privacy and maybe compare the computing cost of the different LPPMs. And finally, we will compare the location trend model to other existing close models for similar requests regarding short, mid and long-term location predictions.

References

[1] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *ACM Sigmod Record*, volume 29, pages 439–450. ACM, 2000.

[2] Marc P. Armstrong, Gerard Rushton, and Dale L. Zimmerman. Geographically masking health data to preserve confidentiality. *Statistics in Medicine*, vol. 18:497–525, April 1999.

[3] Sonia Ben Mokhtar, Antoine Boutet, Louafi Bouzouina, Patrick Bonnel, Olivier Brette, Lionel Brunie, Mathieu Cunche, Stephane D 'alu, Vincent Primault, Patrice Raveneau, Herve Rivano, and Razvan Stanica. PRIVA'MOV: Analysing Human Mobility Through Multi-Sensor Datasets. In *NetMob 2017*, Milan, Italy, April 2017.

[4] Bertil Chapuis, Arielle Moro, Vaibhav Kulkarni, and Benoît Garbinato. Capturing complex behaviour for predicting distant future trajectories. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 64–73. ACM, 2016.

[5] Vincent Etter, Mohamed Kafsi, Ehsan Kazemi, Matthias Grossglauser, and Patrick Thiran. Where to go from here? mobility prediction from instantaneous information. *Pervasive and Mobile Computing*, 9(6):784 – 797, 2013. Mobile Data Challenge.

[6] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. Show me how you move and i will tell you who you are. *Trans. Data Privacy*, 4(2):103–126, August 2011.

[7] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Nuñez Del Prado Cortez. Next place prediction using mobility Markov chains. In *MPM - EuroSys 2012 Workshop on Measurement, Privacy, and Mobility - 2012*, Bern, Switzerland, April 2012.

[8] Győző Gidófalvi and Fang Dong. When and where next: individual mobility prediction. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, pages 57–64. ACM, 2012.

[9] Marco Gruteser and Dirk 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. ACM, 2003.

[10] Abdeltawab M. Hendawi and Mohamed F. Mokbel. Predictive spatio-temporal queries: A comprehensive survey and future directions. In *Proceedings of the First ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, MobiGIS '12, pages 97–104, New York, NY, USA, 2012. ACM.

[11] Hoyoung Jeung, Qing Liu, Heng Tao Shen, and Xiaofang Zhou. A hybrid prediction model for moving objects. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 70–79. Ieee, 2008.

[12] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. An anonymous communication technique using dummies for location-based services. In *Proceedings of the International Conference on Pervasive Services 2005, ICPS '05, Santorini, Greece, July 11-14, 2005*, pages 88–97, 2005.

[13] John Krumm. Inference attacks on location tracks. In *Proceedings of the 5th International Conference on Pervasive Computing*, PERVASIVE'07, pages 127–143, Berlin, Heidelberg, 2007. Springer-Verlag.

[14] Vaibhav Kulkarni, Arielle Moro, and Benoît Garbinato. A mobility prediction system leveraging realtime location data streams: poster. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 430–432. ACM, 2016.

[15] Adam Sadilek and John Krumm. Far out: Predicting long-term human mobility. In *AAAI*, 2012.

[16] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. Quantifying location privacy. In *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, pages 247–262, Washington, DC, USA, 2011. IEEE Computer Society.

[17] Xiaofeng Xu, Li Xiong, Vaidy Sunderam, and Yonghui Xiao. A markov chain based pruning method for predictive range queries. In *Proceedings of the 24th*

ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '16, pages 16:1–16:10, New York, NY, USA, 2016. ACM.

[18] Josh Jia-Ching Ying, Wang-Chien Lee, Tz-Chiao Weng, and Vincent S Tseng. Semantic trajectory mining for location prediction. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 34–43. ACM, 2011.

[19] Hui Zang and Jean Bolot. Anonymization of location data does not work: A large-scale measurement study. In *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, MobiCom '11, pages 145–156, New York, NY, USA, 2011. ACM.

Part III
Mobility Behavior Analysis

Chapter 6

Analyzing Privacy-aware Mobility Behavior Using the Evolution of Spatio-temporal Entropy

Submitted to a Journal in May 2018
Knowledge and Information Systems (KAIS)
Publisher: Springer
Under review

Abstract Analyzing mobility behavior of users is extremely useful to create or improve existing services. Several research works have been done in order to study mobility behavior of users that mainly use users' significant locations. However, these existing analysis are extremely intrusive because they require the knowledge of the frequently visited places of users, which thus makes it fairly easy to identify them. Consequently, in this paper, we present a privacy-aware methodology to analyze mobility behavior of users. We firstly propose a new metric based on the well-known Shannon entropy, called spatio-temporal entropy, to quantify the mobility level of a user during a time window. Then, we compute a sequence of spatio-temporal entropy from the location history of the user that expresses user's movements as rhythms. We secondly present how to study the effects of several groups of additional variables on the evolution of the spatio-temporal entropy of a user, such as spatio-temporal, demographic and mean of transportation variables. For this, we use Generalized Additive Models (GAMs). The main strength of GAMs is that they are not only interesting to predict a response variable, but also to understand the effects of co-variables on this response variable. The results firstly show that the spatio-temporal entropy and GAMs are an ideal combination to understand mobility behavior of an individual user or a group of users. We also evaluate the prediction accuracy of a global GAM compared to individual GAMs and individual AutoRegressive Integrated Moving Average (ARIMA) models. These last results highlighted that the global GAM gives more accurate predictions of spatio-temporal entropy by checking the Mean Absolute Error (MAE). In addition, this research work opens various threads, such as the prediction of demographic data of users or the creation of personalized mobility prediction models by using movement rhythm characteristics of a user.

6.1 Introduction

Studying mobility behavior of users is a key element to understand how they usually move and what are the events that can influence their behavior. We can study mobility trends from a user point of view with her own movements or from a population point of view with multiple users' movements. This mobility analysis can be useful for several goals: for example, studying the movements of a population through a city to adapt transportation modes and mobility paths. It can also be very helpful to predict mobility behavior of a user from other users' movements having common characteristics. However, studying mobility behavior can be extremely intrusive and lead to a location privacy issue.

In the literature, mobility behavior has been analyzed by computing user daily patterns amongst her locations that obviously reveal her most significant places as described in [3, 4, 9, 12]. For example, the most common places studied in mobility analysis can be home and work places because users usually spend a large amount of time in them. However, such analysis is highly intrusive because it is highlighting the most significant places of a user, i.e., where she spends most of her time. Consequently, it should be possible to find the identity of a user from these significant places. For example, in [2], De Montjoye et al. demonstrate that only 4 spatio-temporal points are sufficient to uniquely identify 95% of the users of the dataset they used for their research. All these findings indicate that it is crucial to find a new methodology to analyse the mobility behavior of a user or multiple users by only analyzing their rhythm, without necessarily studying their significant visited locations.

In order to reach the goal of this mobility behavior analysis that takes care of the location privacy of users, we propose a new methodology based on the extraction of movement rhythm of a user from her location history. To do so, we compute a spatio-temporal entropy sequence that describes the mobility of the user during several time windows of the same duration, e.g., one hour. Then, we analyze the effects of different variables on this rhythm to better understand the way the user moves. These variables can belong to various categories and are strongly linked to the availability of data of the chosen dataset we use: spatio-temporal variables (e.g., day of the week, hour of the day, max distance travelled) and demographic variables (e.g., gender, age-group, working-profile, sport exercise frequency). We use a dataset, called *Breadcrumbs*, that has been collected in the Lake Geneva region (Switzerland) at the Lausanne campus. We choose Generalized Additive Models (GAMs) as a support to understand a mobility behavior because they are not only interesting to predict a response variable, but also to understand what kind of co-variables can affect this response variable. In our context, we obviously use the spatio-temporal entropy as the response variable of the GAM because its value reflects the mobility level of a user during a specific time window. Hence, the sequence of spatio-temporal entropy of a user describes her movement rhythms.

In order to evaluate the prediction accuracy of the GAMs for our research analysis, we train one global GAM with 60% of the spatio-temporal entropy sequence of all the users of the dataset, and check the prediction obtained for the 40% remaining sequence of all of them. We individual AutoRegressive Integrated Moving Average (ARIMA) models for each user with 60% of their dataset and compare the Mean Absolute Errors (MAEs) and the Root Mean Squared Errors (RMSEs) obtained of each user for these two different models. We also compare the MAEs and the RMSEs obtained for individual GAMs compared to those resulting of the use of the global GAM. We observe that there is not a significant difference between individual GAMs and the global GAM in terms of accuracy and that the global GAM is more accurate than the individual ARIMA models if we compare the MAE results. In addition, the results show that it is possible to observe the effect of variables on the mobility of a user or a group of users in a privacy-aware manner by using the spatio-temporal entropy only as a response variable.

This research work has two main contributions:

(1) Proposing a new methodology to analyze mobility behavior of users in a privacy preserving manner;
(2) Describing a new entropy-based metric to quantify the level of mobility of a user that expresses users' movements as rhythms.

The roadmap of this paper is the following. In Section 6.2, we present the overall privacy-aware methodology to analyze mobility behaviors. Then, in Section 6.3, we present how we build the spatio-temporal entropy sequence of a user from her location history. In Section 6.4, we describe the entire analysis of the mobility using GAMs, ARIMA models and the results obtained with these two types of predictive models. Section 6.5 details the related work. And finally, we conclude the paper in Section 6.6 and present future work.

6.2 Proposed methodology

As mentioned in the introduction, the goal of this research work is to analyse the mobility behavior of a user or a group of users in a location privacy preserving manner, i.e., without extracting sensitive places frequently visited by a user, such as her home place, work place and favorite other places. In order to be more precise, we want to analyze the mobility behavior of the users according to variables that can affect it.

The first challenge is to convert the location history of a user into a rhythm. This latter must describe the evolution of the level of mobility of a user over time. To do so, we create a metric based on the well-known Shannon entropy in order to quantify the level of mobility of the user at a certain time and through space. This crucial step must be done for the entire location history of the user in order to obtain a sequence of spatio-temporal entropy that represents a time series. Parallel to this first step, it is also important to collect several variables that we want to explore during the next step, which is the mobility analysis. These variables can belong to different categories (e.g., spatio-temporal, demographic and mean of transportation variables) will be used to see their influence on the evolution of the spatio-temporal entropy. For example, these variables can be the day of the week, the maximum speed of the user during the time window, if a user has a job during the week, etc…

The second challenge is now to analyze the mobility behavior of a user or a group of users. Hence, we must evaluate how the user or the users usually move(s). To do so, we decide to user Generalized Additive Models (GAMs) that are extremely interesting for this work because these models can not only predict a response variable, which is obviously the spatio-temporal entropy, but also measure the effects of co-variables on this response variable. A GAM can be formally noted as follows in Equation 6.1. Let Y be the spatio-temporal entropy variable and $X_1, ..., X_m$ a set of co-variables. β_0 is the intercept and f_j, $j = 1, \ldots, m$ are smoothed functions. We consider that we have n observations, such that $\{(y_i, x_{1,i}, \ldots, x_{m,i})\}_{i=1}^{n}$ Finally, ε_i is a random error.

$$y_i = \beta_0 + f_1(x_{1_i}) + \ldots + f_m(x_{m_i}) + \varepsilon_i \qquad (6.1)$$

6.3 From locations to a spatio temporal entropy sequence

This section presents how we can compute the spatio-temporal entropy sequence from a location history of a user. Firstly, we describe how is captured the location history of the user, and secondly, we detail its transformation into a sequence of spatio-temporal entropy.

6.3.1 Location history

In order to collect the location history of a user, we consider that a user owns a mobile device being able to capture its location at any time via the use of an embedded postitioning system. In addition, this user is moving on the surface of the earth and can use different means of transport. A raw location of the user captured by the mobile device has the following notation: $loc = (\phi, \lambda, t)$ in which ϕ is a latitude, λ is a longitude and t is a timestamp representing when the location was caught on the mobile device. Consequently, the location history of a user can be expressed as the following sequence of n locations: $L = \langle loc_1, loc_2, \cdots, loc_n \rangle$.

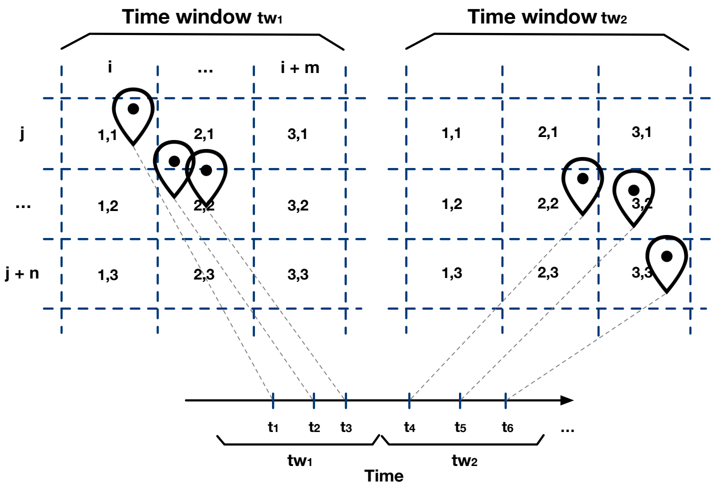


Fig. 6.1: Grid structure supporting the computation of the spatio-temporal entropy sequence

6.3.2 Spatio-temporal entropy sequence

To compute the spatio-temporal entropy sequence of a user based on her location history, we firstly need to create a spatial grid to discretize the space in which the user is moving. The space is divided into a 2-dimensional space in which the position of each cell in this grid is described as a pair (i, j). We also note n and m the numbers of cells along the x-axis and along the y-axis respectively. We divide the time duration of the location history of the user into T time windows having the same period of time, e.g., 3600 seconds (one hour). Figure 6.1 depicts the state of the grid for two successive time windows. Then, we compute the time proportion $p_{i,j}^{tw_k}$ for each visited cell of the grid by the user during the specific time window tw_k, in which k is the kth time window computed.

Before describing the entropy computation of a specific time window, we detail the computation of the time proportion below.

Time proportion computation.

The computation of a time proportion $p_{i,j}$ spent in a cell (i, j) of the grid visited by the user during a specific time window is linked to all successive locations visited in this cell during this specific time window. In order to support our explanation, Figure 6.2 presents the context of time proportion computation of two or three visited cells of the grid.

This part aims at explaining the four different cases that we can encounter when we must compute the time proportion of visited cells during a time window tw_k by a user. The first case describes a simple case in which we can consider that the user spent the entire time window tw_k in the cell $(2, 2)$. In the second case, we assume that the user was in cell $(2, 2)$ since the beginning of the time window tw_k and then in the cell $(3, 3)$ until the end of tw_k. The time spent between the second location and the third location is equally distributed between $(2, 2)$ and $(3, 3)$. The third case is similar to the second case. However, the time spent between the first location and the second location is equally distributed between $(1, 1)$ and $(2, 2)$. Finally, the fourth case is the combination of the last two cases.

In order to formally describe the four cases, we introduce the following elements below. Firstly, we introduce a first new sequence of locations L_{tw_k} that contains the successive locations visited by the user during the time window tw_k, which is obviously a subsequence of the location history of the user L. Secondly, we introduce a second new sequence of locations $L_{i,j_{tw_k}}$ containing all successive locations belonging to the same cell that are visited during the time window tw_k, which is obviously a subsequence of the location history of the user L_{tw_k}. Finally, if the subsequence of locations $L_{i,j_{tw_k}}$ contains more than one location, there are four specific cases described in Equation 6.2, 6.3, 6.4 and 6.5 below in order to illustrate the four specific cases described below. For sake of simplicity, we add $.f$ or $.l$ to the subsequences introduced above in order to identify the first or the last location(s) of them respectively. We also assume that $L_{i,j_{tw_k}}.f$ corresponds to loc_b in the sequence L_{tw_k} and $L_{i,j_{tw_k}}.l$ corresponds to loc_c in the sequence L_{tw_k}. In addition, we define $t_{s_{tw_k}}$ and $t_{e_{tw_k}}$ as the starting and ending timestamp of the time window tw_k respectively.

First case: if $L_{tw_k}.f$ is equal to $L_{i,j_{tw_k}}.f$ and if $L_{tw_k}.l$ is equal to $L_{i,j_{tw_k}}.l$.

$$p_{i,j}^{tw_k} = \frac{(t_{e_{tw_k}} - L_{i,j_{tw_k}}.f) + (L_{i,j_{tw_k}}.l - L_{i,j_{tw_k}}.f) + (t_{e_{tw_k}} - L_{i,j_{tw_k}}.l)}{t_{e_{tw_k}} - t_{s_{tw_k}}} \tag{6.2}$$

Second case: if $L_{tw_k}.f$ is equal to $L_{i,j_{tw_k}}.f$ and if $L_{tw_k}.l$ is not equal to $L_{i,j_{tw_k}}.l$.

$$p_{i,j}^{tw_k} = \frac{(t_{e_{tw_k}} - L_{i,j_{tw_k}}.f) + (L_{i,j_{tw_k}}.l - L_{i,j_{tw_k}}.f) + ((loc_c.t - loc_{c+1}.t)/2.0)}{t_{e_{tw_k}} - t_{s_{tw_k}}} \tag{6.3}$$

Third case: if $L_{tw_k}.f$ is not equal to $L_{i,j_{tw_k}}.f$ and if $L_{tw_k}.l$ is equal to $L_{i,j_{tw_k}}.l$.

$$p_{i,j}^{tw_k} = \frac{((loc_{b-1}.t - loc_b.t)/2.0) + (L_{i,j_{tw_k}}.l - L_{i,j_{tw_k}}.f) + (t_{e_{tw_k}} - L_{i,j_{tw_k}}.l)}{t_{e_{tw_k}} - t_{s_{tw_k}}} \tag{6.4}$$

Fourth case: if $L_{tw_k}.f$ is not equal to $L_{i,j_{tw_k}}.f$ and if $L_{tw_k}.l$ is not equal to $L_{i,j_{tw_k}}.l$.

$$p_{i,j}^{tw_k} = \frac{((loc_{b-1}.t - loc_b.t)/2.0) + (L_{i,j_{tw_k}}.l - L_{i,j_{tw_k}}.f) + ((loc_c.t - loc_{c+1}.t)/2.0)}{t_{e_{tw_k}} - t_{s_{tw_k}}} \tag{6.5}$$

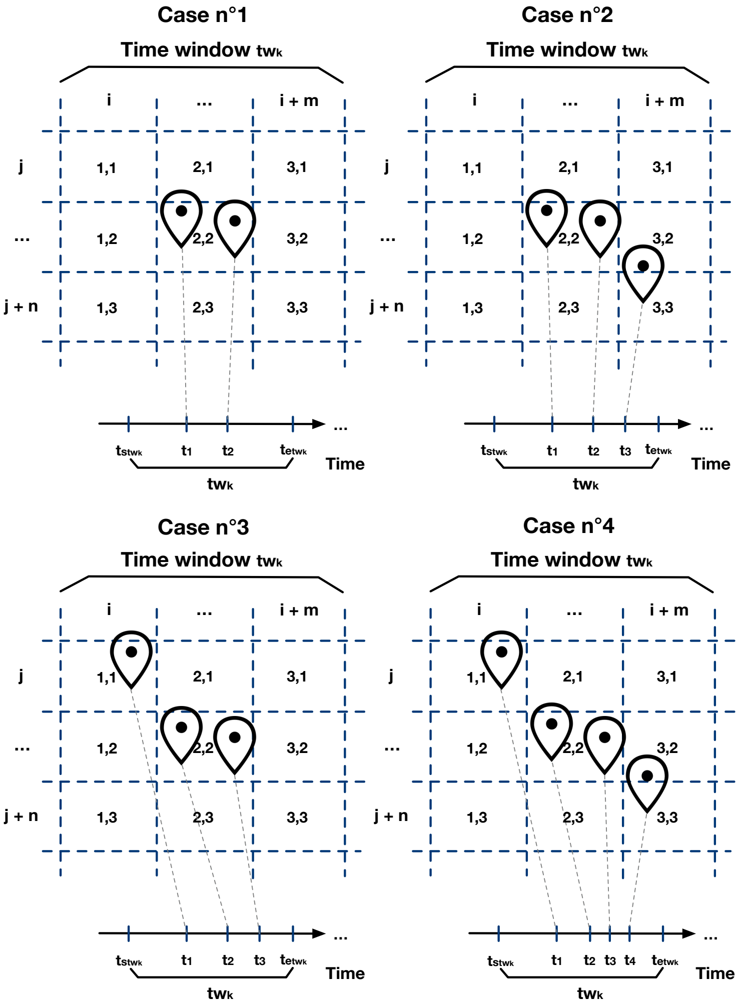


Fig. 6.2: Time proportion $p_{i,j}^{tw_k}$ computation for a specific time window tw_k

If there is no location in the cells of the grid visited by the user during the time window, i.e., in L_{tw_k}, the entropy of this time window takes the simple value of *"NA"*.

Spatio-temporal entropy computation.

Finally, the entropy computation of a specific time window is detailed in Equation 6.6. This equation indicates that the entropy is a percentage ranged between 0.0% and 100.0%.

$$entropy_{tw_k} = -\sum_{i=1}^{n}\sum_{j=1}^{m}\frac{p_{i,j}^{tw_k}\log_2 p_{i,j}^{tw_k}}{\log_2(n \times m)} \times 100.0 \tag{6.6}$$

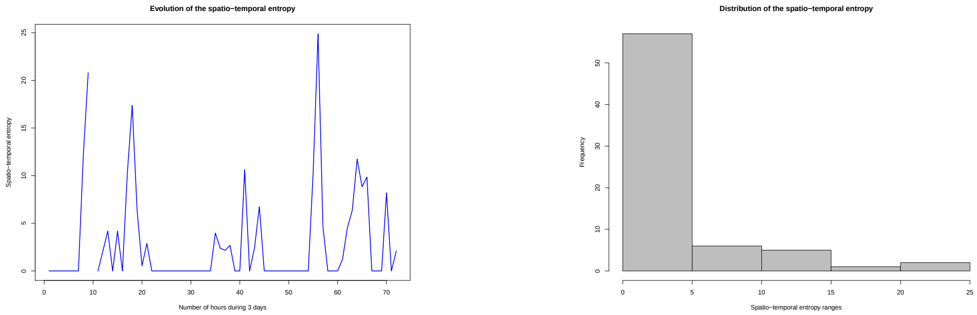


Fig. 6.3: Spatio-entropy sequence detailed for one user during 3 days

At the end of this process, we must obtain a sequence of T entropies that describe the movement rhythm represented by the spatio-temporal entropy sequence of the user. Figure 6.3 (on the left) describes the evolution of the spatio-temporal entropy of a user of the data collection campaign *Breadcrumbs* during three days. We can observe stationary (when the entropy is equal to 0.0%) and non-stationary periods (when the entropy is greater than 0.0%). In addition, Figure 6.3 (on the right) depicts the distribution of the spatio-temporal entropy during these three days for the same user. We notice a high number of low entropy levels, this makes sense because a day of a typical human usually contains a large number of stationary periods per day, especially the users of the data collection campaign *Breadcrumbs* who are mainly students (more information about *Breadcrumbs* is available in Section 6.4.1).

6.4 Mobility behavior analysis and prediction

We present in this section how it is possible to analyze mobility behavior of one user (a group of users) through the evolution of her (their) spatio-temporal entropy and additional variables that could help to better understand her (their) behavior. The dataset used is called *Breadcrumbs* and contains real mobility traces of users. For this research work, we use three categories of variables that are detailed in Section 6.4.2: spatio-temporal variables, demographic variables and mean of transport variables.

We first describe the dataset *Breadcrumbs* we used and the variables we chose for this work. Then, we demonstrate how it is possible to see the effects of these variables on a user's or users' movements expressed as an entropy rhythm with the use of Generalized Additive Models (GAMs). Finally, we detail how we can predict this rhythm and what is the prediction accuracy we can obtain by using two types of models: GAMs and AutoRegressive Integrated Moving Average (ARIMA) models.

6.4.1 Breadcrumbs dataset

Breadcrumbs dataset[1] has been collected during a data collection campaign that started at the end of March 2018 and finished at the end of June 2018 in the Lake Geneva region, at the University campus of Lausanne more specifically (Switzerland). We asked the participants (we called them users) to install an iOS application in order to record their location history during a period of three months. Most users were full-time students. The reason why we decided to launch this data collection campaign was to obtain rich user datasets having a rich density in terms of location tracking: at least one location per hour in the best case. In addition, we wanted to obtain a ground truth of points of interest validated by the users themselves. For this specific research work, we selected 48 users having very rich location history with only 3 days without any location. The duration of the selected user datasets is between 31 and 34 days. Because the data collection campaign was in progress during the period of this analysis, we could only obtain this dataset duration. The added value of this dataset, compared to existing datasets, is that there is a higher location frequency tracking per day and also various additional variables that indicate demographic and mobility characteristics related to the users, such as transportation means used, age group, and much more.

6.4.2 Selected variables

As mentioned in the previous section, we also collected various additional data about demographic and mean of transport characteristics of the users of the *Breadcrumbs* data collection campaign. Moreover, we computed several spatio-temporal variables in addition to the spatio-temporal entropy for each time window analyzed for each user. Below, you find the description of each selected variables for our research work and their possible values.

Spatio-temporal variables (computed for each time window):

- tsnb: Number of the time window (over one week)
- maxdistance: Maximum distance travelled by the user during the time window
- meanspeed: Mean speed of the user during the time window
- maxspeed: Maximum speed of the user during the time window
- campus: It the user is at the campus (No: 0 / Yes: 1)
- hourNb: Hour of the day (from 1 to 24)

[1] Breadcrumbs Data Collection Campaign was a collaborative work of three research laboratories: Distributed Object Programming Lab (`http://doplab.unil.ch`), Information Security and Privacy Lab (`https://people.unil.ch/kevinhuguenin/`) and Business Information Systems and Architecture Lab (`https://wp.unil.ch/bisa/`)

- night: If the current is part of the night (night period: from 20 pm to 7 am)
- dayNb: Day of the week (from 1 to 7)
- prevdayNb: Previous day number of the current day of the week
- nextdayNb: Next day number of the current day of the week
- weekend: If the day is a weekend day (No: 0 / Yes: 1)

This first group of variables is generated for each computed time window. As mentioned in Section 6.3, if there is no location during a time window, the spatio-temporal entropy is equal to *"NA"*. Similarly, the spatio-temporal variables of a time window also have a value equal to *"NA"* if there is no location is captured during this time window.

Demographic variables (for each user analyzed):

- gender: Gender of the user (Male: 0 / Female: 1)
- age_group: Age group of the user (Between 18 and 21 years old: 0 / Between 22 and 27 years old: 1 / Between 28 and 30 years old: 2 / More than 30 years old: 3)
- working_profile: Working profile of the user (Studying full-time: 0 / Studying part-time :1 / Working part-time (less than 80%) :2)
- job: If the user has a job during the week (No: 0 / Yes: 1)
- university: The university of the user (UniCampus1: 0 / UniCampus2: 1 / UniOutsideCampus1: 2 / UniOutsideCampus2: 3 / OtherUniversities: 4)
- section: Section of the user (Bachelor: 0 / Master: 1 / PhD: 2 / Other section: 3)
- living_parent_s_home: If the user lives at her parents' home (No: 0 / Yes: 1)
- parent_s_home_location: The location of the parents' home of the user (Other: 0 / Near the campus: 1)
- family_status: Family status of the user (Single: 0 / Free union: 1 / Married: 2 / Divorced: 3 / Other: 4)
- sport_exercises_frequence: Sport activity frequency of the user (Less than 1 hour: 0 / Between 1 hour and 5 hours: 1 / More than 5 hours: 2)
- student_association: If the user is the member of a student association (No: 0 / Yes: 1)
- smoking_cigarettes: If the user smokes cigarettes (No: 0 / Yes: 1)
- seasonal_allergies: If the user has seasonal allergies (No: 0 / Yes: 1)
- diet: The diet of the user (Diversified and not necessarily organic: 0 / Diversified and most of the time organic: 1 / Vegetarian: 2 / Vegan : 3 / Unspecified: 4)

Mean of transport variables (for each user analyzed):

- car_week: If the user uses a car during the workweek (from Monday to Friday)
- car_weekend: If the user uses a car during the weekend
- public_transportation_week: If the user uses public transportation during the workweek

- public_transportation_weekend: If the user uses public transportation during the weekend
- bike_week: If the user uses a bike during the workday
- bike_weekend: If the user uses a bike during the weekend
- taxi_week: If the user uses taxi during the workday
- taxi_weekend: If the user uses taxi during the weekend
- walking_week: If the user walks during the workday
- walking_weekend: If the user walks during the weekend

6.4.3 Analyzing mobility behavior

Below, we present the results obtained for the analysis of mobility behavior from the point of view of one single user and from the point of you of a group of users. It is very important to indicate that the duration of each time window computed for each user is equal to 1 hour. The cells of the grid, used to compute the spatio-temporal entropy sequence, are generated by adding a latitude/longitude difference of 0.0025. This means that the difference between two latitudes is approximately 278 meters and the difference between two longitudes is approximately 188 meters on an average for all cells of a common grid for the 48 users. This very short duration is very flexible and enables to study seasonal movements: for example, every hour during one day, every day during one week, and for other seasonal scales depending on the entire dataset duration of a user.

6.4.3.1 From one user point of view

We observe firstly the impact of two seasonal variables on the mobility behavior of users of *Breadcrumbs* dataset individually. More specifically, our goal is to see how the spatio-temporal entropy is evolving at a scale of one day and at a scale of one week. To do so, we build an individual GAM for each user, which is described in the following Equation 6.7, in which y_i is the response variable, i.e., the spatio-temporal entropy, β_0 is the intercept and ε_i the random error. $s(hourNb, k = k_{hour})$ is a smooth function of the number of hour of the day and $s(dayNb, k = k_{day})$ is a smooth function of the number of day of the week, in which k is a smoothing parameter representing the number of knots over the study period. Figure 6.3 suggests that we use the Gamma family distribution for individual user spatio-temporal entropy.

$$y_i = \beta_0 + s(hourNb, k = k_{hour}) + s(dayNb, k = k_{day}) + \varepsilon_i \tag{6.7}$$

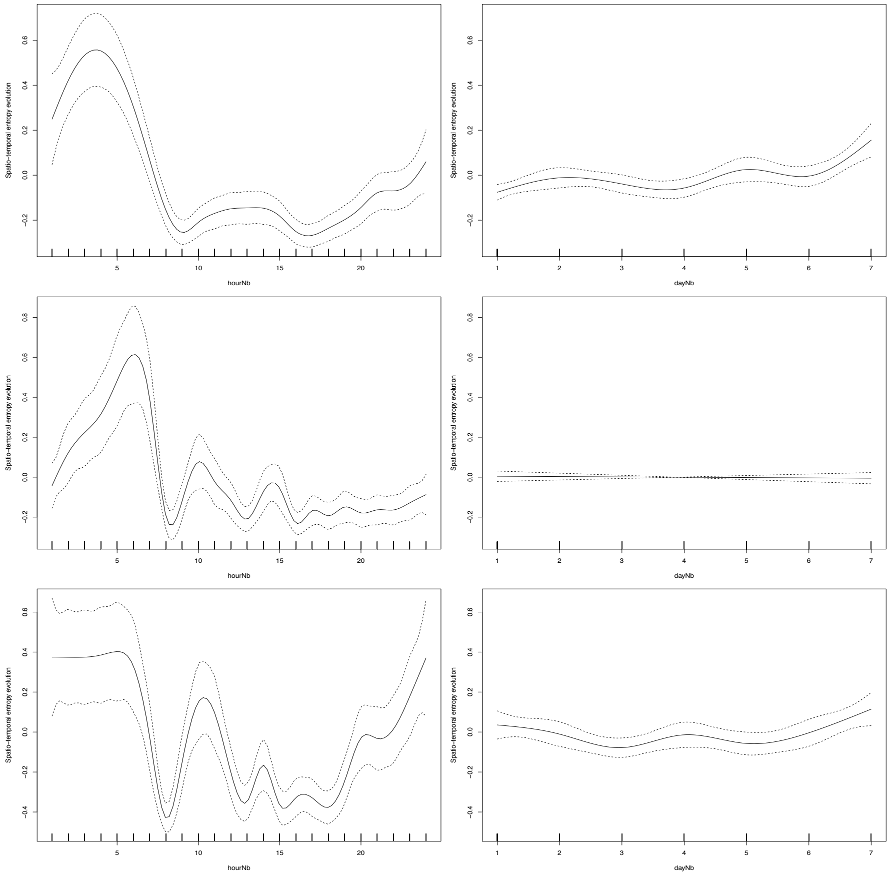
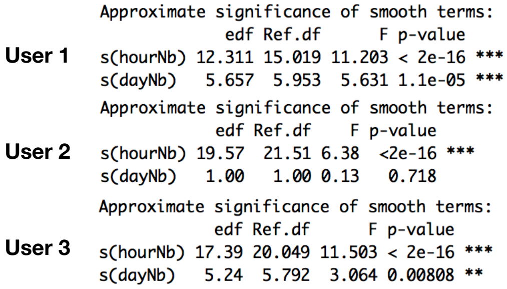


Fig. 6.4: Different mobility behavior of three users according to seasonal variables

Figure 6.4 describes an analysis of three different users of the *Breadcrumbs* dataset (User 1 is at the top, User 2 is in the middle and User3 is at the bottom). We can see on the left side, the evolution of the spatio-temporal entropy according to the hours of one day, and on the right side, the evolution of the spatio-temporal entropy according to the days of one week. On the left side of the figure, if we look at the sub-figure of the User 2 in the middle, the entropy constantly increases and rapidly decreases after (maybe when the user reaches the campus) during the morning. We observe that there are two peaks, in this same sub-figure, that could potentially highlight lecture breaks and the end of the day is relatively calm and constant. On the right side of the figure, the three sub-figures clearly show different weekly mobility behavior, the week of User 2 in the middle is very regular compared to User 1 and User 3 for example. In addition, Figure 6.5 highlights the significance of the smooth co-variables for each user by looking at the *p-value*. Unsurprisingly, the co-variable $hourNb$ is significant for all users, whereas $dayNb$ is only significant for User 1 and User 3.



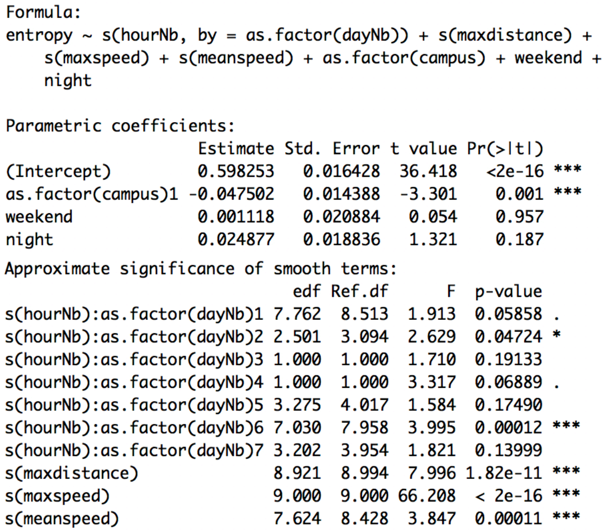
```
        Approximate significance of smooth terms:
                      edf Ref.df      F p-value
User 1  s(hourNb) 12.311 15.019 11.203 < 2e-16 ***
        s(dayNb)   5.657  5.953  5.631 1.1e-05 ***
        Approximate significance of smooth terms:
                     edf Ref.df    F p-value
User 2  s(hourNb) 19.57  21.51 6.38  <2e-16 ***
        s(dayNb)   1.00   1.00 0.13   0.718
        Approximate significance of smooth terms:
                     edf Ref.df      F p-value
User 3  s(hourNb) 17.39 20.049 11.503 < 2e-16 ***
        s(dayNb)   5.24  5.792  3.064 0.00808 **
```

Fig. 6.5: Significance of the smooth terms for the three users

Finally, Figure 6.6 shows the formula and the results of a more complex GAM applied on the data of User 1. We can clearly note that, by seeing the *p-value* of the co-variables, the spatio-temporal entropy decreases when the User 1 is at the campus. The *p-values* of the smooth terms are computed with a Wald test, while a test of Fisher (F-test) is used to compute the *p-values* of the parametric coefficients. Figure 6.7 depicts the evolution of the spatio-temporal entropy for every hour day after day. From this figure, we can highlight different types of mobility per day, Day 3 and Day 4 are very regular compared to the other. It is possible that User 1 stayed at home or at a fixed location during the days analyzed of her dataset. Interestingly, the weekend and the night does not necessarily shave an influence on the user's behavior. Regarding the night, it is maybe due to the fact that the night period is tagged during a long duration (from 20 pm to 7 am) and that a user can move a lot during this period at home, especially if she is working late. The Figure 6.6 also shows that the hours of Saturday (Day 6) have a specific influence on the evolution of the spatio-temporal entropy. We also observe a high effect of the maximum distance, the maximum speed and the mean speed on the spatio-temporal entropy.

6.4.3.2 From multiple users' point of view

This second analysis focuses on the mobility behavior of a group of users and, most specifically, the variables that can have an influence on their mobility. For information, the distribution of the spatio-temporal entropy of the 48 users also follows a Gamma distribution, as for each individual user of the dataset. Figure 6.8 enables to see the influence of the spatio-temporal, demographic and means of transport variables of the evolution of the spatio-temporal entropy of the 48 users of the *Breadcrumbs* dataset. When we see different number for the different variables studied (as.factor(variablename)nb), this means that the variable is a factor with different levels. From this figure, we are now able to highlight the co-variables that have an influence on the evolution of the spatio-temporal entropy for the 48 users. Before talking about the results, we purposely removed some co-variables because some of



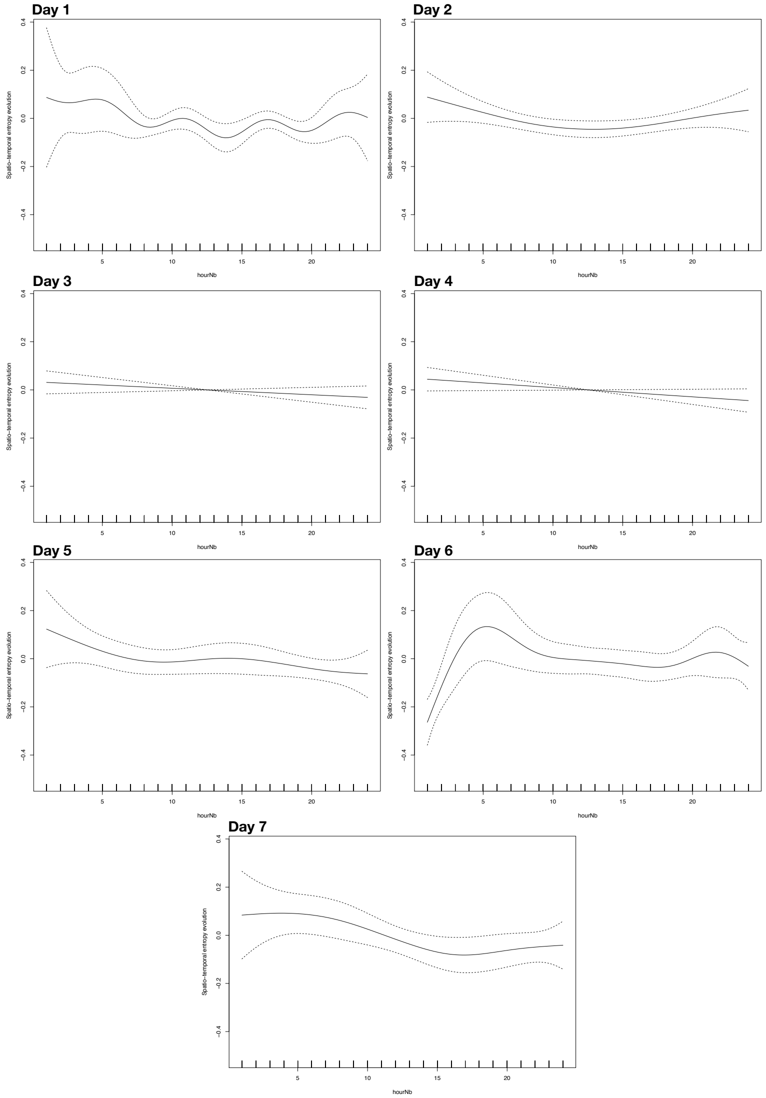
```
Formula:
entropy ~ s(hourNb, by = as.factor(dayNb)) + s(maxdistance) +
    s(maxspeed) + s(meanspeed) + as.factor(campus) + weekend +
    night

Parametric coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)         0.598253   0.016428  36.418   <2e-16 ***
as.factor(campus)1 -0.047502   0.014388  -3.301    0.001 ***
weekend             0.001118   0.020884   0.054    0.957
night               0.024877   0.018836   1.321    0.187
Approximate significance of smooth terms:
                              edf Ref.df      F  p-value
s(hourNb):as.factor(dayNb)1 7.762  8.513  1.913  0.05858 .
s(hourNb):as.factor(dayNb)2 2.501  3.094  2.629  0.04724 *
s(hourNb):as.factor(dayNb)3 1.000  1.000  1.710  0.19133
s(hourNb):as.factor(dayNb)4 1.000  1.000  3.317  0.06889 .
s(hourNb):as.factor(dayNb)5 3.275  4.017  1.584  0.17490
s(hourNb):as.factor(dayNb)6 7.030  7.958  3.995  0.00012 ***
s(hourNb):as.factor(dayNb)7 3.202  3.954  1.821  0.13999
s(maxdistance)              8.921  8.994  7.996 1.82e-11 ***
s(maxspeed)                 9.000  9.000 66.208  < 2e-16 ***
s(meanspeed)                7.624  8.428  3.847  0.00011 ***
```

Fig. 6.6: Additional co-variables added in GAM formula for User 1

their options were underrepresented. Indeed, the 48 users were too homogeneous to analyze all demographic variables, such as age group, working profile and family status. Hence, Figure 6.8 depicts the results that are summarized below:

- Being at the campus (campus 1), using a taxi during the weekend (taxi_weekend 1), walking during the weekend (walking_weekend 1) and doing a sport activity at a high frequency (sport_exercices_frequence 2) decrease the spatio-temporal entropy;
- Using a taxi during the week (taxi_week 1), walking during the week (walking_week 1), studying in a university that is not located at the campus (university 4), being in a master section (section 2) and having a specific diet (diet 2) increase the spatio-temporal entropy.

Interestingly, the results show that co-variable *dayNb* has a negative influence on the evolution of the spatio-temporal entropy. All these indicators give us good insights to better understand the way a specific population is moving. These results also indicate that both demographic and means of transport co-variables can influence our mobility behavior. Further work should be done to study the behavior of specific demographic groups with GAMs to explore them at a finer scale.


Day 1

Day 2

Day 3

Day 4

Day 5

Day 6

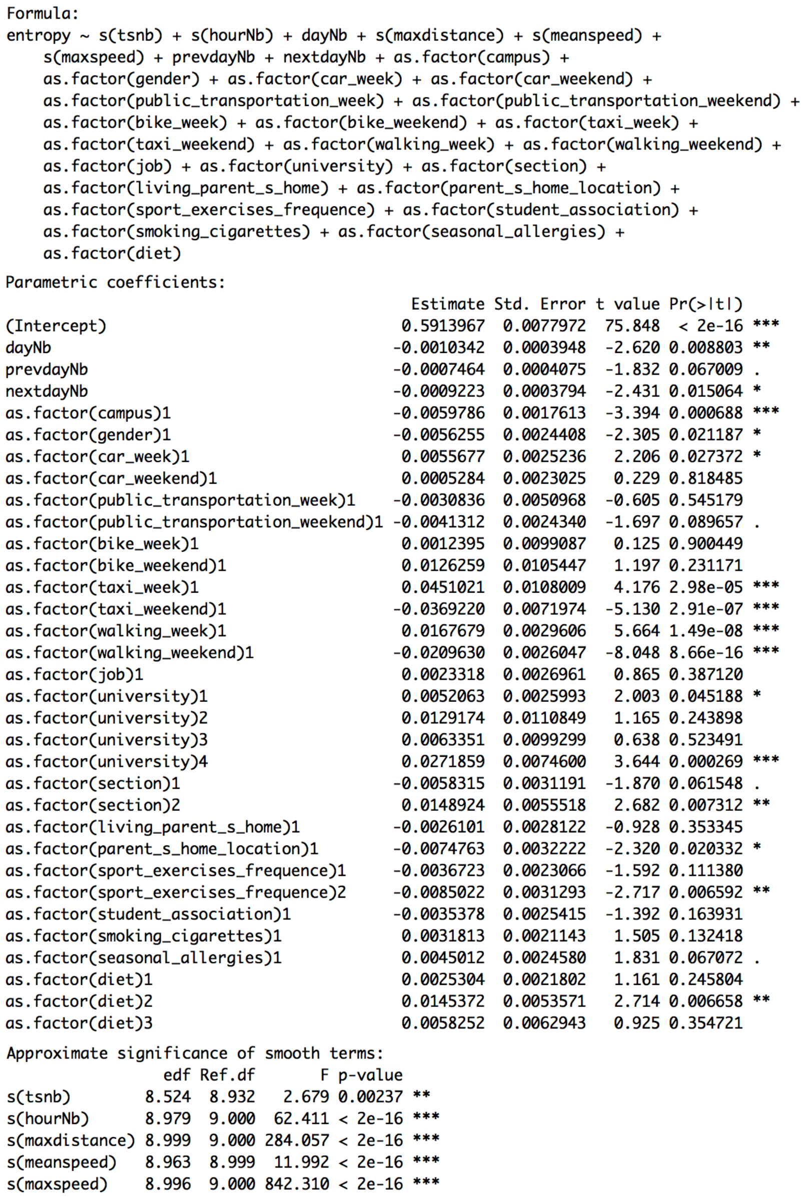
Day 7

Fig. 6.7: Spatio-temporal evolution day after day for User 1

6.4.4 Predicting mobility behavior

In order to evaluate GAMs from a prediction perspective regarding the mobility of users, we decide to compare the prediction accuracy between GAMs and AutoRegressive Integrated Moving Average (ARIMA) models.

The GAMs predict a variable based on the value of the co-variables given, while the ARIMA models predict a variable based on its past values and its past variance according to three parameters p, d and q. Parameter p is the number of autoregressive terms, parameter d is the number of seasonal differences and parameter q is the number of seasonal moving average terms. For our research work, we entirely delegate

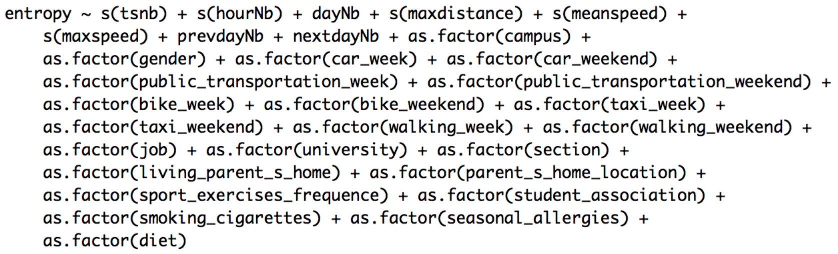


```
Formula:
entropy ~ s(tsnb) + s(hourNb) + dayNb + s(maxdistance) + s(meanspeed) +
    s(maxspeed) + prevdayNb + nextdayNb + as.factor(campus) +
    as.factor(gender) + as.factor(car_week) + as.factor(car_weekend) +
    as.factor(public_transportation_week) + as.factor(public_transportation_weekend) +
    as.factor(bike_week) + as.factor(bike_weekend) + as.factor(taxi_week) +
    as.factor(taxi_weekend) + as.factor(walking_week) + as.factor(walking_weekend) +
    as.factor(job) + as.factor(university) + as.factor(section) +
    as.factor(living_parent_s_home) + as.factor(parent_s_home_location) +
    as.factor(sport_exercises_frequence) + as.factor(student_association) +
    as.factor(smoking_cigarettes) + as.factor(seasonal_allergies) +
    as.factor(diet)
Parametric coefficients:
                                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                               0.5913967  0.0077972  75.848  < 2e-16 ***
dayNb                                    -0.0010342  0.0003948  -2.620 0.008803 **
prevdayNb                                -0.0007464  0.0004075  -1.832 0.067009 .
nextdayNb                                -0.0009223  0.0003794  -2.431 0.015064 *
as.factor(campus)1                       -0.0059786  0.0017613  -3.394 0.000688 ***
as.factor(gender)1                       -0.0056255  0.0024408  -2.305 0.021187 *
as.factor(car_week)1                      0.0055677  0.0025236   2.206 0.027372 *
as.factor(car_weekend)1                   0.0005284  0.0023025   0.229 0.818485
as.factor(public_transportation_week)1   -0.0030836  0.0050968  -0.605 0.545179
as.factor(public_transportation_weekend)1 -0.0041312  0.0024340  -1.697 0.089657 .
as.factor(bike_week)1                     0.0012395  0.0099087   0.125 0.900449
as.factor(bike_weekend)1                  0.0126259  0.0105447   1.197 0.231171
as.factor(taxi_week)1                     0.0451021  0.0108009   4.176 2.98e-05 ***
as.factor(taxi_weekend)1                 -0.0369220  0.0071974  -5.130 2.91e-07 ***
as.factor(walking_week)1                  0.0167679  0.0029606   5.664 1.49e-08 ***
as.factor(walking_weekend)1              -0.0209630  0.0026047  -8.048 8.66e-16 ***
as.factor(job)1                           0.0023318  0.0026961   0.865 0.387120
as.factor(university)1                    0.0052063  0.0025993   2.003 0.045188 *
as.factor(university)2                    0.0129174  0.0110849   1.165 0.243898
as.factor(university)3                    0.0063351  0.0099299   0.638 0.523491
as.factor(university)4                    0.0271859  0.0074600   3.644 0.000269 ***
as.factor(section)1                      -0.0058315  0.0031191  -1.870 0.061548 .
as.factor(section)2                       0.0148924  0.0055518   2.682 0.007312 **
as.factor(living_parent_s_home)1         -0.0026101  0.0028122  -0.928 0.353345
as.factor(parent_s_home_location)1       -0.0074763  0.0032222  -2.320 0.020332 *
as.factor(sport_exercises_frequence)1    -0.0036723  0.0023066  -1.592 0.111380
as.factor(sport_exercises_frequence)2    -0.0085022  0.0031293  -2.717 0.006592 **
as.factor(student_association)1          -0.0035378  0.0025415  -1.392 0.163931
as.factor(smoking_cigarettes)1            0.0031813  0.0021143   1.505 0.132418
as.factor(seasonal_allergies)1            0.0045012  0.0024580   1.831 0.067072 .
as.factor(diet)1                          0.0025304  0.0021802   1.161 0.245804
as.factor(diet)2                          0.0145372  0.0053571   2.714 0.006658 **
as.factor(diet)3                          0.0058252  0.0062943   0.925 0.354721
Approximate significance of smooth terms:
                 edf Ref.df       F p-value
s(tsnb)        8.524  8.932   2.679 0.00237 **
s(hourNb)      8.979  9.000  62.411 < 2e-16 ***
s(maxdistance) 8.999  9.000 284.057 < 2e-16 ***
s(meanspeed)   8.963  8.999  11.992 < 2e-16 ***
s(maxspeed)    8.996  9.000 842.310 < 2e-16 ***
```

Fig. 6.8: Summary of the global GAM of 48 users

the selection process of these three parameters by using a specific R function called $auto.arima()$ of the $forecast$ package. An ARIMA model can be formally described as the following Equation 6.8.

$$y_{t'} = \mu + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} - \theta_1 e_{t-1} - \ldots - \theta_q e_{t-q} \tag{6.8}$$



```
entropy ~ s(tsnb) + s(hourNb) + dayNb + s(maxdistance) + s(meanspeed) +
    s(maxspeed) + prevdayNb + nextdayNb + as.factor(campus) +
    as.factor(gender) + as.factor(car_week) + as.factor(car_weekend) +
    as.factor(public_transportation_week) + as.factor(public_transportation_weekend) +
    as.factor(bike_week) + as.factor(bike_weekend) + as.factor(taxi_week) +
    as.factor(taxi_weekend) + as.factor(walking_week) + as.factor(walking_weekend) +
    as.factor(job) + as.factor(university) + as.factor(section) +
    as.factor(living_parent_s_home) + as.factor(parent_s_home_location) +
    as.factor(sport_exercises_frequence) + as.factor(student_association) +
    as.factor(smoking_cigarettes) + as.factor(seasonal_allergies) +
    as.factor(diet)
```

Fig. 6.9: Global GAM formula trained with 60% of the dataset of the 48 users

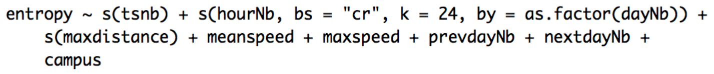
In this equation, μ is a constant, $\phi_1 y_{t-1} + \ldots + \phi_p y_{t-p}$ are the autoregressive terms and $\theta_1 e_{t-1} - \ldots - \theta_q e_{t-q}$ are the moving average terms. $y_{t'}$ is the integrated part with a certain difference noted d, such that:

- if d = 0, $y_{t'} = y_t$ (no difference)
- if d = 1, $y_{t'} = y_t - y_{t-1}$ (first difference)
- if d = 2, $y_{t'} = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2})$ (second difference)
- ...

6.4.4.1 Evaluation process

For this evaluation, we decide to compare the prediction accuracy of a global GAM with all user data, individual GAMs for each user and individual ARIMA models for each user. We firstly split the datasets of the users into two parts, the first part contains 60% and the second part 40% of the entire dataset of one user. We train the global GAM (having the formula depicted in Figure 6.9) with the aggregation of the first part of all user datasets and the individual GAMs (formula in Figure). and ARIMA models with each individual first part of the users. Then, we try to predict the remaining 40% part of each user dataset with the global GAM and the individual GAM and ARIMA model linked to the each user. It is important to indicate that individual GAMs do not include demographic and means of transport variables of users, whereas the global GAM includes them.

For this comparison, we compute the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE) of each predicted user set compared to the real values of the last 40% remaining of the dataset of each user. These two measures are complementary because the first metric highlights the absolute average magnitude of errors, whereas the second metric shows the standard deviation of these errors and gives more value to larger errors.



```
entropy ~ s(tsnb) + s(hourNb, bs = "cr", k = 24, by = as.factor(dayNb)) +
    s(maxdistance) + meanspeed + maxspeed + prevdayNb + nextdayNb +
    campus
```

Fig. 6.10: Individual GAM formula trained with 60% of the dataset of each user

Models	**MAE (average for 48 users)**	**RMSE (average for 48 users)**
Global GAM	2,74	5,45
Individual GAMs	2,73	5,66
Individual ARIMA models	3,09	4,62

Table 6.1: Global GAM, individual GAMs and individual ARIMA models comparison: results

6.4.4.2 Results

Table 6.1 summarizes the results and shows that there is no a strong difference between the results obtained for the global GAM and the individual GAMs. It is important to note that the results are expressed in the unit of the predicted variable, i.e., spatio-temporal entropy. We can also note that the average of RMSE is lower for the global GAM compared to the individual GAMs. This indicates that larger errors are maybe less numerous for the global GAM. However, we can see that the average of the RMSE is slightly higher for the global GAM compared to that of the individual ARIMA models. In addition, we can observe that the average of the MAE is almost similar for the global and individual GAMs, and slightly lower for the GAMs compared to that of the individual ARIMA models.

To conclude this prediction accuracy analysis, GAMs are more efficient to predict individual spatio-temporal entropy than ARIMA in terms of MAE.

6.5 Related work

This section presents the related work close to our work according to its two main subjects: the analysis of mobility behavior and the use of entropy-based metrics in the literature.

6.5.1 Mobility behavior analysis

Most of the work realized in the literature about the analysis of mobility focus on mobility patterns and use the significant locations visited by users. Hence, it is possible to analyze the patterns extracted from the significant locations of users at different time scale and how they evolve over time. In [4], Gonzalez et al. analyze mobility patterns from the point of view of the significant locations visited by a user and observe the spatio-temporal regularity of user patterns. In [12], Zion and Lerner also analyze the mobility of users by computing patterns with a preliminary analysis using the semantic of significant locations and a *Latent Dirichlet Allocation* method personalized for temporal analysis. In [9], Sadilek and Krumm perform a long-term analysis of patterns of users using Fourrier analysis and Principal Component Analysis. They demonstrated the influence of temporal attributes, e.g., day of week, on user patterns. Finally, in [5], Kulkarni et al. analyze user mobility behavior in order to update mobility prediction models when major changes occur in the mobility of users. In order to reach this goal, they analyze the frequency of movements and also the major changes that appear at the level of the significantly and frequently visited places by users.

6.5.2 Entropy-based analysis

The entropy is a well-known notion in the field of information theory, which can be applied to various domains, from physics to text analysis. In the field of this work, i.e., human mobility analysis, the Shannon entropy is mainly used to evaluate the level of predictability of a user. In [11], Song et al. propose three entropy measures: the *random entropy* capturing the degree of predictability of the user's locations, the *temporal-uncorrelated entropy* highlighting the visit probability of a location by a user, and finally, the *actual entropy* characterizing the level of order in the visit patterns of the user. This paper demonstrates that users are highly predictable because their patterns seem to be very regular. The authors also observe that there are insignificant variations between different user groups, e.g., different genders, different age-groups. The issue of this paper is that the dataset used was not very precise in terms of location tracking. More specifically, they use a hourly time window to compute the third entropy whereas a large number of hours was missing, 70% to be precise for a typical user analyzed of the dataset. And this low number of precise data maybe leads to an homogeneous vision of the mobility of the entire set of users. In [8], Qin et al. also use the entropy to identify the level of predictability of a user. They use a first entropy metric to measure the regularity level of a user over time slots. In addition, they use a second entropy measure to evaluate the quality of the clusters because this can strongly have an influence on the prediction of the next significant location of a user. The first measure of the entropy is mainly based on the significant location that dominates during a time slot in order to extract the regularity of the user over the entire time slots during one day at the end of the process. In [6], Lu et al. compute the entropy in a three-step process as described in the first paper (see [11]) in order to highlight the level of predictability of users. They discover that the population they study is highly predictable. They also indicate the importance of weekly cycle in the mobility of their users. As for the first paper mentioned in this section, they use a dataset in which the locations are cell phone towers. The quality of the dataset used can also be a problem because a user can only have one location per day in the worst case. In [1], Austin et al. study the regularity and predictability of movements at home. At this scale, they also find that the patterns of the users analyzed are significantly regular and open the work to the health domain. In a very different domain, Sharma et al. study collaborative remote work amongst users in [10]. They propose a computation of entropy according to a time proportion spent in different zones of a screen in order to analyze the effectiveness of collaborative remote work between two users. Our computation of the spatio-temporal entropy is close to this last work and has been personalized to our analysis.

To the best of our knowledge, the research works, about mobility and mentioned above, always use the most significant locations of users to study the mobility of the users. Our work opens a new entropy-based metric and a new complete methodology to study mobility in a privacy manner. Indeed, we avoid the step of the extraction of the significant locations of users in order to analyse the mobility of a user as a pure rhythm that can be influenced by several variables. In addition, we use Generalized Additive Models (GAMs) to analyze the mobility rhythm of a user, and, to the best of our knowledge, these models were not applied on this field before. We find their use in other scientific domain, such as environmental domain. For example, in [7], Pearce et al. analyse the impact of different local meteorology variables on air quality.

6.6 Conclusion and future work

This research work has two main contributions. Firstly, it contributes to the description of a new methodology that enables to study user's movements in a privacy-aware manner without extracting sensitive places of users. Secondly, we show that we are able to highlight the variables that can affect these movements using a sequence of spatio-temporal entropy that represents the user's movement as a rhythm. A sequence of spatio-temporal entropy describes the level of mobility of a user as a time series, in which each spatio-temporal entropy is computed for a specific time window. We also present the use of Generalized Additive Models (GAMs) for this mobility analysis by using a dataset we collected, called *Breadcrumbs* during the data collection campaign of the same name. This research work also highlights that a GAM can also be used to analyze the movements of an entire population, as we did it this work with the 48 students analyzed. Compared to individual GAMs and ARIMA models, a global GAM provide the best prediction in terms of accuracy regarding the MAE.

This work opens three other interesting threads. Firstly, we could use GAMs with longer user datasets, in terms of duration, in order to see if we observe lower MAE and RMSE and study their evolution over time for each user dataset. This first analysis could also give a good understanding of the level of predictability of a user. A second future work could focus on the prediction of the demographic data of a user based on the learning of the mobility trends of an aggregation of multiple datasets including the spatio-temporal entropy and other related variables. Finally, the third work could be linked to the creation of individual personalized mobility prediction models for a pure location prediction goal. These personalized could be based on the seasonal mobility trends observed for each user in order to adjust their temporal scale.

References

[1] Daniel Austin, Robin M. Cross, Tamara Hayes, and Jeffrey Kaye. Regularity and predictability of human mobility in personal space. *PloS one*, 9(2):e90256, 2014.

[2] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376, 2013.

[3] Vincent Etter, Mohamed Kafsi, and Ehsan Kazemi. Been there, done that: What your mobility traces reveal about your behavior. In *Mobile Data Challenge by Nokia Workshop, in conjunction with Int. Conf. on Pervasive Computing*, number EPFL-CONF-178426, 2012.

[4] Marta C. Gonzalez, Cesar A. Hidalgo, and Albert-László Barabási. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.

[5] Vaibhav Kulkarni, Arielle Moro, and Benoît Garbinato. Mobidict: a mobility prediction system leveraging realtime location data streams. In *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming*, page 8. ACM, 2016.

[6] Xin Lu, Erik Wetter, Nita Bharti, Adrew J. Tatem, and Linus Bengtsson. Approaching the limit of predictability in human mobility. *Scientific Reports*, 3:2923, October 2013.

[7] John L. Pearce, Jason Beringer, Neville Nicholls, Rob J Hyndman, and Nigel J Tapper. Quantifying the influence of local meteorology on air quality using generalized additive models. *Atmospheric Environment*, 45(6):1328–1336, 2011.

[8] Shaomeng Qin, Hannu Verkasalo, Mikael Mohtaschemi, Tuomo Hartonen, and Mikko Alava. Patterns, entropy, and predictability of human mobility and life. *CoRR*, abs/1211.3934, 2012.

[9] Adam Sadilek and John Krumm. Far out: Predicting long-term human mobility. In *AAAI*, 2012.

[10] Kshitij Sharma, Valérie Chavez-Demoulin, and Pierre Dillenbourg. Non-stationary modeling of tail-dependence of two subjects concentration. *Submitted to Annals of Applied Statistics*, 2016.

[11] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of Predictability in Human Mobility. *Science*, 327(5968):1018–1021, February 2010.

[12] Eyal Ben Zion and Boaz Lerner. Learning human behaviors and lifestyle by capturing temporal relations in mobility patterns. 2017.

Chapter 7
Conclusion

In this thesis, we have explored several research questions, presented in Section 1.2 and organized in three groups. Although each chapter of this thesis has its own conclusion, we first summarize our contributions in Section 7.1. Then, we present future work perspectives that are opened from this thesis, in Section 7.2.

7.1 Contributions

In this thesis, we have proposed various solutions for preserving the location privacy of users when they use location-based services, such as system architectures and algorithms potentially linked to mobility prediction. In addition, we have proposed a new metric for quantifying location privacy as a value during the sharing of locations with location-based services. We have also presented a privacy preserving analysis of the mobility of users by using the evolution of their spatio-temporal entropy. All our main contributions in this thesis are summarized hereafter. For consistency purposes, we follow the same structure in all three parts to summarize them.

Part I - Location Privacy.

In Chapter 2, we have described an architecture that includes a sharing protocol, called *ShareZ*: it preserves the location privacy of users when they use location-based services. This architecture focuses on sending location aggregations, expressed as zones of interest, to location-based services in order to avoid the sharing of a location stream of a user with these services. It contains an algorithm that enables us to automatically create a location privacy tree made of the zones of interest of a user. Instead of revealing an entire sequence of locations, the user has the possibility to choose the granularity of the zones sent to location-based services. However, this work has some limitations because it is focused only on location-based services that require zones of interest in order to deliver location-based information to users. For example, a navigation application could not use it. In summary, instead of uninstalling location-based services, we let the user control, at a fine-grained scale, the location information she wants to provide to location-based services.

In Chapter 3, we have presented an estimator for quantifying location privacy. This work also helps us to unify the way we represent the spatio-temporal effects of certain blurring mechanisms on a raw location or a sequence of locations by using spatio-temporal uncertainties generated when these mechanisms are applied. We have evaluated this metric by comparing the success of several localization attacks and the level of location privacy obtained with our metric; both were reached for different blurring mechanisms. We have provided a definition of a generic metric that takes into account space and time uncertainties, which are created after the application of a blurring algorithm on raw location(s). In addition, the user herself can weigh the spatial and/or temporal dimensions of the effects of a blurring mechanism. Consequently, this means that this measure is user-oriented.

Part II - Location Privacy and Mobility Prediction.

A prediction system, called *MobiDict*, has been presented in Chapter 4. This system can process location streams in real time to provide predicted locations and could, theoretically, run on mobile devices. Its evaluation has clearly shown that it is possible to compute mobility prediction models and to provide predicted locations with only a low number of locations. In addition, we also have highlighted that it is required to update the models when modifications occur in the movements of the user.

In Chapter 5, we have described an architecture that contains a system for preserving the location privacy of users when they use location-based services that need to compute future locations of users. Hence, this architecture enables location-based services to request future location of users. The entire system, called *ResPred*, includes a mobility prediction component and a location privacy component. The latter internally blurs predicted locations in order to reach a tradeoff between user location privacy and location-based service utility. This system also enables us to ensure with fine-grained control the location privacy of users because they can define their privacy preferences and, more specifically, the location privacy they are willing to sacrifice to use the service.

Part III: Mobility Behavior Analysis.

In Chapter 6, first we have used the evolution of spatio-temporal entropy of a user to extract her mobility behavior as rhythms. Unlike the standard mobility behavior analysis, this research does not start with the searching of highly sensitive frequently visited places by users. Second, Generalized Additive Models (GAMs) have been used to explore the influence of seasonal and demographic variables on the evolution of the spatio-temporal entropy of a user. We also evaluate the prediction accuracy of GAMs, compared to Autoregressive Integrated Moving Average (ARIMA) models. We have highlighted that the new methodology proposed in this chapter enables researchers to explore the mobility of users in a privacy preserving manner, i.e., without studying the user's movements amongst her frequently visited places. This opens a perspective on further work about the prediction, based on this mobility rhythms, of demographic attributes of users.

7.2 Future Work

The entire research work, presented in this thesis, opens multiple options, some of them are linked to the research papers already presented in this thesis and the others are stimulated by the emergence of new technologies and/or paradigms.

Implementing System Architectures on Mobile Devices.

The solutions proposed in the three following chapters, in Chapter 2, 4 and 5, could be implemented on mobile devices. Since the main part of the algorithms, which are the support of the proposed system architectures, are written in Swift, we could port them on iPhone devices by performing a jailbreak on them. Thus, this implementation could enable to evaluate the feasibility of the system architectures proposed and improve them by taking all mobile device constraints into account, such as a rapid battery consumption. In addition, these tests could enable to create user-friendly interfaces to involve the user in the privacy protection process.

Adapting the Location Privacy Metric to the Analysis of a Real-Time Stream of Locations.

In Chapter 3, we have presented a location privacy estimator based on spatio-temporal uncertainties. The idea is now to start from this metric and to modify it in order to use it in real time when a user starts using a location-based service. Then, this new metric could also be evaluated on a mobile device in order to evaluate its relevance during a real case study. This also requires a jailbreak of the mobile device.

Conducting a Longer Data-Collection Campaign (Breadcrumbs 2.0).

We have used, in Chapter 6, a dataset from a data-collection campaign that we launched in March 2018, called *Breadcrumbs.* One of the purposes of this dataset was to capture the location histories of users as frequently as possible, while avoiding critical location gaps in them. These gaps are one of the main weakness of existing datasets. This campaign was approved by the ethical committee of the Faculty of Business and Economics (HEC) of the University of Lausanne accepted for a short duration of only three months. As the iOS application the server, to store all the user datasets, are already built, we can use them to launch a longer data collection campaign. In addition, we could implement an Android version of the iOS application. This could enable us to analyze long-term mobility behavior and to extend the analysis of the last research paper, as indicated in the next paragraph.

Utilizing Spatio-temporal Entropy Rhythm to Discover Demographic Attributes and Create Personalized Mobility Prediction Models of Users.

In Chapter 6, we have described the use of the evolution of the spatio-temporal entropy as a means of discovering the mobility behavior of users and the events that can affect it with Generalized Additive Models (GAMs). As the evolution of the spatio-temporal entropy of a user can be interpreted as the user's mobility rhythms, these rhythms could be used as a starting point of the creation of a personalized mobility prediction model. Hence, we could include the rhythms as a key element of the prediction model. We could also try to discover demographic characteristics from the movement rhythms of a user by using GAMs. To do so, we must obtain larger, in terms of duration, user datasets as mentioned in the previous paragraph in order to learn mobility rhythms during a long period.

Implementing a Decentralized Architecture to Share Points of Interest using a Blockchain Technology.

In these past few years, blockchains have become increasingly well-known, along with the popularity of cryptocurrencies, more specifically Bitcoin. This technology enables us to create a decentralized trust supported by a shared ledger amongst all users. In paper [2], Zyskind et al. presented a system that uses a blockchain and digitally-signed transactions to enable users to manage the sharing of their personal data. In this solution, the users themselves always own their data and control the services that will have an access to their data. Another paper [1] describes how a blockchain can be interesting for the sharing of information captured, in a city, by sensors and other devices. In this future work proposition, the idea could be to use the blockchain technology as a support for sharing points of interest, in order to receive rewards of services and automatically share information with other users. The first paper [2] would be an interesting starting point of this work. In addition, the proposed system could be personalized by adding smart contracts in the blockchain; this could automatically trigger specific actions according to the location information shared by the users.

References

[1] Kamanashis Biswas and Vallipuram Muthukkumarasamy. Securing smart cities using blockchain technology. In *High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on*, pages 1392–1393. IEEE, 2016.

[2] Guy Zyskind, Oz Nathan, and Alex 'Sandy' Pentland. Decentralizing privacy: Using blockchain to protect personal data. In *Security and Privacy Workshops (SPW), 2015 IEEE*, pages 180–184. IEEE, 2015.

◊◊◊