Year : 2019

Modeling, Predicting and Capturing Human Mobility

Kulkarni Vaibhav

UNIL | Université de Lausanne

FACULTÉ DES HAUTES ÉTUDES COMMERCIALES

DÉPARTEMENT DES SYSTÈMES D'INFORMATION

MODELING, PREDICTING AND CAPTURING HUMAN MOBILITY

THÈSE DE DOCTORAT

présentée à la

Faculté des Hautes Études Commerciales
de l'Université de Lausanne

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

par

Vaibhav KULKARNI

Directeur de thèse
Prof. Benoît Garbinato

Jury

Prof. Felicitas Morhart, Présidente
Prof. Michalis Vlachos, expert interne
Dr. Sonia Ben Mokhtar, experte externe
Prof. Philippe Cudré-Mauroux, expert externe

LAUSANNE
2019

UNIL | Université de Lausanne
HEC Lausanne
Le Décanat
Bâtiment Internef
CH-1015 Lausanne

IMPRIMATUR

Sans se prononcer sur les opinions de l'auteur, la Faculté des Hautes Etudes Commerciales de l'Université de Lausanne autorise l'impression de la thèse de Monsieur Vaibhav KULKARNI, titulaire d'un bachelor en Electronics and Telecommunication Engineering de Goa College of Engineering, d'un master en Information and Communication Technology de Technische Universität Berlin et d'un master en Embedded Systems de Technische Universiteit Eindhoven, en vue de l'obtention du grade de docteur ès Sciences en systèmes d'information.

La thèse est intitulée :

MODELING, PREDICTING AND CAPTURING HUMAN MOBILITY

Lausanne, le 1er octobre 2019

Le doyen

p.o.

Jean-Philippe Bonardi

Jury

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

Professor Felicitas Morhart
Professor at the Faculty of Business and Economics of the University of Lausanne.
President of the Jury.

Professor Michalis Vlachos
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 Philippe Cudré-Mauroux
Professor at the University of Fribourg, Switzerland. Head of the eXascale Infolab.
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

Vaibhav Kulkarni

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: 27.9.2019

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

Vaibhav Kulkarni

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: 27.9.2019

Prof. Michalis Vlachos
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

Vaibhav Kulkarni

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: 27.9.2019

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

Vaibhav Kulkarni

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: 27.9.2019

Prof. Philippe Cudré-Mauroux
External member of the doctoral committee

Abstract

In this thesis, we focus on studying human mobility from three different perspectives: (1) modeling human mobility, (2) predicting human mobility, and (3) capturing human mobility. In order to gain a comprehensive understanding of the underlying dynamics that govern human mobility, we draw out parallels between mobility behaviors, statistical physics and information theory. Contrary to the widely accepted assumption that human mobility is Markovian in nature, we prove that the applicability of Markov processes should be an exception, while non-Markov should be the norm while modeling mobility. Realistic models of human mobility are critical for modern day applications that span several domains, such as urban planning, telecommunication, traffic forecasting, infectious disease diffusion, etc. These services are facilitated by large volumes of geolocation data gathered through mobile devices equipped with global positioning functionality (GPS) and Internet connectivity. Given the wide range of mobility behaviors among individuals and the disparate application requirements, the first section of the thesis aims at facilitating the mobility modeling task by adopting a data-driven approach. To this end our contribution is twofold wherein we propose: (1) a set of meta-attributes to quantify the mobility dynamics, and (2) a set of criteria for selecting the modeling technique that best suits a given performance/complexity tradeoff. We also facilitate mobility modeling research by drawing insights from a comprehensive literature survey of human mobility modeling based on 1680 articles published between 1999 and 2019.

In the second part of the thesis, we present the design and implementation of an end-to-end privacy-preserving architecture for predicting human mobility. The proliferation of location-based services (LBS) and wearable technologies has facilitated the collection and tracking of geolocation data. Data aggregation at scale is promoting the integration of novel paradigms in applications to enhance the services being provided. Mobility prediction is rapidly becoming one of these new paradigms opening up avenues for mobility-aware applications such as targeted advertising, ride-sharing optimization, smooth network handovers, etc. However, sharing sensitive location trajectories with service providers presents several privacy implications. Currently used privacy-preserving techniques, such as location obfuscation and anonymisation, have been shown to come at the cost of drastic usability limitation. The second part of the thesis thus focuses on our proposed architecture, where we are leveraging signal processing principles to offload the computation tasks onto user smartphones, consequently minimizing data sharing with service providers. On the server side, we deploy the service application in a trusted computation environment to preserve the service usability level.

Abstract

In the third part of the thesis, we address the problem associated with the restricted access and sharing of geospatial mobility datasets due to the growing privacy concerns. Yet mobility datasets are fundamental for algorithmic design, analysis and experimental reproducibility and thus paramount for facilitating geographical information systems research. In this part of the thesis, we focus on capturing geospatial mobility datasets with two different approaches: (1) synthesizing mobility trajectories by applying deep learning and statistical learning techniques to replicate mobility behaviors, and (2) collecting real-world geolocation traces through a mobile application. With regards to the first technique, We analyze six different deep-learning architectures on their ability to learn and memorize trajectory patterns and propose metrics to quantify the realism of the generated synthetic trajectories. Secondly, we propose a large-scale mobility dataset, *Breadcrumbs* collected in the city of Lausanne, Switzerland, which addresses problems associated with the current mobility datasets, such as irregular location sampling rate, single sensor tracking and lack of ground truth information. We release this dataset along with the ground-truth information to the community in order to facilitate research in the direction of supervised human mobility learning schemes.

Key words: Human mobility modeling, point of interest retrieval, next-place prediction, synthetic geospatial data, mobility meta-attributes

Résumé

Dans cette thèse de doctorat, nous nous focalisons sur trois aspects majeurs de l'analyse de la mobilité humaine : (1) la modélisation de la mobilité humaine, (2) la prédiction de la mobilité, et (3) la collecte de traces de mobilité. Afin de mieux comprendre les différents aspects qui conditionnent et influencent la mobilité humaine, nous utilisons des principes issus de la physique, des statistiques et de la théorie de l'information. Bien que la mobilité humaine soit considérée de nature Markovienne, nous prouvons dans cette thèse que les processus Markoviens ne doivent être utilisés que dans certains cas uniquement. Des modèles plus réalistes doivent être donc utilisés pour répondre à des problématiques de planification urbaine, de télécommunication, de prévisions de trafic, de diffusion de maladies infectieuses, etc. L'optimisation de ces services est bien évidemment facilitée par l'obtention de larges volumes de données géo-localisées via des smartphones équipés de connexion Internet et de système de géolocalisation par satellites (GPS). En considérant que la mobilité humaine varie fortement d'un individu à l'autre et que les exigences des services sont diverses, la première partie de cette thèse se focalise naturellement sur la modélisation de la mobilité. Cette première partie contient deux principales contributions : (1) la proposition d'un ensemble de méta-attributs pour quantifier la dynamique de la mobilité humaine et (2) la proposition d'un ensemble de critères pour sélectionner une technique de modélisation qui permet d'obtenir un compromis optimal entre performance et complexité. Dans cette partie, nous avons également réalisé une étude approfondie de la littérature sur la modélisation de la mobilité à partir de 1680 articles scientifiques publiés entre 1999 et 2019.

Dans la seconde partie de la thèse, nous présentons la conception et l'implémentation d'une architecture permettant de prédire la mobilité d'individus tout en préservant leur vie privée de bout en bout. L'utilisation croissante des services géo-localisés (LBS) et des objets personnels connectés ont facilité la collecte et la traçabilité de données géo-localisées. L'agrégation de ces données à grande échelle a également permis l'intégration de nouveaux paradigmes dans les applications afin d'améliorer ces services. La prédiction de la mobilité est donc devenue un concept incontournable permettant de personnaliser et d'améliorer des applications telles que la publicité ciblée, l'optimisation du co-voiturage ou encore le transfert de réseau. Cependant, le partage de données sensibles, telles que les localisations d'individus, avec les fournisseurs de services, ont mis au jour plusieurs problématiques liées à la vie privée. Les techniques actuelles de préservation de la vie privée (obscurcissement et anonymisation des localisations par exemple) ont un fort impact sur l'utilisation de ces services et réduisent considérablement la qualité d'utilisation de ceux-ci. La seconde partie de cette thèse se focalise donc sur cette proposition d'architecture, dans laquelle nous avons utilisé des principes issus du traitement du signal afin

de réduire les tâches de calcul directement sur les smartphones des utilisateurs afin de réduire le partage de données sensibles avec les fournisseurs de services. Du côté serveur, nous déployons l'application du service dans un environnement de calculs approuvé et fiable pour préserver la qualité d'utilisation du service.

Dans la troisième partie de la thèse, nous explorons le problème des accès restreints aux datasets contenant des données sensibles géo-spatiales de mobilité d'individus et du partage de ceux-ci en lien avec les enjeux de vie privée. Ces datasets sont essentiels pour concevoir de nouveaux algorithmes, effectuer des analyses, reproduire des tests scientifiques, et faciliter la recherche dans le domaine des systèmes d'information géographiques. Dans cette partie de la thèse, nous nous focalisons sur la collecte des données géo-spatiales avec deux approches différentes : (1) la génération de trajectoires de mobilité en utilisant des techniques d'apprentissage profond (i.e., deep learning) et de statistiques afin de reproduire des comportements de mobilité, et (2) la collecte de données géo-localisées réelles à l'aide d'une application mobile. En ce qui concerne la première approche, nous analysons six différentes architectures d'apprentissage profond permettant d'apprendre et de mémoriser des motifs de mobilité et proposons des mesures pour quantifier le réalisme des trajectoires synthétiques générées. Dans la seconde approche, nous présentons un dataset de données géo-localisées, appelé Breadcrumbs, collecté dans la région de Lausanne (Suisse). Ce dataset vise à combler des manques identifiés dans les datasets existants, tels que l'irrégularité d'échantillonnage de données géo-localisées, le manque de diversité dans l'utilisation de capteurs et l'absence de points d'intérêt vérifiés par les individus eux-mêmes (i.e., ground-truth). Nous mettons ce dataset à disposition de la communauté scientifique afin de permettre de nouvelles perspectives d'analyses supervisées dans le domaine de l'étude de la mobilité humaine.

Mots-clés : modélisation de la mobilité humaine, recherche de points d'intérêt, prédiction de la prochaine localisation visitée, données géo-spatiales synthétiques, méta-attributs de mobilité

Contents

Introduction

Background and Motivation

While the term *mobility* has several meanings, in the context of this thesis it refers to the movement of human beings in space and time. Human mobility has been fundamental for societal development from the migration of the *Homo sapiens* out of the African continent 70,000 years ago, through the discovery of the European *New World* to the contemporary *Indus valley civilization* in the 3300 BC. These migratory patterns primarily driven by food scarcity, climatic and warfare factors significantly influenced dissemination of cultural, economic and technological ideas along the route. The long migratory flows in the hunter gatherer era have been either replaced today by rapid commutes in large metropolitan areas or completely phased out due to the advancement of communication technologies. Although the impact of physical mobility on today's socio-economic setting has lowered, the applications based on short-term commutes have manifested in the technological progress.

Understanding and modeling mobility has become an integral part of modern day applications that span diverse domains such as urban planning, telecommunication network optimization, traffic forecasting and ride-sharing services. Such services are facilitated by large volumes of geolocation data (e.g., GPS traces, mobile phone records, social media records) generated by the daily movements of large and growing crowds through their mobile devices. The proliferation of location-based services (LBS) and the inclusion of tracking technologies in personal wearable devices has also resulted in collections of geolocation data to further enhance and personalize their services. Such services extend location-aware service provisioning to areas such as recommendation systems, route optimization and advertising, etc. A comprehensive understanding of the underlying human mobility dynamics is therefore critical for the advancement of modern day services.

In conjunction with the aggregation of large volumes of user geolocation data and the progress of computational tools in extracting knowledge from these digital traces, an indelible volume of information regarding the user whereabouts is exposed to the service providers. Location traces are not only a set of positions on a map but can reveal a large amount of information about the individuals' activities and habits, such as social relationships, political leaning and religious affiliation. Advanced data storage facilities combined with sophisticated data mining

and deep learning techniques are used by governments or private corporations to profile users for financial, security or any other strategic reasons. As the volume of these sensitive location traces collected from users increases along with the potency of inference algorithms added with the lowering costs of data storage, the ill effects of privacy loss become more and more apparent. For instance, it has recently been shown that user privacy is not preserved in aggregated mobility datasets [251]. As a result, our motivation stems from realizing location-aware services that can leverage highly representative mobility models while preserving user privacy. Thus, in this thesis, we thus focus on achieving high realism in the formulated mobility models, realizing privacy-aware location-based services and facilitating mobility modeling research by providing (synthetic) datasets and tools.

Dissertation Objectives and Research Questions

Given the importance of understanding mobility dynamics for improving location-aware mobile applications, this thesis studies human mobility and addresses the consequences of dealing with sensitive user location information. In particular, we focus on three perspectives of human mobility, categorized in three parts: (I) modeling human mobility, (II) predicting human mobility, and (III) capturing human mobility. Each perspective constitutes a separate part of the thesis. We describe our objectives and research questions associated with each of the perspectives below.

Modeling Human Mobility

Modeling human mobility can be defined as quantifying the underlying regularities associated with human movement within a set of locations. Mobility in general can be viewed as a stochastic process defined by observable and latent variables. The observable variables correspond to the geolocation coordinates and the associated timestamps, whereas the latent variables represent individual behavioral dynamics. Some examples of these mobility behaviors include periodicity of movement, stay duration at points of interests, commute times between different locations, influence of transportation mode, etc., which are governed by individual habits, tastes, social relationships and geographical area. Accurately modeling human mobility implies quantifying the interplay between the observable and latent variables.

In this part of the thesis, we explore these interactions to study the characteristics of human mobility by adopting a data-driven approach. Mobility modeling in itself is a large domain containing several subsets such as models for position, models for movement, population-based models, individual-based models, semantic trajectory modeling, modeling movements based on visual analytics. In this thesis, we explicitly focus on modeling individual next stop movements and modeling points of interests within large-scale geolocation trajectories. In order to gain a deeper understanding of the processes governing human mobility along these two domains, we draw out parallels between information theoretic measures, statistical physics. We also leverage the underlying theory behind the domain of natural language processing to compute improved

entropy estimates of human mobility trajectories. The formulated models are assessed on their effectiveness to forecast the next user location(s) and their precision to infer points of interests from location trajectories. The key objectives and research questions associated with each of them in this part of the thesis are listed hereafter.

Objective 1. To conduct a systematic and comprehensive literature review on human mobility modeling to identify key trends and shortcomings.

Q1. Which techniques/approaches are commonly used to model human mobility?

Q2. Is there any preference between the type of the dataset (GPS, GSM, WiFi), spatiotemporal granularity and the modeling technique chosen?

Objective 2. To identify the legitimate source of empirical gains (forecasting accuracy) in next-stop modeling in the state-of-the-art human mobility models.

Q3. What is the contribution of a particular modeling approach or an architectural amendment on next-stop modeling efficacy derived from the proposed model?

Q4. What is the influence of the mobility dataset used to quantify the model performance and the evaluation methodology selected on the empirical gains derived from the proposed model?

Q5. What is an ideal evaluation methodology to quantify the performance of a mobility model to eliminate all biases and facilitate cross-model comparison?

Objective 3. To quantify human mobility dynamics by identifying mobility meta-attributes that facilitate the modeling technique selection procedure.

Q6. Is it possible to decide a priori on the most appropriate technique to model individual next-stop movements for a certain mobility dataset?

Q7. Which meta-attributes best represent the characteristics of the mobility trajectories in a dataset to driven the model selection process?

Objective 4. To revisit and validate the widely accepted assumptions regarding human mobility dynamics.

Q8. Is human mobility dynamics truly governed by Markovian process and possesses a memoryless structure or is it a consequence of the dataset characteristics?

Q9. Does the currently accepted approach to estimate the temporal mobility entropy achieve asymptotic convergence?

Q10. Is the currently-used upper-bound limit of mobility predictability still valid given the advancement of deep learning techniques?

Objective 5. To eliminate the dependence on *a priori* determined parameters for extracting user points of interests from their trajectory datasets.

Q11. How to extract points of interest without relying on a priori determined parameters such as spatiotemporal bounds or behavioral criteria?

Predicting Human Mobility

There has been a growing trend to integrate location-based services (LBS) into mobile applications facilitated by mobile internet access, global positioning and intuitive graphical interfaces. LBS technology market could be segmented into several application usages such as navigation, search and recommendation, advertising, tracking and infotainment, gaming and augmented reality. Information regarding the next location of an individual could be integrated into the LBS, in order to improve the performance of several applications in the domain of resource utilization, content retrieval and handover latency optimization. While LBS providers are continuously tracking user location information to formulate the next-place prediction models to enrich our mobile experience, major privacy concerns are raised. Applying advanced data mining and deep learning approaches on large volumes of user location traces can reveal the attached contextual information such as their religious and political affiliations and social relationships. The existing design methodologies to preserve user privacy, such as location obfuscation, anonymisation and cryptographic primitives comes at the cost of reduced application utility.

This part of the thesis focusses on formulating a privacy-aware location prediction approach to safeguard user location-privacy while preserving service utility. We propose and implement amendments on the user's end and at the service provider's end to device an end-to-end privacy preserving approach. Our objectives in this part are listed hereafter.

Objective 6. To minimize the volume of data required to achieve satisfactory levels of prediction accuracy in order to shift server side tasks onto user smartphones, consequently minimizing the sharing of sensitive location traces with service providers.

Q12. What is the computational complexity involved in executing the points of interest extraction technique on the user smartphone?

Q13. How to lower the complexity involved in extracting these points while maintaining a sufficient level of accuracy?

Q14. What is the resulting tradeoff and the power consumption of offloading this task at the user's smartphone?

Objective 7. To formulate a mobility prediction system based on realtime location data-streams to eliminate long waiting times before the prediction model kicks in on mobile devices.

Q15. What is the minimum volume of historical trajectory data necessary to achieve a satisfactory mobility prediction accuracy while minimizing the computational complexity involved?

Q16. How could temporally evolving user mobility behaviors be integrated in a real-time mobility prediction model without overlooking on the small time-bounded user movements?

Objective 8. To quantify the overhead involved in deploying the service providers application in a trusted execution-environment.

Q17. What is the computational complexity involved when deploying a typical location-based service in a hardware based trusted execution-environment as opposed to its bare-metal implementation?

Capturing Human Mobility

Mobility datasets are fundamental for evaluating algorithms pertaining to geographic information systems and facilitating experimental reproducibility. Privacy implications however, restrict sharing such datasets, as even aggregated location-data is vulnerable to membership inference attacks. In this thesis, we address this problem through the means of two approaches: (1) simulating mobility trajectories, and (2) collecting mobility trajectories. Current synthetic trajectory generators attempt to superficially match *a priori* modeled mobility characteristics which do not accurately reflect the real-world characteristics. Modeling human mobility to generate synthetic yet semantically and statistically realistic trajectories is therefore crucial for publishing trajectory datasets having satisfactory utility level while preserving user privacy. Publicly accessible mobility datasets are usually not adequate for large scale experimental evaluations, compromising scalability tests. This issue incentivizes synthetic mobility trajectory generators that simulate the behavior of moving objects required to attain comprehensive performance valuations.

Currently available real-world mobility datasets are also restricted to geospatial information collected through a single sensor at low spatiotemporal granularities. The passively collected data also lacks ground-truth information regarding points of interest and their semantic labels. These features are critical in order to push the possibilities of geospatial data analysis towards analyzing mobility behaviors and movement patterns at a fine-grained scale. This part of the thesis focuses on the objectives and research questions discussed hereafter.

Objective 9. To simulate human mobility trajectories given a real-world dataset using recurrent neural architectures (RNNs), generative adversarial networks (GANs) and nonparametric copulas and to benchmark their performance and the associated trade-offs.

Q18. Which metrics should be utilized to quantify the realism of the synthetic mobility trajectories with respect to the real-world mobility data?

Objective 10. To collect a real-world mobility (dataset) trajectories which addressees some of the concerns depicted by the currently available public mobility datasets.

Q19. How to ensure a constant sampling rate from multiple geolocation sensors throughout the duration of the mobility data collection campaign?

Q20. How to collect ground-truth information from the participants while minimizing the biases involved?

Contributions and Structure

This thesis is structured as a collection of nine peer-reviewed articles that were published in conference proceedings and journals in the field of computer science and statistical physics. We group these nine articles according to the three perspectives on human mobility presented earlier. These articles are self-contained with a formal introduction of the terminology used in each and thus could be read comprehensively and independently from the others. The limitations and avenues for future work associated with each of the perspectives are also discussed in their respective chapters. Consequently, there exists minor content overlap within the chapters and slight differences between the terminology in order to suit the venue of the publication and their respective audience.

Contributions to Modeling Human Mobility

In Chapter 1, we focus on the first three objectives and answer Question 1-7. We highlight the inconsistencies and pitfalls in human mobility modeling and prediction research through a large scale systematic literature review. Through this survey, we systematize knowledge and provide guidelines towards performing credible mobility modeling research. We expose the consequences of relying on data-agnostic model selection and adopting inaccurate validation methodologies through experiments on three real-world mobility datasets. In order to address these problems, we propose four meta-attributes, that can accurately characterize a mobility dataset for selecting an appropriate modeling technique. Through a range of experiments, we show the applicability of our data-driven approach of model selection and analyzed the accuracy vs. complexity trade-offs associated with each. This chapter is based on the extended version of the following article.

Vaibhav Kulkarni, and Benoît Garbinato. 20 Years of Mobility Modeling and Prediction: Trends, Shortcomings and Perspectives. In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2019 [Full Version: arXiv preprint arXiv:1906.07451]

In Chapter 2, we scrutinize the widely accepted assumptions (Objective 4) governing human mobility dynamics. We demonstrate the non-Markovian character in human mobility by conducting the statistical tests which confirm the emergence of scaling laws in the distributions of dwelling times and inter-event times. We show that mobility trajectories contain scale-invariant long-distance dependencies similar to natural languages unaccounted for by the current upper bound computation methodology. This chapter is based on the following article.

Vaibhav Kulkarni, Abhijit Mahalunkar, Benoît Garbinato, and John D. Kelleher. Examining the Limits of Predictability of Human Mobility. Entropy 21, no. 4 (2019): 432

In Chapter 3, we address Objective 5 wherein propose a technique to detect points of interest

from user trajectories without relying on any *a priori* assumptions. We depict the bias resulting due to the stringent parameter bounds while extracting user points of interest. We also depict the problems arising from such bounds that are based on non-empirical calculations and extended to operate on some other datasets containing users having different mobility behaviors. We address this problem by treating user movements as spatiotemporal signals, effectively converting the point of interest detection to a peak-detection problem by using signal-processing algorithms. This chapter is based on the following article.

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, p. 79. ACM, 2017

Contributions to Predicting Human Mobility

In Chapter 4, we address Objective 6 and to facilitate the trend of on-board processing at the user's end by leveraging properties of spatiotemporal signals to reduce the computational complexity and power consumption. We demonstrate that the transformed spatiotemporal signals not only preserve all the key knowledge contained in the trajectories but also formulate the mobility models with a high degree of accuracy. We perform the complexity and power consumption analysis by implementing our approach on a DSP chip commonly present in many smartphones. This chapter is based on the following article.

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, p. 8. ACM, 2016 (*co-primary authors)*

In Chapter 5, we design and implement a real-time mobility prediction system, to provide swift next place predictions on hand-held devices (Objective 7). Our approach couples the prediction system with dynamic user mobility behaviors to restrict the data required for model training to short durations as opposed to conventional training approaches. We also evaluate the computational cost associated with our approach and theoretically validate the feasibility to operate on a mobile device. This chapter is based on the following article.

Vaibhav Kulkarni, Arielle Moro, Bertil Chapuis, and Benoît Garbinato. Capstone: Mobility modeling on smartphones to achieve privacy by design. In 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications (TrustCom), pp. 964-971. IEEE, 2018

In Chapter 6, we demonstrated the applicability of a hardware-based trusted execution-environment, i.e. Intel SGX to offer a privacy preserving location-based service. We implement a point of

interest-locator application using the security guarantees offered by SGX, adopting a privacy-by-design principle. We quantify the overheads involved due to the SGX implementation and compare it with the bare-metal execution. We show that SGX-based approach leads to a marginal overhead and provides near-to-the-perfect results. This chapter is based on the following article.

Vaibhav Kulkarni, Bertil Chapuis, and Benoît Garbinato. Privacy-preserving location-based services by using Intel SGX. In Proceedings of the SenSys Workshop on Human-centered Sensing, Networking, and Systems, pp. 13-18. ACM, 2017

Contributions to Capturing Human Mobility

In Chapter 7, we address Objective 9 where we propose architectural amendments to recurrent neural network architectures (RNNs) and evaluate their efficacy to synthesize mobility trajectories. This chapter is based on the following article.

Vaibhav Kulkarni, and Benoît Garbinato. Generating synthetic mobility traffic using RNNs. In Proceedings of the SIGSPATIAL Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery, pp. 1-4. ACM, 2017

Chapter 8 also addresses Objective 9 by evaluating discriminative models including generative adversarial networks (GANs) and nonparametric copulas to synthesize geolocation traces given a real-world geospatial dataset. We also propose several metrics to assess the performance of these models to learn, memorize and regenerate trajectories with characteristics comparable to real-world trajectories generated by moving entities. This chapter is based on the following article.

Vaibhav Kulkarni, Natasa Tagasovska, Thibault Vatter, and Benoît Garbinato. Generative Models for Simulating Mobility Trajectories. In proceedings of Neurips Workshop on Control and Decision Making in Spatiotemporal Domain, 2018

In Chapter 9, we propose a feature rich geolocation mobility dataset containing demographic attributes, contact and calendar records, social relationships, along with the ground truth and semantic labels for the points of interest. We describe the complete data collection process and our methodology to collect ground-truth information. We specify the use cases, applicable research domains and validation methodologies using the unique features present in our dataset. we perform a comparative study of four clustering approaches to extract points of interest cluster from GPS trajectories. We also propose a validation methodology while using the ground-truth labels. This chapter is based on the extended version of the following article.

Arielle Moro, Vaibhav Kulkarni, Pierre-Adrien Ghiringhelli, Bertil Chapuis, Kevin Huguenin and Benoît Garbinato. Breadcrumbs: A Rich Mobility Dataset with Point of Interest Annotations. In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2019 [Full Version: arXiv preprint arXiv:1906.12322]

Research Methodology

This research incorporates several domains in computer science including geographical information systems (GIS), signal processing, machine learning, information security and privacy. Computer science discipline focusses on several issues from the technical perspective such as theoretical modeling, numerical analysis, data structures and algorithms, manipulating relationships between different pieces of software, interaction between the software and hardware components etc. In addition to computer science, this thesis also contains some elements of Information Systems discipline. The field of Information Systems is concerned with the interaction between social and technological issues. Considering the diverse set of fields involved, we applied a design science research methodology for the research process as illustrated in Figure 1. The design science paradigm introduced by Alan Henver [105] is the foundation of the information systems discipline, positioned at the confluence of people, organizations, and technology. This methodology seeks to extend the boundaries of human organizational capabilities by formulating novel and innovative artifacts.

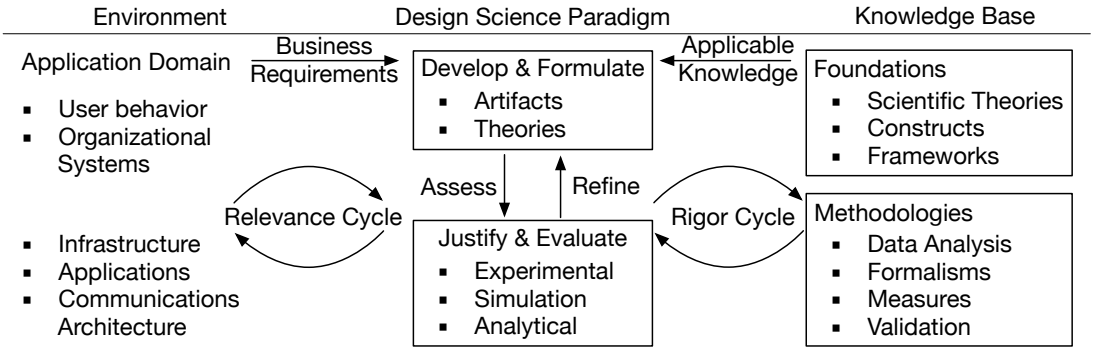


Figure 1 – Design Science Research Methodology

The overall reflexion of this thesis with regards to taking a step backwards to revisit the widely accepted notion surrounding human mobility and offloading traditional server side tasks on to user devices has been heavily influenced by workshops in design science and design thinking. As a result, from the relevance science perspective, this reflexion aims at the alignment of research in human mobility modeling and location-aware mobile computing with the requirements of the end users. The need for such an alignment resulted in the identification of the research questions proposed by this thesis and the related opportunities. The rigor cycle perspective aims at exploring the existing knowledge to solidify the identified problems and their existing solutions with the goal of incremental improvements in science. Following this approach, each of our

objectives started out with a systematic literature survey to lay out the existing contribution by the community in order to identify research gaps and area of improvements. For instance, our data-driven mobility modeling framework relies heavily on the knowledge arising from our comprehensive literature review spanning last two decades. This approach helped us identify the trends in the current state of research in the domain, related shortcomings and ideate on their respective solutions. The design science perspective aims at iterative artifact formulation and evaluation while continuously adapting results from the relevance and rigor cycles.

Our contribution are mainly published at venues related to computer science and statistical physics that demand formal and empirical methods to formulate and evaluate the proposed artifacts. In line with design science paradigm and the requirements of the respective domain, our process initiated with problem identification and model definition formulated in mathematical terms. The next step consisted of ideating original algorithmic solutions or architectural amendments to specifically address the problem. The evaluation criteria were selected from well established community guidelines related to algorithmic evaluations consisting of theoretical proofs or empirical assessments. In certain cases, wherein the evaluation metrics were not clearly defined, we followed the rigor cycle to ideate and access the related metrics. From the empirical perspective, the proposed solutions were implemented and compared with existing solutions on either real-world or synthetic datasets by adopting the set evaluation criterions. The solution ideation evaluation process is iterative and continuous stemming from the results driven from the rigor cycle which enabled us to quantify the performance of the proposed solution and perform successive model refinements accordingly.

Supplementary Research and Related Publications

The collection of articles considered in this thesis does not include all the work that was done during this research. For instance, aggregating large volumes of location trajectories of moving entities necessitate development of scalable indexing strategies. Novel location-based services at scale can only be realized in practice if the predicted locations and trajectories could be efficiently indexed and queried. Therefore, during our research, we addressed some of these issues associated with rapid indexing and querying of trajectories from moving object databases. Results from this research thread can be found in the following paper.

Bertil Chapuis, Arielle Moro, Vaibhav Kulkarni, and Benoît Garbinato. Capturing complex behavior for predicting distant future trajectories. In Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems, pp. 64-73. ACM, 2016

In addition to the technical aspects of preserving user privacy while using location-based services, we also focus on quantifying location information disclosure behaviors of users in order to improve the service design and architecture. Quantifying the disclosure behaviors and understanding user motivations while using location-based services can aid in better application design decisions

and incorporating user requirements in privacy-aware systems. To this end, we provide a more nuanced conceptualization of location-data disclosure behavior based on a survey conducted in US and Germany totaling 1050 individuals. Our proposed model views location disclosure across four distinct dimensions; (1) purpose, (2) sensitivity, (3) extent, and (4) sharing parties to account for the recent data privacy regulations, emerging data streaming economy and interdependent (extrinsic) privacy risks. Results from this research thread can be found in the following paper.

Dana Naous, Vaibhav Kulkarni, Christine Legner, and Benoît Garbinato. Location-Information Disclosure: A Multi-Dimensional Privacy Calculus Model. In the proceedings of International Conference in Information Systems (ICIS), 2019

Complete list of Publications

Hereafter, we list all the publications that resulted from the research carried out in the context of this thesis. As already stated, only nine of articles constitute the core of this thesis.

1. *Vaibhav Kulkarni, and Benoît Garbinato. 20 Years of Mobility Modeling and Prediction: Trends, Shortcomings and Perspectives. In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2019*

2. *Dana Naous, Vaibhav Kulkarni, Christine Legner, and Benoît Garbinato. location-Information Disclosure: A Multi-Dimensional Privacy Calculus Model. In the proceedings of International Conference in Information Systems (ICIS), 2019*

3. *Arielle Moro, Vaibhav Kulkarni, Pierre-Adrien Ghiringhelli, Bertil Chapuis, Kevin Huguenin and Benoît Garbinato. Breadcrumbs: A Rich Mobility Dataset with Point of Interest Annotations. In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2019*

4. *Vaibhav Kulkarni, Abhijit Mahalunkar, Benoît Garbinato, and John D. Kelleher. On the Inability of Markov Models to Capture Criticality in Human Mobility. In the proceedings of 28th International Conference on Artificial Neural Networks (ICANN). Springer, Cham, 2019*

5. *Vaibhav Kulkarni, Abhijit Mahalunkar, Benoit Garbinato, and John D. Kelleher. Examining the Limits of Predictability of Human Mobility. Entropy 21, no. 4 (2019): 432*

6. *Vaibhav Kulkarni, Natasa Tagasovska, Thibault Vatter, and Benoît Garbinato. Generative Models for Simulating Mobility Trajectories. In proceedings of Neurips Workshop on Control and Decision Making in Spatiotemporal Domain, 2018*

7. *Vaibhav Kulkarni, Arielle Moro, Bertil Chapuis, and Benoît Garbinato. Capstone: Mobility modeling on smartphones to achieve privacy by design. In 17th IEEE International*

Conference On Trust, Security And Privacy In Computing And Communications (TrustCom), pp. 964-971. IEEE, 2018

8. *Vaibhav Kulkarni, and Benoît Garbinato. Generating synthetic mobility traffic using RNNs. In Proceedings of the SIGSPATIAL Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery, pp. 1-4. ACM, 2017*

9. *Vaibhav Kulkarni, Bertil Chapuis, and Benoît Garbinato. Privacy-preserving location-based services by using Intel SGX. In Proceedings of the SenSys Workshop on Human-centered Sensing, Networking, and Systems, pp. 13-18. ACM, 2017*

10. *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 25^{th} ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, p. 79. ACM, 2017*

11. *Bertil Chapuis, Arielle Moro, Vaibhav Kulkarni, and Benoît Garbinato. Capturing complex behavior for predicting distant future trajectories. In Proceedings of the 5^{th} ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems, pp. 64-73. ACM, 2016*

12. *Vaibhav Kulkarni*, Arielle Moro*, and Benoît Garbinato. Mobidict: A mobility prediction system leveraging realtime location data streams. In Proceedings of the 7^{th} ACM SIGSPATIAL International Workshop on GeoStreaming, p. 8. ACM, 2016 (*co-primary authors)*

Modeling Human-Mobility Part I

1 20 Years of Mobility Modeling & Prediction: Trends, Shortcomings & Perspectives

Abstract

In this paper, we present a comprehensive survey of human-mobility modeling based on 1680 articles published between 1999 and 2019, which can serve as a roadmap for research and practice in this area. Mobility modeling research has accelerated the advancement of several fields of studies such as urban planning, epidemic modeling, traffic engineering and contributed to the development of location-based services. However, while the application of mobility models in different domains has increased, the credibility of the research results has decreased. We highlight two significant shortfalls commonly observed in our reviewed studies: (1) data-agnostic model selection resulting in a poor tradeoff between accuracy vs. complexity, and (2) failure to identify the source of empirical gains, due to adoption of inaccurate validation methodologies. We also observe troubling trends with respect to application of Markov model variants for modeling mobility, despite the questionable association of Markov processes and human-mobility dynamics. To this end, we propose a data-driven mobility-modeling framework that quantifies the characteristics of a dataset based on four mobility meta-attributes, in order to select the most appropriate prediction algorithm. Experimental evaluations on three real-world mobility datasets based on a rigorous validation methodology demonstrate our frameworks ability to correctly analyze the model accuracy vs. complexity tradeoff. We offer these results to the community along with the tools and the literature meta-data in order to improve the reliability and credibility of human mobility modeling research.

Keywords: Systematic literature review; Data-driven modeling; Meta-attributes.

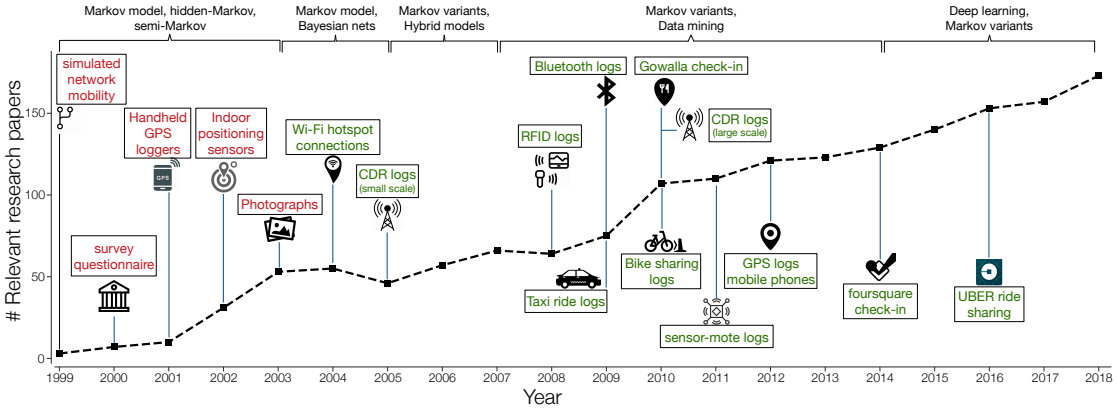


Figure 1.1 – Mobility modeling and prediction: 20 years in review. The figure presents a summary of the review, highlighting the key techniques dominant (in terms of number of papers) in the respective era and the dataset (private datasets in red, public in green) driving this research.

1.1 Introduction

Over the last two decades, we have seen a large number of studies on human mobility modeling and prediction by the Geographic Information Systems (GIS) community. This testifies of the importance of mobility prediction in context-aware systems where a user's future location is used to seamlessly trigger service execution. These systems span services such as ride sharing, traffic prediction, point of interest recommendation, resource/urban planning, and network optimization among others. Given the sensitive nature of mobility trajectories and the enforcement of binding privacy regulations, privacy-preserving modeling approaches such as federated learning [223] and Google Rappor are advancing. In such approaches, the process of model training and updating is executed locally on resource-constrained smartphones [223]. A heuristically driven framework that analyzes the model performance vs. complexity trade-off for algorithmic selection is therefore essential.

To understand the current methodologies driving prediction-model selection and performance validation strategies, we performed a systematic literature review spanning last two decades amounting to 1680 articles. Based on the reviewed literature, mobility modeling an be defined as the process of estimating the probability distribution over an individual's future movement by minimizing the negative log-likelihood over the currently known user trajectory. Research in this domain can thus be classified in three distinct categories: (1) theoretical modeling of mobility dynamics, (2) quantifying the uncertainty in next-place prediction, and (3) leveraging stochastic optimization algorithms to model human-mobility and benchmark the next-place forecasting capability. This paper focuses on the third category, where the stochastic approaches applied for constructing the next-place forecasting model fall into three categories: (1) Markov model variants (2) data mining techniques, and (3) neural network architectures. In order to validate the model's predictive performance, several types of datasets are used in these works that either contain GPS trajectories of pedestrians, recurrent WiFi connections, Bluetooth records or social

network check-ins.

Despite the large number of studies, it is not trivial either to objectively compare cross-model performance, nor to identify the source of the empirical gains provided by the proposed models. This difficulty stems from the fact that each modeling approach is implemented with distinct search heuristics which introduces a range of inductive biases [156]. This results in delivering different performance depending upon the dataset attributes [189]. We find that for a majority of instances, empirical gains predominantly stem from erroneous validation methodology and selection of datasets with opportune characteristics rather than modeling/architectural amendments. Furthermore, the availability of several datasets and no explicit procedure to perform mobility-prediction validation makes it challenging to determine the appropriate learning algorithm. We observe that 68% of the reviewed work relies on variants of Markov models despite the unclear association between Markov processes and human mobility dynamics [134]. As presented in Figure 1.1, this trend declines to a certain extent after the onset of deep learning and the availability of the associated implementation frameworks.

Knowing that there is no single algorithm that can perform uniformly over all the modeling tasks on different datasets (no free lunch theorem) [249], it is essential to select the appropriate mobility-modeling algorithm depending upon the dataset characteristics. Meta-learning is a bias minimization perspective on learning algorithm selection, by accumulating meta-knowledge about the data [194]. Such an approach considers statistics inspired measures, such as skewness, entropy and autocorrelation, as a source for the definition of dataset characteristics. These characteristics are known as meta-attributes. Correctly estimating them speeds up and improves the mobility modeling pipeline design by achieving faster convergence, optimal local minimum and discernible models. We find a lack of research directed towards estimation of these meta-attributes for human mobility that can act as metrics characterizing the mobility datasets. To this end, we experimentally show the importance of selecting prediction models based on the meta-attributes and highlight the consequences of adopting misleading validation methodologies. Relying on robust cross-validation approaches for sequential data is also essential to present a fair contribution of a model and to perform intuitive comparison with other models. To this end, our paper makes contributions on the following fronts:

- Through a large-scale literature survey, we systematize knowledge on mobility-modeling research and provide our insights on how this research should be conducted and which challenges it should address. We offer the tools and the literature meta-data to the community with the hope of improving the credibility of mobility modeling research. [1]

- We propose four meta-attributes to quantify mobility dataset characteristics grounded on statistical and information theoretic primitives: (1) average length of an individual trajectory, (2) number of points of interest, (3) number of points of interests interacting with each other, and (4) distance between the interactions of points of interest.

[1] Link: https://bit.ly/2HRZGk5

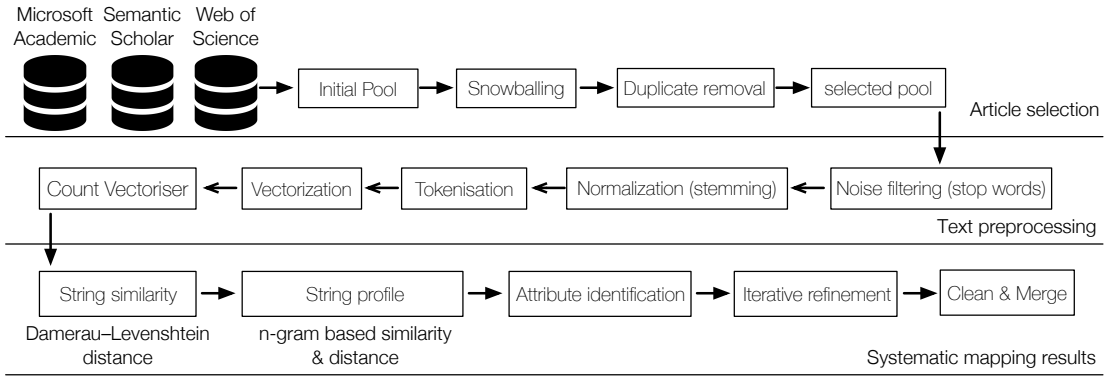


Figure 1.2 – Overview of the search & data extraction strategy, study selection and quality assessment methodology.

- Based on the meta-attributes, we present a mobility modeling framework to assist in selecting an appropriate learning algorithm. We then present a validation methodology to perform performance assessment of mobility prediction models allowing for a fair comparison and facilitating the process of incremental improvements. We evaluate the performance of our approach on three mobility datasets and discuss the associated accuracy vs. complexity trade-offs.

The remainder of the paper is organized as follows. Section 1.2 presents the survey methodology, followed by the survey findings in Section 1.3. Section 1.4 highlights the shortcomings of the current mobility modeling approaches. In Section 1.5, we present our data-driven mobility modeling framework, followed by the results and discussion in Section 1.6. We conclude the paper in Section 1.7.

1.2 Survey Methodology

The increasing number of articles published on human mobility-modeling and prediction testifies to the growing interest from researchers and practitioners. Figure 1.1 shows the most significant prediction models and the datasets used for mobility prediction for a given span of year(s). The goal of the systematic literature review is to classify the state-of-the-art in the area of human mobility modeling and answer the following research questions: (1) which techniques are used for constructing the next-place prediction models?, (2) which methodologies are adopted to validate the performance of the models?, and (3) which datasets are used to perform the performance quantification?.

Despite the large number of hits on these topics, there is a dearth of coherent understanding on what kind of studies have been conducted under the term *human mobility modeling*, with which methods, what kind of results they yield, and under which circumstances. Thus in this section, we describe the methodology we adopted for source selection, search keywords, followed by quality

assessment and application of article inclusion, exclusion criteria. Schreckenberger et al. [210] perform a similar survey spanning from 2013 to 2018 and present the descriptive statistics with regards to the data types and techniques used. On the contrary, our study spans the last 20 years and we not only provide the overview, but we perform an in-depth study on the impact of datasets, techniques, validation methodologies and experimentally validate the problematic trends, In addition, we also conduct experiments on real-world mobility datasets to expose the pitfalls in the currently adopted techniques and propose amendments.

1.2.1 Search Strategy & Article Selection

To find relevant studies, we searched three major academic article search engines: Web of Science, Microsoft Academic Search and Semantic Scholar. Three distinct platforms were chosen to ensure result completeness, as a single platform does not cover all major publisher venues. The search terms included, *human-mobility prediction, human-mobility modeling, next place forecasting, Predicting Significant locations* among others. The search terms were used for the title, abstract and keywords. These search domains consisted of *mobile and ubiquitous computing, geographic information systems and knowledge discovery and data mining*. Furthermore, only the studies that were peer reviewed, published in an international venue, written in English language and that were electronically available were included. Additional papers were identified by using the citation and reference list of a given paper to minimize the risk of omitting relevant studies. This step is according to the guidelines recommended by Keele et al. [121] to conduct a systematic literature review and is known as selected snowballing. We selected the top five cited articles in the domain of human mobility prediction [12, 76, 159, 168, 226] to bootstrap the snowballing procedure. At this stage, studies in the pool were ready for application of inclusion/exclusion criteria. We complete list of keywords, search strings, domains and the associated scripts are published along with the literature meta-data.

1.2.2 Preprocessing & Quality Assessment

To eliminate duplicates, the article titles were preprocessed by removing the stop words and performing normalization (stemming). String similarity was estimated by using Damerau-Levenshtein distance, which measures the edit distance between two sequences. To calibrate the distance threshold, we manually assessed the top 20 top cited articles and labeled them according to their relevance. The prediction technique used in each paper was identified by first normalizing the abstract followed by stop-words removal, tokenization, vectorization and using a count vectorizer to analyze relative frequency of the tri-grams [24]. This is a simple statistical approach that analyzes the position of word in the abstract, the term frequency and inverse document frequency. The preprocessing and quality assessment pipeline is summarized in Figure 1.2. This cleaned list of articles were used to perform the analysis presented in the next section of the paper.

Approach	Dataset (#participants, duration, type, location)	Validation methodology
Markov model [10]	1 participant, 4 months, GPS traces	Not specified
semi-Markov model [48]	10 participants, unknown duration , GPS, GSM, WiFi	Not specified
hidden Markov model [159]	GeoLife (182 participants, 5 years, GPS traces)	Leave one out validation
mixed Makov model [9]	Simulated data: 691 participants, 1hr;31 minutes, GPS	k-fold cross validation (k=10)
mobility Markov chain [76]	GeoLife, synthetic dataset, private dataset (6 researchers)	Holdout validation, 50% split train-test
extended mobility Markov chain [6]	1 participant, 54 weeks, CDR	Not specified
variable-order Markov model [253]	GeoLife (182 participants, 5 years, GPS traces)	Holdout, Random selection, 90%-10% train-test
hidden semi-Markov model [258]	Simulated data	No empirical validation
hierarchical semi-Markov model [18]	GeoLife (182 participants, 5 years, GPS traces)	Holdout validation (no split % specified)
spherical hidden-Markov model [272]	Simulated data, Twitter dataset (geotagged tweets)	Random selection 70%-30% train-test split
Adaboost-Markov model [247]	GeoLife (182 participants, 5 years, GPS traces)	Random selection 90%-10% train-test split

Table 1.1 – Different variants of Markov models used to model human-mobility, datasets used to corroborate the model performance and the different types of cross-validation strategies applied.

1.3 Survey Findings

The first application of human mobility prediction was in the context of ad-hoc wireless networks by Gerla [81]. The knowledge about the user's next-location was used to anticipate topological changes and minimize the connectivity disruption caused by mobility. Application of human mobility prediction in wireless networks became prominent after Su et al. in [230], proposed a location-aware routing scheme and demonstrated its effectiveness using simulations. The seminal work however, in the context of prediction location on road networks using GPS information, was made by Ashbrook et al. [10]. They first identified significant places (points of interest) by clustering the raw GPS trajectories and then built a Markov-based predictor to forecast the next significant place. In their next article, Ashbrook et al. [12] extended this approach to perform mobility prediction across a dataset having multiple users. Several approaches along these lines were presented in the subsequent years using geolocation datasets consisting of data-points from different mobile phone sensors [185, 186, 224].

The reviewed studies can be categorized based on the techniques applied for mobility prediction as follows: (1) Markov model variants, (2) neural network techniques, and (3) data-mining based approaches. Table 1.1 presents a meta-summary of the Markov model variants, the dataset used to quantify the model performance and the validation methodology used. A similar analysis based on neural network and data mining techniques is presented in Table 1.2 and Table 1.3 respectively. These studies have used several datasets differing with respect to the data type, number of users, collection duration and geographic regions. We also highlight that in several cases the datasets are obtained privately from the telecommunication operators or generated synthetically using unspecified mobility simulators.

Accessibility of larger datasets prompted development and application of several variants of predictors based on Markov model such as hidden Markov model [159], mixed Markov model [9], semi-Markov model [48], hidden semi-Markov model [258], mobility Markov chain [76], extended mobility Markov chain [6], variable order Markov model [256], hierarchical hidden Markov model[18] and spherical hidden Markov model [18]. Each of the variant claims to address and account for different aspects of mobility trajectories, such as missing data from some time intervals [258], correlation with the stay duration [48], memory requirement [256],

Approach	Dataset	Validation methodology
Attentional RNN [69]	Foursquare Check-ins Mobile apps (private) CDR data (private)	Holdout (80%-20% split)
Bi-dir. LSTM [262]	Didi dataset GPS data, (private)	Holdout (4-1 month)
Seq2Seq model [120]	MIT Reality dataset private dataset	Holdout (70%-30% split)
STF-RNN [5]	Geolife	3-fold cross validation
ST-RNN [152]	GTD dataset Gowalla dataset	70%-20% split, 10% validation

Table 1.2 – Neural network approaches used to construct mobility prediction models, datasets used to corroborate model performance and the respective cross-validation strategies adopted.

user behavioral characteristics [9], spatiotemporal associations with the path connecting the stay-points [18], semantic trace data [272] or location specific characteristics [159]. Other approaches, such as non-parametric Bayesian model were also leveraged for construction of mobility-prediction models [113]. Here, particle filters and expectation-maximization techniques were applied to forecast the next place of the users.

After the prominence of big data in 2008, several data mining techniques, which explore periodic patterns and association rules, were employed for mobility prediction. Monreale et al. [168] define a trajectory as an ordered sequence of time-stamped locations and propose a modified version of the a-priori algorithm based on sequence analysis. Such techniques do not generalize as well as state-space models, due to ignoring the notion of spatiotemporal distance. Another type of approach falls in the category of template matching, where extracted features from time-stamped sequences are compared with pre-stored templates, using similarity search metrics such as dynamic time warping [205]. Other variations such edit distance related metrics were also applied for template matching schemes [182].

After 2014, we observe an onset of application of neural network techniques for mobility prediction. The principle argument for applying deep learning networks is that the probabilistic models lack fine granularity in prediction and suffer from data sparsity problem. A recurrent neural network (RNN) based attention model for predicting the next place was proposed by Feng et al. [69]. This model is able to captures the multi-level periodicities present in human mobility. A bidirectional Long Short-Term Memory (LSTM) network architecture was proposed by Zhao et al. [262] to predict the trip destination location. This network provides higher attention to locations having strong correlations with the destination due to the attention mechanism. A sequence to sequence (Seq2Seq) approach was applied by Karatzoglou et al. [120] to human semantic trajectories, in order to improve prediction of semantically annotated trajectories. Al-Molegi et al. [5] proposed a space time feature-based RNN for predicting next user movement. The model operates by conditioning the RNN inputs based on the spatial features and temporal elements present in the trajectory. Liu et al. [152] proposed an extension of the vanilla-RNN model, where each layer is upon different time intervals and distance specific transition matrices for distinct geographical distances.

Approach	Dataset	Validation methodology
Template matching [205] (SW-local alignment)	Simulated data	Random selection, 30-fold validation
Template matching [182] (Trajectory alignment)	Simulated data	Not specified
Data Mining [168] (Decision tree)	Car trajectories	Wednesday-train, Thursday-Test
Data Mining [155] (Decision tree)	Geolife dataset Weibo (private)	Not specified

Table 1.3 – Different data mining approaches used to model human-mobility, datasets used to corroborate the model's performance and the types of cross-validation strategies.

In addition to the usage of different datasets for performance validation, we observe that the validation methodologies applied also differ widely from one another or bootstrapped using differing parameters. We categorize the currently used cross-validation techniques for assessing model predictability as: (1) random shuffling before holdout validation, (2) train-test split ratio ranging from 90%-10% to 50%, (3) 3-fold to 30-fold cross-validation, and (4) weekday/month/year-based splitting. In the last case, model training is typically performed on the first four days of the week and tested on the remaining three. We also found several occurrences where the validation methodology was not specified. Selecting an appropriate validation approach to correctly assess the model performance on a given dataset is imperative to draw legitimate conclusions. To the best of our knowledge, we did not find any argumentation to select a particular dataset or a particular validation strategy in the reviewed literature. In the next Section, we delve deeper in these shortcomings as we highlight that selecting arbitrary validation technique is detrimental to the incremental process of model improvement and provides misleading measures.

1.4 Shortcomings

In this section, we experimentally investigate the ill-effects of the shortcomings described in Section 1.3. We categorize these shortcomings in two major domains: (1) inefficient accuracy vs. complexity trade-off arising from data-agnostic prediction model selection, (2) inconclusive model performance quantification due to adoption of inaccurate validation methodologies. We also expose the systematic bias involved in the model assessment due to the selection of the dataset and validation methodology devoid of any heuristics.

1.4.1 Experimental Setup

Real world Mobility Datasets. We conduct the experiments by using three mobility datasets publicly available upon request. The PrivaMov dataset [112] was collected through GPS, WiFi and GSM sensors in the city of Lyon (France) and includes university students, staff and their family members. The Nokia mobile dataset [137] (MDC) was collected in the Lake Geneva region of Switzerland and consists young individuals' trajectories, collected through GPS, WLAN, GSM and Bluetooth sensors. The GeoLife dataset [266] was collected in Beijing (China) and

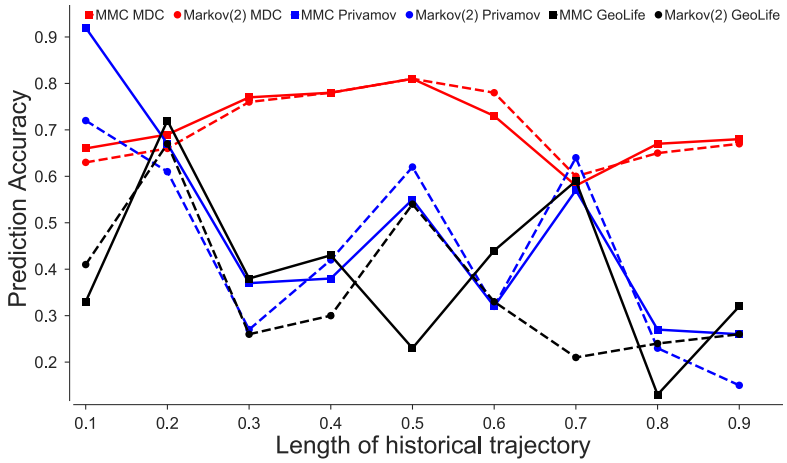


Figure 1.3 – Prediction accuracy of Markov models variants on the three datasets. The horizontal axis signifies the proportion of trajectory length considered for the train-test split and vertical axis signifies the precision of the prediction model.

contains trajectories recorded through GPS loggers and GPS-phones.

Mobility Prediction can be defined as forecasting the transitions between places, after eliminating all self-transitions. A preliminary step in achieving this consists of transforming the raw GPS locations into a sequences of points of interest (POIs). A POI is defined as any location where an individual visits with an intentional purpose with a perceived priority for e.g., home/work place, gym, train station etc. We use a POI extraction technique that is independent of *a priori* assumptions regarding the data and individual mobility behaviors [135]. We then convert the raw GPS trajectory of a user u, $T_u = \langle (lat_1, lon_1, t_1), (lat_2, lon_2, t_2) ... (lat_n, lon_n, t_n) \rangle$, where lat_i, lon_i are the latitude and longitude coordinates respectively and t_i is the timestamp such that $t_{i+1} > t_i$ into a sequence of temporally ordered points of interest, $s(t) = \langle (poi_1, t_1), (poi_2 t_2) ... \rangle$, where poi_i is the point of interest at index i. The mobility prediction task is thus formulated as: given a sequence $s(t)$ up to a timestamp n, predict the next POI at timestamp $n+1$.

Prediction Accuracy is estimated by averaging the accuracy across all the individuals present in that dataset. The individual prediction accuracy is computed by measuring the proportion of accurate predictions over all days of that individual (users who were not active on a day are excluded in the prediction). The accuracy of a model can thus be formalized by Equation 2.8.

$$\pi_{acc} = \frac{\sum_{t=1}^{T} \mathbb{1}_{poi_t = poi_t^*}}{T}, \tag{1.1}$$

where poi_t is the true next POI of an individual at time t, poi_t^* is the predicted next point of interest and T is the total number of prediction time-steps. The data is split into 10 windows consisting of 10% training set and the subsequent 10% as test set as performed in [226]. The training is performed in a cumulative manner such that the previous training instance is not lost (more details in section 1.5).

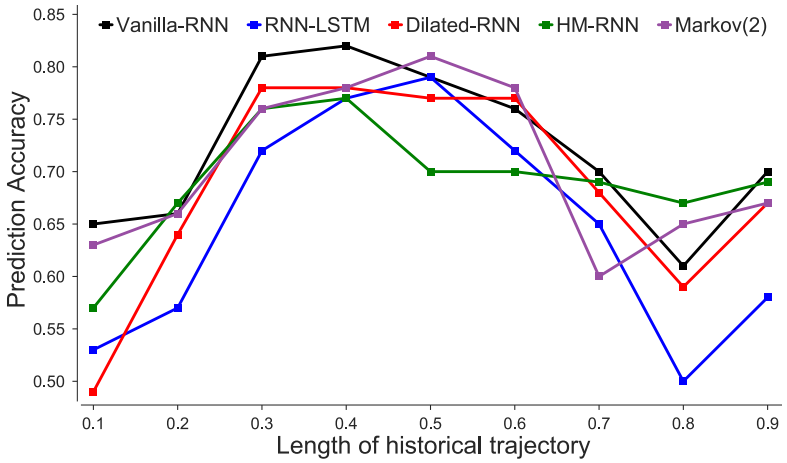


Figure 1.4 – Prediction accuracy of RNN variants variants on the MDC dataset. The horizontal axis signifies the proportion of trajectory length considered for the train-test split and vertical axis signifies the precision of the prediction model.

1.4.2 Data-agnostic Model Selection

In order to understand mobility dynamics at a high-granularity and to facilitate integration of these findings in mobility prediction models, it is paramount to evaluate distinct models of the same dataset. Additionally, dataset selection for performance quantification without analyzing the dataset characteristics can mask critical aspects of the model behaviors. In this section, we focus on the following two aspects: (1) generalizability of model performance formulated using certain dataset(s), and (2) effects of selecting a model while ignoring dataset attributes on performance vs. complexity trade-off.

In order to analyze the generalizability of the model performance, we apply the same prediction model on mobility datasets differing widely with respect to some key characteristics. We provide more details regarding these characteristics and quantify them in Section 1.5 (see Table 1.6). We also compare the accuracy of different prediction models on the same dataset using a fixed cross-validation methodology. Instead of presenting the average accuracy over the test-set, we adopt the canonical approach to perform sequential data cross-validation by rolling through the dataset (more details in Section 1.5). To facilitate easier result comprehension, we split the analysis first based on the usage of Markov models and neural networks.

In Figure 1.3, we compare the prediction accuracy of two approaches specified in Table 1.1: (1) mobility Markov chain (MMC), and (2) second-order Markov model (Markov(2)). We observe that the average accuracy and the variation trend of the Markov models differ by a large extent over the three datasets. Interestingly, the accuracy variation across the trajectory length is substantial for GeoLife and PrivaMov datasets as opposed to the MDC dataset. These accuracy variations stem from the fluctuation in dependencies between the POIs in a given dataset. In Figure 1.4, we present the accuracy comparison of different RNN variants on the MDC dataset. Although Vanilla-RNN and Markov-model (order-2) have lower computational complexity and

Extension	Architecture	Features
Vanilla-RNN [94]	• no gating mechanism • recurrent connections	• faster,stable training • simple architecture
RNN-LSTM [108]	• Vanilla-RNN connections • cell gating mechanism	• active self-connecting loops • prevents memory degradation
Dilated-RNN [37]	• LSTM cell structure • dilated skip connections	• parallelized computation • long-term memorization
HM RNN [50]	• variable dimensionality • long credit assignments	• hierarchical-temporal representation • ovel update mechanism

Table 1.4 – Recurrent Neural Network variants with their respective architectural differences and features.

lack long term memorization capability, they provide higher average accuracies as compared to RNN-variants. The lower accuracies of RNN-variants stem from overfitting on the training set which results in dropping accuracy on the validation set. We performed similar experiments on the PrivaMov and GeoLife datasets and observed that RNN-LSTM and HM-RNN provide the highest average accuracy on each respectively. From these experiments, we emphasize that the performance of the same prediction approach can differ widely across datasets. Moreover, a prediction model that performs poorly on one dataset can provide sufficiently favorable results on another. As a result, it is clear that conclusions regarding algorithmic performance cannot be justified without defining the dataset characteristics.

Markov models differ in their capacity to incorporate n-previous locations and their temporal representation. The RNN extensions on the other hand differ in their capacity to manipulate the internal memory and propagate gradients along the network. This is due to the difference between the gating mechanisms employed, regularization techniques and the connections within the individual neurons and the hidden layers. For instance, Dilated-RNNs [37] account for short and long-range correlations present in a sequence depending on the configuration of the skip-connections in the network. On the other hand, Hierarchical-Multiscale RNNs (HM-RNNs) [50] captures the latent hierarchical structure in the sequence by encoding the temporal dependencies with different timescales and thus being effective in representing the dependencies lying at different levels. This choice of the RNN variants was based on the ability of each to address distinct performance issues of the vanilla-RNN while modeling complex dependencies present in the dataset.

We present a summary of the major architectural differences and the features associated with each extension selected in this work to carry of experiments in Table 2.2. We use the standard implementations of the predictive algorithms as described in their respective papers, i.e., the same architecture (#neurons, #hidden layers, vertical depth) and the same hyperparameters (learning rate, unroll steps, activation function, optimizer and dropout rate). Mobility Markov chains [76] and Markov models order(2) [219] are implemented using the standard libraries.[2] Vanilla-RNN [94], RNN-LSTM [108] and dilated-RNN [37] are based on predicting the next character (language modeling) in the text, whereas HM-RNN [50] models the prediction task as

[2]hmmlearn: https://hmmlearn.readthedocs.io

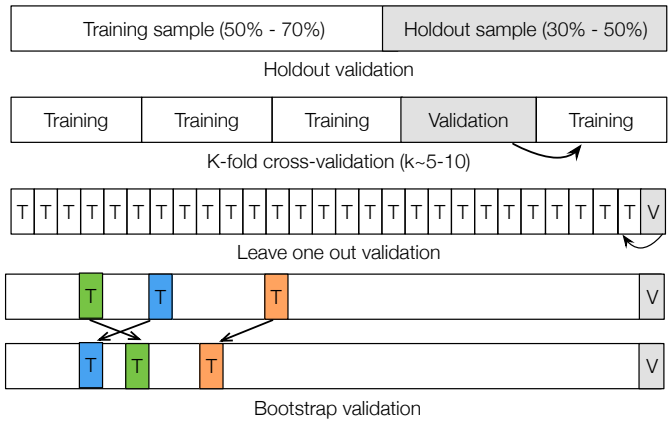


Figure 1.5 – Different methodologies of cross-validation. The figure shows the proportion and position of train-set (white) and the validation set (grey).

multivariate classification. For dilated-RNN [37] we use the dilations of 1, 2, 4, 8, 16, 32 and 64 and provided the results for dilation 32 after which we observe a drop in the accuracy.

We repeat the experiment shown in Figure 1.4 with the PrivaMov and GeoLife datasets. We observe that on the PrivaMov dataset, RNN-LSTM and dilated-RNNs provide a comparable accuracy and outperform the other models. However, HM-RNN provides the best accuracy on the GeoLife dataset. This leads us to question the generalizability of the performance results specified in the current studies. We thus argue that data agnostic model selection often results in an ineffective trade-off which necessitates a mobility data attribute-driven modeling approach.

1.4.3 Flawed Validation Methodology

In this Section, we focus on the second shortcoming that relates to the usage of inaccurate validation strategies by the existing studies and we present the misleading effects of their adoption. Table 1.1, 1.2, 1.3 show distinct validation strategies used to validate the proposed prediction models. These strategies can be categorized as follows: (1) holdout cross-validation, (2) leave one out cross-validation, (3) k-fold cross-validation, and (4) bootstrapping. Figure 1.5 presents the overview of these approaches. Holdout is the most common cross-validation techniques used where the data is typically split 70%-30% for training and testing; the model accuracy is reported on the test-set. In the case of k-fold cross-validation, the dataset is first split into k equal partitions and the training is performed on all but one, which is the validation set. This set is replaced k times with other partitions of the dataset to perform validation, the average accuracy on the k sets is then reported. The leave one out cross-validation technique is similar to k-fold, where a single data-point is treated as a sub-sample. The bootstrapping technique performs the train-test split after resampling the data points with replacement from the original dataset for every validation iteration.

Although the above validation approaches are suitable for accessing model performance on the

Dataset	Holdout cross-validation			K-fold cross-validation		
	80-20	70-30	60-40	3-fold	5-fold	10-fold
MDC	0.78	0.81	0.66	0.63	0.72	0.65
PrivaMov	0.63	0.65	0.45	0.68	0.57	0.52
GeoLife	0.83	0.65	0.63	0.75	0.70	0.61

Table 1.5 – Prediction accuracies derived by using different splits for holdout validation and different values of k for k-fold cross-validation.

dataset devoid of any temporal component, they cannot be applied for mobility trajectories as it is a sequential form of data. In order to validate prediction models based on sequential data, no future observations can be used to train the model. Furthermore, the data cannot be randomly split into train-test groups due to the temporal dimension of the geolocation observations. Instead, the data must be split with respect to the temporal order in which the values were observed. These approaches are highly susceptible to selection bias if the distribution of the data in the train-test set is not identical. We further discuss the implications of these validation techniques and propose adoption of a standardized technique in Section 1.5.

In order to highlight the misleading nature of the above validation approaches in the context of application to mobility modeling, we apply the holdout and k-fold cross-validation to access the performance of a Vanilla-RNN model (see Table 1.5). We select Vanilla-RNN due to discernibility compared to other RNN variants. We observe that different train-test split ratios result in different prediction accuracies in case of holdout validation for all the three datasets under consideration. A similar behavior is observed in case of k-fold cross-validation for distinct values of k. Thus, it is clear that the accuracy results computed by these validation measures are neither a conclusive evidence of model performance, nor do they provide a comparative measure to analyze performance with respect to another model. Based on the above, we argue that application of flawed validation methodology is detrimental to the advancement of human-mobility modeling.

1.5 Mobility-Modeling Framework

In this section, we address the shortcomings in the reviewed studies described in Section 1.4. We first present the data-driven mobility modeling framework and introduce mobility meta-attributes selected through experimentation based on rigorous statistical tests. We also present the block-rolling validation methodology to correctly assess the performance of the prediction model.

Figure 1.6 presents our approach for data-driven selection of the modeling technique. We identify the following four meta-attributes to quantity the characteristics of a mobility dataset: (1) Average trajectory length, (2) #POIs, (3) # POI interactions, and (4) POI interaction distance. Figure 1.6 also shows the experimentally determined attribute thresholds to select a particular approach.

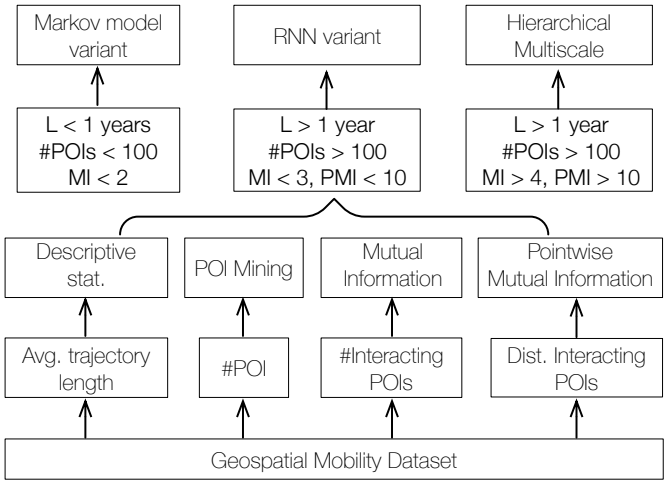


Figure 1.6 – Data-drive mobility model selection framework.

1.5.1 Meta-Attribute Selection

Mobility meta-attributes aid in characterizing a dataset to predict performance of forecasting algorithms and subsequently in selecting an appropriate modeling algorithm. We use statistical and information theoretic measures to estimate these meta-attributes and present the general descriptive statistics of the three considered mobility datasets in Table 1.6. The temporal mobility entropy is estimated using the approach specified by Song et al. in their seminal work regarding quantification of predictability limits [226] based on Lempel-Ziv data compression [275]. We refer the readers to [134] for further reading regarding computation of mobility entropy and subsequently determining the predictability.

Dataset	#users	#months	traj. length	POIs	Granularity	entropy	predictability
PrivaMov	100	15	1560000	2651	246 meters 24 seconds	7.63	0.7646
MDC	191	24	685510	2087	1874 meters 1304 seconds	6.08	0.8323
GeoLife	182	36	8227800	3892	7.5 meters 5 seconds	9.77	0.6219

Table 1.6 – Descriptive statistics with entropy and predictability.

In order to analyze the correlations between the characteristics presented in Table 1.6, we first segment the user trajectories from all the considered datasets into substrings of lengths ranging from 1 month to 60 months. The partitioned and aggregated datasets are then used to compute the descriptive statistics presented in Table 1.6. The above step is performed in order to generate a large amount of mobility data from to achieve higher confidence in the statistical tests. We first analyze variables in Table 1.6 to find significant correlations with predictability (see the correlation matrix in Figure 1.7). We find a significant correlation between the duration (length) of the user trajectory and mobility predictability. We find that trajectories collected for longer durations incorporate varying mobility behaviors, quantified in terms of periodicity variation with the POIs. However, this effect is moderated by the number of POIs which determine the

predictability. Therefore, trajectories spanning longer duration along with growing number of POIs shows a significant positive correlation with the mobility entropy and hence predictability. The t-test avail a p-value of 0.0002 for predictability and trajectory length and 0.000045 for predictability and number of POIs. In general, if we consider two mobility datasets D_1 and D_2, collected for time durations t_1 and t_2, where $t_1 > t_2$, the number of POIs in $D_1 > D_2$ This would result in the the predictability of $D_1 < D_2$ due to higher entropy of D_1 because of the varying periodicities in larger number of POIs.

In order to understand the influence of the POIs and the trajectory length in more detail, we estimate the dataset structure to analyze the relationship between groups of POIs (n-grams) and their periodicities. To derive the structure, we arrange the dataset in terms of stacked sequences belonging to each user, where each sequence contains time ordered POIs, henceforth termed as *symbols*. We base this approach on Yan et al. [253] work on aggregating individual mobility patterns to analyze aggregate scaling properties. Due to space restrictions, we only present the dataset structure of the PrivaMov dataset in Figure 1.8. We upload the high resolution dataset structure images on our github page of this project [3]. The dataset structure is with regards to symbol repetition statistics adapted from the file viewer utility contributed by Matt Mahoney [4]. We observe the distinct repetitive structure in each dataset at different levels and for different symbol-lengths. In the case of MDC dataset, we see symbol repetitions of length 1 occurring at a distance ranging between 1 to 10, and a few symbol matches of length 4 separated by at least 10^1 symbols. In the case of PrivaMov and GeoLife dataset however, the blue bands at the top shows that symbol matches of length 8 often separated by a distance of 10^5. The green band shows matches of length 4 commonly separated by 10^3 to 10^4 symbols, whereas the red bands show that the matches of length 2 are separated by about 10 to 500 symbols. The grey band shows that single symbol matches are usually separated by a distance of 1, 3 or 10.

Based on the dataset structure analysis, we see that there are longer dependencies spanning larger symbol lengths in case of GeoLife dataset as compared to PrivaMov dataset, whereas the MDC dataset mostly contain short-term dependencies. We therefore propose to leverage long-distance dependencies present in a dataset as another means to quantify the dataset characteristics. Therefore in addition to the average trajectory length and the number of unique POIs, we include LDDs as a meta-attribute.

1.5.2 Long-distance Dependencies

A long-distance dependency describes a contingency or interaction between two or more elements in a sequence that are separated by an arbitrary number of positions. More formally, LDDs are related to the rate of decay of statistical dependence of two symbols with increasing time interval or spatial distance between them. LDDs are commonly observed in natural languages, for instance in English, there is a requirement for the subjects and verbs to agree, i.e., words bear

[3] Github Page: https://bit.ly/2HRZGk5

[4] File Viewer (fv): http://mattmahoney.net/dc/textdata.html

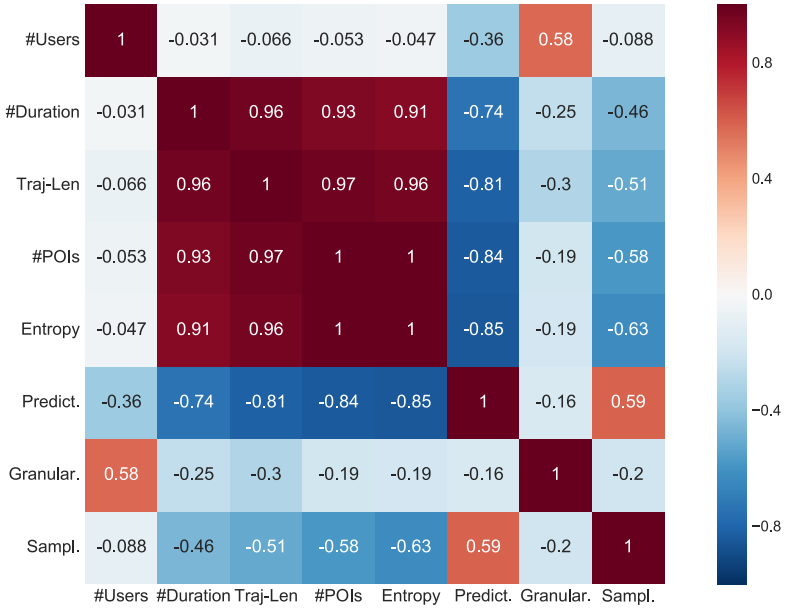


Figure 1.7 – Correlation matrix for all the descriptive statics variables, entropy and predictability

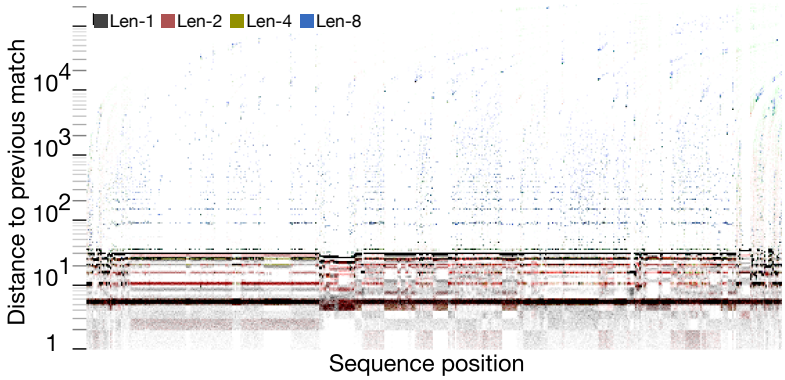


Figure 1.8 – The figure shows the distribution of symbol (POI) matches of length (1) black, 2 (red), 4 (green), and 8 (blue) in the PrivaMov dataset. The horizontal axis represents the position of the symbol sequence in the dataset, whereas the vertical axis shows the distance backwards to the previous match on a logarithmic scale.

relations and interact with other words in every sentence. Such a relation valuate one item with respect to the other within a certain search space. We extend this concept to human mobility where the POIs can be viewed as symbols in a natural language. Thus, mobility trajectory might display different degree of LDD depending on an individuals behavior, thus making them challenging to model computationally.

Mutual Information. Computing the mutual information of the data under consideration can be seen as a statical framework for discerning and quantifying the presence of LDDs. Mutual information I is a quantity that measures the relationship between two symbols and quantifies the measure of information communicated, on average by one symbol about another. I as a function

of distance between the individual events indicates the distribution of large but rare events and identify the presence of memory in the sequence. Mutual information between symbols X, Y is given by Equation 2.3.

$$\begin{aligned} I(X;Y) &= \sum_{X,Y} p(X,Y) \log \frac{p(X,Y)}{p(X).p(Y)} \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y), \end{aligned} \tag{1.2}$$

where $p(X,Y)$ is the joint distribution of two random variables X and Y, $p(X)$ and $p(Y)$ are the marginal distributions of X and Y. $H(X,Y)$ is the joint entropy of two random variables, X, Y distributed according to the pmf $p(X,Y)$ and $H(X|Y)$ is the conditional entropy of X given Y. Mutual Information can thus be used to quantify the interactions between the POIs in the dataset.

Pointwise Mutual Information. The strength of the interaction between individual symbols can then be estimated using a related concept known as pointwise mutual information (PMI). Unlike I, which quantifies the average information communicated by one symbol in the sequence about another, PMI quantifies the actual probability of co-occurrence of events $p(X,Y)$ differing from the expectation. It is computed on the basis of the probabilities of the individual events under the assumption of independence $p(X)p(Y)$ according to Equation 2.4.

$$PMI(X,Y) = log_2 \frac{N.C(X,Y)}{C(X).C(Y)} \tag{1.3}$$

$PMI(X,Y) = 0$ indicates that X and Y are statistically independent. Here, $C(X)$ and $C(Y)$ is the total number of occurrences of X and Y respectively and $C(X,Y)$ is the co-occurrence of (X,Y).

Figure 1.9 shows the LDD characteristics of all the three datasets considered in this work. All the measured curves for the three datasets are seen to decay roughly as power laws, and the value of exponent α indicates the extend of LDDs (power-law with cut-off for MDC dataset). Recall from Figure 1.8, where we noted a trend in LDDs, which is corroborated by the mutual information analysis in Figure 1.9. We also observe the effect of LDDs on the prediction accuracy results presented in Figure 1.3 and Figure 1.4. The MDC dataset provides higher accuracy as compared to the other two datasets and has a lower variation within the accuracies of different algorithms. This stems from the presence of short-distance dependencies in the individual trajectories present in the dataset (see Figure 1.9). Analyzing the mutual information trend also sheds light on the reasons pertaining to lower accuracies provided by RNN architectures, compared to Markov models at certain positions of trajectory-lengths. One reason could be the tendency of RNN models to actively seek for long-range dependencies while overlooking the short-term dependencies. We validate this behavior in the case of dilated-RNN's, where an increase in dilations (to account for

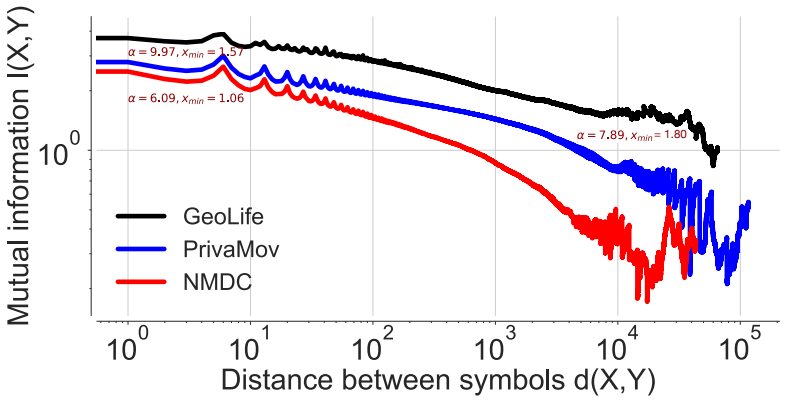


Figure 1.9 – Mutual information decay in the three datasets. The vertical axis represents the bits per symbol as a function of separation $d(X,Y) = |i - j|$, where the symbols X and Y are located at positions i and j in the considered sequence.

longer dependencies) results in lowering prediction accuracies.

1.5.3 Validation Methodology

In order to assess to performance of mobility modeling techniques, we adopt the block-rolling validation strategy used to validate time-series prediction models [23]. Sequential data such as mobility trajectories is subjected to autocorrelation [23], where the assumption made by the currently used validation approaches of i.i.d observations does not hold. Therefore, techniques such as holdout and k-fold cross-validation cannot be applied. For instance, 3-fold cross-validation applied over 3 time periods ignores the sequential nature of time, mixing up the past, present and future trajectory data points. Application of leave-one out or bootstrap is also not valid in this scenario as filtering out a data point does not remove the associated information due to the correlations with other observations.

In the block-rolling validation strategy, the dataset is split into *k*equal size blocks. The train set always consist of p contiguous blocks and the validation is performed on the block $p+1$ as shown in Figure 1.10. In the case of rolling cross-validation strategy, the blocks might be partitioned to include POI pairs that change minimally in their visitation periodicity, but frequently in time. The same partitions can include POI pairs that do not change in their periodicity over a long time periods, making this type of splitting misleading. The block-rolling technique incorporates these changes in the long-running variable and hence provides an unbiased validation after averaging over all the test sets.

1.6 Evaluation

In this section, we experimentally assess the trade-offs involved in model selection, following out data-driven modeling framework. We first compute the prediction accuracies using the three

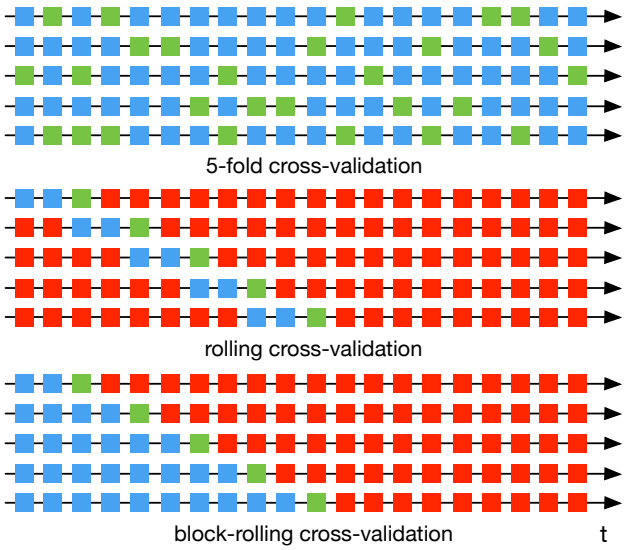
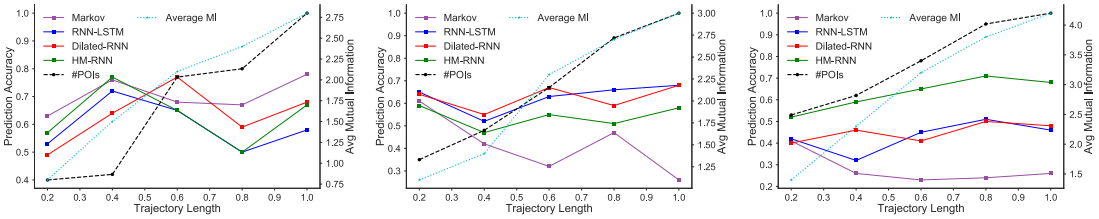


Figure 1.10 – Comparison of the 5-fold cross validation, rolling cross-validation, and block-rolling cross-validation techniques. The blocks in blue indicate data-points seen by the training model, the green blocks indicate the validation data-points and the red blocks are not seen by the training model.

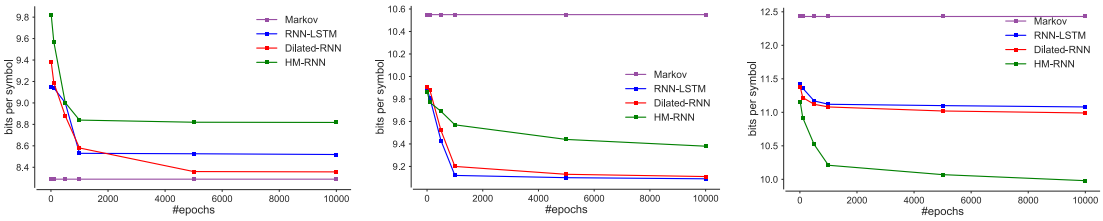
classes of modeling techniques: (1) Markov model, (2) RNNs, and (3) HM-RNN. We compare the block-rolling accuracies at each stage (every segment of trajectory length splits), with the number of POIs and the LDDs (mutual information) in Figure 1.11. For each dataset, we run the model 5 times and report the highest accuracy of the RNN and the HM-RNN models. We can see that Markov model (order-2) performs reasonably well on MDC dataset containing very short dependencies and lower number of POIs as compared to the other datasets. However, when the LDDs/dependency depth exceeds 2, we find that the performance of Markov model drops very quickly and comparable with random guess's performance ($\approx$ 10% variance). On the other hand, the performance of RNN-LSTM and Dilated-RNNs drops after mutual information exceeds 3, but is substantially better than random guessing. When the mutual information is between 2 and 3, LSTMs and Dilated RNN perform similarly, however Dilated-RNNs demonstrate an unstable behavior during training. Thus, we can clearly see that the Markov model outranks neural network models when the LDD depth does not exceed 2, and RNN models performs the best when the depth in the dependencies does not exceed 3. The higher accuracy of Markov models and RNN-LSTM when the dependency depth (mutual information), exceed 2 and 3 during certain trajectory lengths could be explained by fewer number of POIs. We argue that the dataset containing fewer POIs and longer dependencies could be modeled by Markov processes (POI <100, MI < 2) or RNN-LSTM (POI < 100, MI >2) as they still would fit in the representational capacity of the respective models. However for the datasets exceeding a collection duration of 2 years where the number of POIs/user > 100, and the dependency depth exceeds 3, HM-RNN models are necessary to model the intricate relationships.

In order to quantify the model complexity, we first analyze the computational efficiency to



(a) MDC Dataset. (b) PrivaMov Dataset. (c) GeoLife dataset.

Figure 1.11 – Experimental validation of the proposed framework by analyzing the prediction accuracy and its relationship with #POIs and dependency depth (mutual information).



(a) MDC Dataset. (b) PrivaMov Dataset. (c) GeoLife dataset.

Figure 1.12 – Compression ratio computed by each prediction model in terms of bits/symbol (POI).

reach a stable loss according to Equation 2.8. We find that Markov models require a fraction of time need by the NN-based approaches and typically lies in the range of 1-5 seconds, followed by RNN-LSTM ($\approx 9.5k$ seconds), Dilated-RNNs ($\approx 11k$ seconds) and HM-RNNS ($\approx 14k$ seconds). Furthermore, we also quantify the model's memorization and representational capacity by estimating the compression ratio provided over each of the datasets in terms of bits required to represent each POI (see Figure 1.12). We observe that for each dataset, the best compression ratio is obtained by the model suggested by the meta-attributes. This experiment also highlights the computational complexity in terms of number of epochs to arrive at the global minima.

1.7 Conclusion

In this paper, we have highlighted the inconsistencies and pitfalls in human mobility modeling and prediction research through a large scale systematic review. Through this review, we have attempted to systematize knowledge and provide guidelines towards performing credible mobility modeling research. We have exposed the consequences of relying on data-agnostic model selection and adopting inaccurate validation methodologies through experiments on three real-world mobility datasets. In order to address these problems, we have proposed four meta-attributes, that can accurately characterize a mobility dataset for selecting an appropriate modeling technique. Through a range of experiments, we have shown the applicability of our data-driven approach of model selection and analyzed the accuracy vs. complexity trade-offs associated with each. We offer the literature meta-data and the tools to the community with the hope to improve

and advance the reliability of human-mobility modeling research. In the future work, we will analyze the tradeoff between model size (depth, height, #neurons) and memorization capacity specific to human-mobility behaviors. Furthermore, we also propose adoption of early-stopping in the future work to halts the model training at the *best performance* epoch, where the training loss is decreasing but the validation loss starts increasing.

2 Examining the Limits of Predictability of Human Mobility

Abstract

We challenge the upper bound of human-mobility predictability that is widely used to corroborate the accuracy of mobility prediction models. We observe that extensions of recurrent-neural network architectures achieve significantly higher prediction accuracy, surpassing this upper bound. Given this discrepancy, the central objective of our work is to show that the methodology behind the estimation of the predictability upper bound is erroneous and identify the reasons behind this discrepancy. In order to explain this anomaly, we shed light on several underlying assumptions that have contributed to this bias. In particular, we highlight the consequences of the assumed Markovian nature of human-mobility on deriving this upper bound on maximum mobility predictability. By using several statistical tests on three real-world mobility datasets, we show that human mobility exhibits scale-invariant long-distance dependencies, contrasting with the initial Markovian assumption. We show that this assumption of exponential decay of information in mobility trajectories, coupled with the inadequate usage of encoding techniques results in entropy inflation, consequently lowering the upper bound on predictability. We highlight that the current upper bound computation methodology based on Fano's inequality tends to overlook the presence of long-range structural correlations inherent to mobility behaviors and we demonstrate its significance using an alternate encoding scheme. We further show the manifestation of not accounting for these dependencies by probing the mutual information decay in mobility trajectories. We expose the systematic bias that culminates into an inaccurate upper bound and further explain as to why the recurrent-neural architectures, designed to handle long-range structural correlations, surpass this upper limit on human mobility predictability.

Keywords: Mobility predictability limits; Entropy convergence; Mutual information; Mobility modeling.

2.1 Introduction

The proliferation of mobile devices equipped with internet connectivity and positioning systems has resulted in the collection of large amounts of human-mobility data. Real-time user locations are typically collected using the global positioning system (GPS), call detail record logs (CDR) and wireless-LAN (WLAN). The resulting location datasets can be then used to study and model user mobility behaviors, beneficial to a variety of applications, such as traffic management, urban planning and location-based advertisements. One of the applications of mobility modeling consists of formulating predictive models to forecast individual trajectories. For this, various methods have been proposed, including Markov chains [148], neural networks [136] and finite automata [192]. Existing research has used several datasets differing with respect to their spatial and temporal granularity, resulting in vastly contrasting prediction accuracies ranging from over 90% to under 40% [54].

2.1.1 Benchmarking Limits of Mobility Prediction

In this context, the seminal paper of Song et al. [226] laid the foundations for computing a theoretical upper bound on the maximum predictability of human mobility. This work establishes a benchmark for quantifying the performance of different prediction models and generalizes its approach across various datasets. The goal of mobility prediction is to predict the next visited user location with the highest possible accuracy, quantified in terms of the proportion of accurate predictions, noted as π_{acc}. Song et al. [226] define predictability upper bound, noted π^{max}, as the highest potential accuracy of modeling the mobility behavior of individuals present in a given dataset (highest possible π_{acc}). The value of π^{max} is defined by the entropic level of this dataset, and lower entropy would imply higher predictability. The derived π^{max} is experimentally corroborated by constructing a prediction model and computing π_{acc}, accuracy of forecasting user locations on the same dataset. Given that π^{max} is the upper bound of prediction accuracy as defined by Song et al. [226], $\pi_{acc} \leqslant \pi^{max}$ should always hold.

We highlight that π^{max} should not be confused with *predictability horizon* [14], which is defined as the limit of how far ahead one can predict (utmost prediction range), given a mobility dataset. The question therefore is not how long is the horizon of the predictability limit, but given a horizon (the next time instance in this case) what is the maximum possible predictability. The prediction model will contain some amount of uncertainty within this horizon which is limited by the chaotic nature of the individuals' mobility behavior present in the dataset. Furthermore, the computation of π^{max} is dependent exclusively on the mobility patterns of individuals and does not account for any supplementary information. To this end, Qin et al. [202] estimate the maximum predictability given a single location instance and quantify how predictable individuals are in their mobility.

In practice, Song et al. [226] compute π^{max} by first estimating the entropy of the mobility trajectories contained in the dataset based on Lempel–Ziv data compression [275]. This entropy

estimate is used to solve the limiting case for Fano's inequality [200]. Fano's inequality [200] is an information-theoretical result used to compute lower bound on the minimum error probability in multiple-hypotheses testing problems. The estimated lower bound is then used to compute the maximum possible accuracy of predictability (π^{max}). The proposed theoretical upper bound by [226] ($\pi^{max} = 93\%$) is computed using a call detail record (CDR) dataset consisting of 50,000 users collected by a telecommunications operator for a duration of three months. They also show that π^{max} is independent of radius of gyration and movement periodicity, hence they observe an insignificant level of variation across a heterogeneous population.

Several subsequent works computed π^{max} using datasets of different types, collected for varying durations and performed empirical validation by constructing Markov based prediction models [76]. Lu et al. [154] estimate π^{max} to be 88% for a call detail record (CDR) dataset. dataset consisting of 500,000 users, collected for a duration of five months. In order to validate this bound, they use *order-1* Markov chain based prediction model and achieve an average prediction accuracy (π_{acc}) of 91%. They also show that higher-order Markov chain models do not significantly improve the prediction accuracy. Their interpretation behind surpassing their own estimated theoretical bound is that trajectories exceeding this bound are non-stationary, whereas the accuracy of stationary trajectories prevails within the bound. A trajectory is considered to be stationary when people tend to remain still during short time-spans. This conclusion directly contradicts findings of Song et al. [226], because non-stationary trajectories should by definition have a higher entropy. Later, Cuttone et al. [54] show that the stationary nature of trajectories plays a significant role in the higher accuracies resulting from Markov models [54] as they often predict the user will remain in the previous location, i.e., self-transitions. Lin et al. [148] also show that π^{max} is independent of the spatial granularity (Δt) data sampling rate (Δt) which was later questioned by Smith et al. [222] ($\pi^{max} = 93\% - 74\%$ for varying values of Δs and Δt) and Cuttone et al. [54] ($\pi^{max} = 65\%$). Smith et al. [222] and Cuttone et al. [54] used mobility datasets [228, 266] containing GPS trajectories and showed that predictability has a direct correlation with the temporal resolution and an inverse correlation with the spatial resolution.

The CDR datasets used in the preliminary works [154, 226] are known to have inherent gaps due to the short bursts of calls masking the user's true entropy. Therefore, it should be noted that CDRs are a rough approximation of human mobility due to the low granularity of GSM cell IDs. Since human mobility varies under time translations, the entropy not only depends on the duration of past observations but also on number of visited locations; these factors tend to be hidden in such datasets [16, 25]. Additional inconsistencies become evident due to the fact that the authors in [154, 226] group the user locations into one-hour bins when constructing the historical trajectory of a user. Further inspection suggests that these models can foresee future locations at π^{max}, only when an individual is present in one of the top n bins [54]. The first two works [154, 226] thus consider the last location of each day, consequently predicting only the user's home place. Under such a scenario, Ikanovic et al. [111] and Cuttone et al. [54] showed that the predictability of the true next location is significantly lower ($\pi^{max} = 71.1 \pm 4.7\%$) than the predictability of the location in the subsequent bin. They further showed that an individual's

mobility entropy is directly proportional to the number of visited locations. The authors also point out that the generating function behind the stochastic mobility behavior is often unknown. Therefore the bounds cannot be estimated theoretically and require empirical derivation. Cuttone et al. [54] achieve an even lower bound on π^{max} of 65% on the same datasets with the same methods as Ikanovic et al. [111].

In this paper, we build upon the work of Zhao et al. [265], who demonstrate the non-Markovian character of the online and offline human behavior. They analyze datasets consisting of user web browsing and location-visit patterns and estimate the rank distribution of these visits. They show the presence of the scaling law associated with the dwelling times at the visited websites and locations. This study hints at the non-Markovian character and is based on a small scale CDR dataset using one-point statistics [180]. However, mobility trajectories involve complex dynamics, which are better characterized by two-point statistics [148], hence the work of Zhao et al. [265] is inconclusive. On the contrary, we study the mobility characteristics through the lens of both these methodologies on three large scale datasets collected at varying levels of spatiotemporal granularities. This minimizes any bias and substantiates our findings.

Problem Definition. Before going into further details, we can state the central objective of this work as follows: knowing that we observe a discrepancy between the predictability upper bound and the empirical prediction accuracy, we aim at investigating the methodology behind the upper bound estimation and at understanding the primary reasons for this discrepancy. To this end, we adopt the approach consisting in the three steps listed hereafter, where each step acts as a causal verification for the next.
Approach:

1. Confirm the discrepancy between the upper limit of predictability and prediction accuracy through extensive experimentation using widely contrasting prediction models on contrasting datasets.

2. Following the discrepancy confirmation, revisit the assumptions underlying the upper bound computation methodology.

3. Scrutinize the assumptions, analyze the reasons contributing to the failure of the methodology.

2.1.2 Discrepancies and Inconsistencies

Table 2.1 summarizes the findings of previous works by indicating the π^{max} values, the prediction accuracy score (π_{acc}), prediction model used and the type/duration of the dataset used. As seen in the Table 2.1, [226] and [222] do not compute the π_{acc} scores, while [154] estimates $\pi_{acc} > \pi^{max}$, contradicting findings of [226]. We also observe that π^{max} is impacted by Δs and Δt as evident from the work of [222] and [111], contracting [226] and [154]. Therefore, we observe

an inconsistency regarding the maximum predictability bound π^{max} and its relation to π_{acc}. We also observe disagreements regarding the impact of entropy, on the number of uniquely visited locations and on the spatiotemporal resolution of the trajectory on π^{max}. Moreover, the π_{acc} derived by some works [111, 154] based on Markov chains surpass the limits of their own π^{max}. To systematically revisit the above discrepancies and inconsistencies, we compute the values of π^{max} for three large scale mobility datasets and estimate π_{acc} using seven different prediction techniques for benchmark and comparison purpose.

Table 2.1 – Comparison of π^{max} and π_{acc} at varying granularities of Δs (spatial granularity) and Δt (temporal granularity) reported by existing literature.

Authors (year)	π^{max} $(\Delta s, \Delta t)$	π_{acc}	**Prediction Model**	**Dataset Duration**	**Dataset Type**
Song et al. [226] (2010)	93% (3–4 km)	–	–	3 months	CDR
Lu et al. [154] (2013)	88% (3–4 km)	91%	Markov (first-order)	five months	CDR
Smith et al. [222] (2014)	93.05–94.7% (350 m, 5 min) 81.45–85.57% (100 m, 5 min) 74.23–78.20% (350 m, 60 min)	–	–	36 months	GPS
Ikonovic et al. [111] (2017)	95.5 $\pm$ 1.8% (1.7 km) 71.1% (25 m)	88.3 $\pm$ 3.8% 75.8%	Markov (first-order)	36 months (same as previous)	GPS

2.1.3 Questioning the Predictability Upper Bound

In this paper, we challenge the validity of the currently established mobility predictability upper bound following our own observation that recurrent-neural networks surpass this limit. Our central objective is a comprehensive inspection of the methodology behind the derivation of this upper limit and identify the probable causes behind this anomaly. This involves analyzing and confirming the two phenomenon described hereafter.

1. Substantiate the observed discrepancy between π_{acc} and π^{max}. To this end, we build prediction models using seven distinct approaches and conduct a comprehensive accuracy analysis based on three real-world mobility datasets.

2. Revisit the assumptions hereafter, which might have lead to this discrepancy.

   (a) Human mobility is Markovian and thus possesses a memoryless structure.

   (b) The mobility entropy estimating technique achieves an asymptotic convergence.

   (c) The predictability upper bound accounts for (all) the long-distance dependencies in a mobility trajectory.

2.1.4 Roadmap and Main Findings

We now sketch the relevant contributions and present the organization of this paper.

1. We discuss all the relevant concepts used in this work in Section 2.2 and illustrate how diverse concepts such as entropy, mutual information and predictive information interact with each other in the light of the predictability upper bound.

2. In Section 2.3, we describe the mobility datasets used in this work and confirm the discrepancy between the maximum upper bound of mobility prediction derived by the previous works and the empirical prediction accuracy derived using recurrent-neural network variants. In order to minimize any bias, we construct seven different prediction models and compute the accuracy across three datasets differing with respect to their collection timespans, region, demographics, sampling frequency and several other parameters.

3. In Section 2.4, we audit three underlying assumptions in the currently used methodology for π^{max} computation.

   (a) In Section 2.4.1, we demonstrate the non-Markovian character of human mobility dynamics contrary to the previously held assumption. Our statistical tests to confirm the nature of human mobility include (i) rank-order distribution, (ii) inter-event and dwell time distribution, and (iii) mutual information decay.

   (b) In Section 2.4.2, we analyze the entropy convergence by comparing entropies derived by using Lempel–Ziv 78 and Lempel–Ziv 77 encoding schemes on mobility trajectories. Based on this result, we show that there does not exist an ideal entropy estimation scheme for mobility trajectories that achieves an asymptotic convergence.

   (c) In Section 2.4.3, we assert that the current methodology used to estimate S^{real} does not represent an accurate entropy estimate of mobility trajectory. To this end, we demonstrate that the individual elements present in a mobility subsequence derived by the currently used encoding schemes have non-zero dependencies un-accounted for, when deriving the mobility entropy. We validate such a manifestation by computing the pointwise mutual information associated with mobility trajectories which indicate an on average positive pointwise mutual information (PMI).

4. In Section 2.5 we discuss the likely causes behind this discrepancy being overlooked. We also present the potential reasons as to why recurrent neural networks (RNN) extensions exceed the theoretical upper bound and discuss the applicability of the prediction models in different contexts. We conclude the paper in Section 2.6.

2.2 Relevant Concepts

In this section, we present the key concepts and principles relied upon in this work. We present a brief description of Markov processes which have been widely used in human mobility prediction, followed by the definition of long-distance dependency (LDD). We further relate the discussion regarding long-distance dependencies (LDDs) with their quantification through mutual information. We then present the relationship between entropy, encoding and data compression, followed by the conceptual understanding of predictability theory and its relation to the above concepts.

2.2.1 Mobility Modeling

A mobility modeling task aims at estimating the probability distribution over a user's location traces by minimizing the negative log-likelihood of the training sequences [50]:

$$min\theta - \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T^n}\log p(L_t^n|L_{<t}^n;\theta), \tag{2.1}$$

where θ is the model parameter, N is the number of training location traces, and T^n is the length of the n-th location trace. A location or a symbol at time t of trace n is denoted by L_t^n, and $L_{<t}^n$ denotes all the previous locations (symbols) at time t.

2.2.2 Markov Processes

Several of the previous works related to human mobility modeling [76], mobility prediction [15, 225] and derivation of the upper bound [154, 226] rely on the Markov assumption of human mobility. Markov processes are a natural stochastic extension of finite state automata, where the state transitions are probabilistic and there is no input to the system in contrast to a finite state automaton. Thus, the observation at a given time t_i only depends on events at previous time step t_{i-1} or on previous n time steps for an n-order Markov chain. Such stochastic processes are characterized in terms of the transition probabilities, where the probability for transitioning to the next state is an exponentially distributed random variable. Formally, a sequence of random variables $X_1, X_2, X_3, ...$ abide the Markov property as expressed in Equation (2.2).

$$P(X_n = x|X_{n-1} = x_{n-1}, ..., X1 = x_1) = P(X_n = x|X_{n-1} = x_{n-1}), \tag{2.2}$$

where $P(A|B)$ is the probability of A given B. The possible values of X_i form a countable set called the state space of the Markov chain.

2.2.3 Long-Distance Dependencies

A long distance dependency (LDD) describes a contingency or interaction between two or more elements in a sequence that are separated by an arbitrary number of positions. More formally, LDDs are related to the rate of decay of statistical dependence of two points with increasing time interval or spatial distance between them. Commonly observed in natural languages, for example in English, there is a requirement for the subjects and verbs to agree, i.e., words bear relations and interact with other words in every sentence. LDDs are thus a pervasive feature of language which involve different faces such as agreement, binding, control and displacement among others [47], that imply a relation between two or more items. Such a relation valuate's one item with respect to the other within a certain search space or domain, non-linearly but structurally defined [47]. Hauser et al. [104] show that natural languages go beyond purely local structure by including a capacity for recursive embeddings of phrases within phrases, which

can lead to statistical regularities that are separated by an arbitrary number of words or phrases. Such long-distance, hierarchical relationships are found in all natural languages for which, at a minimum, a phrase-structure grammar is necessary [104]. Similarly, a mobility trajectory might display different degree of long-distance dependency depending on an individuals behavior, thus LDDs are challenging to model computationally. Typically, the predictability is directly dependent on the model's ability to account for the LDDs present in a sequence.

2.2.4 Recurrent Neural Networks and Extensions

Several approaches have been used in an attempt to model the presence of LDDs in natural languages such as stochastic gradient decent [22]. RNN [209] have been proven efficient to model temporal data while accounting for the LDDs to a certain extent [61]. RNNs are a class of supervised machine learning models, comprised of artificial neurons with one or more feedback loops. The feedback loops are recurrent cycles over time or sequence which forms a hidden memory representation beneficial for processing and learning the dependencies between input sequences. A recurrent network is trained using a dataset consisting of a large number of input-target pairs with the objective to minimize the difference between the output and target pairs. This is performed by optimizing the weights of the network.

Modeling LDDs however is still challenging for simple/vanilla RNNs [163] due to the exploding and vanishing gradient (exponential decay of gradient, as it is back-propagated) problem [22]. Since RNN is a structure through time, the typical gradient decent is extended through time to train the network, called back-propagation through time (BPTT) [231]. However, computing error-derivatives through time is challenging, due to the unstable dynamics of RNN which renders gradient decent ineffective [207]. Thus, extensions to the vanilla-RNN were designed such as RNN-long short term memory (LSTM) [108] that enforce a constant error flow through the network thereby bridging the lags in the individual steps and thus addressing the above problem to some extent.

These extensions differ in their capacity to manipulate the internal memory and propagate gradients along the network [207]. More specifically, they differ with respect to the gating mechanisms employed [207], regularization techniques and the connections within the individual neurons and the hidden layers. We present a summary of the major architectural differences and the respective features associated with each extension selected in this work to carry of experiments in Table 2.2. We select the above models based on the contrasting nature of their connection architecture and the cell structure to quantify the contribution of such parameters on the modeling result. For instance, recurrent highway networks (RHNs) [273] are built to account for short and long-range correlations present in a sequence. On the other hand, pointer sentinel mixture models (PSMMs) [161] weigh long-range dependencies much higher than short-distance correlations in the sequence. In this paper, we quantify the performance of these variants to capture LDDs present in three datasets and analyze their applicability in various contexts.

Table 2.2 – Recurrent neural network variants with their respective architectural differences and features.

Extension	Architecture	Features
Vanilla-RNN [94]	• no cell state/gating mechanism • recurrent connections	• faster and stable training • simple architecture
RNN-LSTM [108]	• similar connections as Vanilla-RNN • diff. cell state with gating mechanism	• actively maintain self-connecting loops • prevents memory degradation
Dilated-RNN [37]	• similar cell structure as LSTM • dilated skip connections	• increased parallelism in the computation • improves long-term memorization capabilities
RHN [273]	• diff. cell design • long credit assignment paths	• handles short-term patterns • reduces data-dependent parameters for LDD memorization
PSMM [161]	• diff. gating function, shortcut connections • variable dimensionality hidden state	• improves handling of rare symbols • allows for better long-distance gradients

2.2.5 Mutual Information

LDDs are challenging to detect and characterize due to a large number of associated parameters. Computing the mutual information of the data under consideration can be seen as a statical framework for discerning and quantifying the presence of LDDs and thus nature of the data generation process. Mutual information I is a quantity that measures the relationship between two random variables that are simultaneously sampled and quantifies the measure of information communicated, on average by one random variable about the other. I as a function of distance between the individual events can indicate the distribution of large but rare events and identify the presence of memory in the sequence. Mutual information, between two discrete random variables X,Y jointly distributed according to probability mass function $p(x,y)$ is given by Equation (2.3).

$$\begin{aligned} I(X;Y) &= \sum_{X,Y} p(X,Y)\log\frac{p(X,Y)}{p(X).p(Y)} \\ &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= H(X) + H(Y) - H(X,Y), \end{aligned} \tag{2.3}$$

where $p(X,Y)$ is the joint distribution of two random variables X and Y, $p(X)$ and $p(Y)$ are the marginal distributions of X and Y. $H(X,Y)$ is the joint entropy of two random variables, X,Y jointly distributed according to the pmf $p(X,Y)$ and $H(X|Y)$ is the conditional entropy of X given Y.

A related concept also used in our work is PMI. Unlike I, which quantifies the average information communicated by one symbol in the sequence about another, PMI quantifies the actual probability of co-occurrence of events $p(X,Y)$ differing from the expectation. In the context of mobility, a symbol refers to a point of interest. Thus, a sequence representing an individual's trajectory is composed of temporally ordered points of interest It is computed on the basis of the probabilities of the individual events under the assumption of independence $p(X)p(Y)$ according

to the Equation (2.4).

$$PMI(X,Y) = log_2 \frac{N.C(X,Y)}{C(X).C(Y)}. \tag{2.4}$$

$PMI(X,Y) = 0$ indicates that X and Y are statistically independent. Here, $C(X)$ and $C(Y)$ is the total number of occurrences of X and Y respectively and $C(X,Y)$ is the co-occurance of (X,Y). PMI is defined only over particular values of X and Y, and can therefore be negative, zero, or positive; it only considers the independence of those two particular values. A positive value of PMI indicates that the two events co-occur more frequently than would be expected under an independence assumption and a negative PMI means they cooccur less frequently than would be expected. Unlike PMI, $I(X,Y)$ always takes non-negative values. $I(X,Y) = m$ can be interpreted as the reduction in uncertainty about the event Y by m bits knowing the value of X.

Computing mutual information of a given dataset can quantify the presence of LDDs, leading to a suitable selection of predictive model a-priori. Pointwise Mutual information on the other had provides a fine-grained understanding of the dependencies within two symbols in a sequence. In this paper, we will rely on these measures to quantify LDDs present in the three mobility mobility datasets.

2.2.6 Entropy, Encoding and Compression

The seminal work of Shannon [213] defines entropy as the absolute minimum amount of storage and transmission required for capturing any information as opposed to raw data. Thus the entropy $H(X)$ is equal to the amount of information learnt on an average from one instance of the random variable X. It is important to highlight that, the entropy does not depend on the value that the random variable takes, but only on the probability distribution $p(x)$. The probabilities of different values can be leveraged to reduce the number of bits needed to represent the data if and only if the variable has non-uniform distribution. Thus, entropy can also be defined as the measure of compressibility of the data, or a measure that defines the predictability of a single random variable. Lower entropy therefore generally signifies higher predictability.

$$H(X) = -\sum_i (p(i) \times log_2(p(i))) = -E[\log(p(x))]. \tag{2.5}$$

Entropy rate on the other hand extends the concept of entropy from random variables to stochastic processes [243]. It is defined as the lower bound on the per-symbol description length when a process is losslessly encoded. In order to estimate the entropy rate of a stationary ergodic process, Kontoyiannis et al. [126] discuss a family of estimators and prove their point-wise and mean consistency. This approach runs a universal coding algorithm on the segment of the source output and averages the longest match-lengths. The resulting compression ratio can then be used as an upper bound for the entropy. If the segment length is long enough for the compression algorithm to converge, the compression ratio will be a good estimate for the source entropy. However it is

important to note that there is no universal rate of compression [215].

Wyner et al. [250] performed an asymptotic analysis of the Lempel–Ziv algorithm [275] and found a relationship between the entropy rate and the asymptotic behavior of longest match-lengths. Building upon this relationship, Grassberger [90] suggested an entropy estimator based on average match-lengths for measuring the information of signals containing strong long-range correlations. Thus for any sequence S_N of N binary digits, well-defined probabilities $P_N\{S_N\}$ exist for finding S_N starting at any chosen site within S. Shannon's entropy can also be written as $h = \lim_{N\to\infty} h_N$ where h_N is defined as below:

$$h_N = -\frac{1}{N}\sum_{S} p_N\{S_N\} log_2 \times p_N\{S_N\}. \tag{2.6}$$

The quantities h_N are called block entropies [91] where $h \leqslant h_N$ as the limit converges from the Shannon entropy equation. Grassberger [90] shows that these bounds are tight if the sequence has no long-range correlations. More precisely, $h_n = h$ for all $n \geqslant N$ if the sequence can be modeled by an N^{th} order Markov chain. In this paper, we show that the currently used encoding schemes ignore the presence of dependencies in individual subsequences present in a mobility trajectory and thus result in an inflated entropy estimation and consequently in a deflated predictability bound.

2.2.7 Predictive Information

We now briefly discuss the predictive information theory, which provides a fine grained understanding of the interaction amongst the preceding concepts and they influence predictability. Predictive Information measures and quantifies how much of the past observations (can) tell us about the future [25]. The relationship between predictability, compressibility and temporally correlated entropy (time-series data) has been explored at length by Bialek et al. [25]. This is an important concept that ties the notions regarding event prediction, entropy and mutual information.

We restrict the discussion to sequential data, in which case predictive information diverges when the observed series allows to learn a more precise model of the data dynamics. Different variants of the predictions (next event, average event rate, event uncertainty etc.) are different slices through the conditional probability distribution. Greater concentration of this conditional distribution implies smaller entropy as compared to the prior distribution. The reduction in entropy can be viewed as the information that the past provides about the future [212]. Furthermore, in a time series if there is invariance under time translations, the entropy of the past data depends only on the duration of the observations [25]. The entropy of the sequence is thus in direct proportion to the observed duration and therefore the predictability is associated with the deviation of the entropy from extensivity. The average amount of information about the current state of a time-series is independent of how long the time-series has been observed. For models with a finite

number of parameters, the stochastic complexity is proportional to the number of parameters and logarithmically dependent on the number of data points [107, 212].

Finally we look at the result stated in Lin et al. [148]: mutual Information between two symbols, as a function of the number of symbols between the two, decays exponentially in any probabilistic regular grammar, but decays like a power law for a context-free grammar. This is an important observation relied upon in our work, given that human mobility has been known to follow context-free grammar [82]. Lin et al. [148] further state that exponential distribution is the only continuous distribution with the memory-less property. In order for a process to have a non-exponential probability distribution and satisfy the Markov property, the precise transition probability given the current state must be known. If a process has two or more states and transitions from each state with some non-exponential probability, then knowledge of the current state will not be sufficient to estimate the future distribution (next event prediction). It is important to note here that, for very short distances, power law decay and exponential decay are non-trivial to distinguish. Finally, Lin et al. [148] state that in a LDD driven system the number of bits of information provided by a symbol about another, drops as a power law with distance in sequences. This distance is defined as the number of symbols between the two symbols of interest. In this paper, we evaluate this observation made on natural languages on human mobility and use the results to verify the nature of human mobility.

2.3 Confirming π^{max} Discrepancy with Real-World Datasets

In this section, we present the datasets used for all the empirical analysis conducted in this work and we formalize the notion of human mobility prediction. We discuss the accuracy results estimated using seven prediction techniques and compare them with respect to the theoretical upper bound. We confirm the discrepancy between π_{acc} and π^{max} and show that $\pi_{acc} \leqslant \pi^{max}$ does not always hold.

2.3.1 Experimental Setup

We now present the experimental setup for all the analysis performed in this work, starting with the description of the datasets used. We emphasize that the value of π^{max} was dependent upon the experimental setup and dataset characteristics. Therefore it was essential to keep the same setup for computing π^{max} and the empirical prediction accuracy for a legitimate comparison.

Real world Mobility Datasets. We conducted all the experiments by using three mobility datasets whose specifications are shown in Table 2.3. The PrivaMov dataset [167] was collected through GPS, WiFi and GSM in the city of Lyon (France) and includes university students, staff and their family members. The Nokia mobile dataset [137] (NMDC) was collected in the Lake Geneva region of Switzerland and consists young individuals' trajectories, collected through GPS, WLAN, GSM and Bluetooth. The GeoLife dataset [266] was collected in Beijing (China) and

contains trajectories recorded through GPS loggers and GPS-phones. Table 2.3 also presents the values (theoretical) of S^{real} and π^{max} computed using the approach mentioned by Song et al. [226] and Lu et al. [154] as per Equation (2.7) which is based on Lempel-Ziv data compression [275].

$$S^{real} = (\frac{1}{n}\sum_{i=1}^{n}\lambda_i)^{-1} log_2(n), \tag{2.7}$$

where n is the length of the trajectory (total number of locations) and λ is defined as the length of the shortest substring at an index i not appearing previously from index 1 to $i-1$. Note that we use the same base (2) in entropy estimation as for the logarithm in Fano's inequality. Furthermore, the length of the substrings is set to zero upon reaching index i, when no more unique substrings can be computed using the above method. π^{max} is then estimated by solving the limiting case of Fano's inequality [80]. The computation of S^{real} and π^{max} at the aggregate level for the dataset was based on the independence of predictability on travel distance (radius of gyration r_g) in human mobility as demonstrated by previous studies [154, 226, 253].

Table 2.3 – Mobility dataset specifications and their respective S^{real} and π^{max} values.

Datasets	Num. Users	Duration (months)	Avg. Trajectory Length	Distinct Locations	Avg. Spatio-Temporal Granularity	S^{real}	π^{max}
PrivaMov	100	15	1,560,000	2651	246 meters 24 s	6.63	0.5049
NMDC	191	24	685,510	2087	1874 meters 1304 s	5.08	0.6522
GeoLife	182	36	8,227,800	3892	7.5 meters 5 s	7.77	0.4319

Mobility prediction. We relied on the widely used definition of mobility prediction [76], which describes it as forecasting the transitions between places, after eliminating all self-transitions [54, 222]. A preliminary step in achieving this consists of transforming the raw GPS locations into a sequences of points of interest [135]. A point of interest was defined as any location where an individual visits with an intentional purpose with a perceived priority for e.g., home/work place, gym, train station etc. Among the plethora of existing works dedicated to the problem of extracting these points, we rely on our approach that is independent of a priori assumptions regarding the data and individual mobility behaviors [135]. We then convert the raw GPS trajectory of a user u, $T_u = \langle (lat_1, lon_1, t_1), (lat_2, lon_2, t_2) ... (lat_n, lon_n, t_n) \rangle$, where lat_i, lon_i are the latitude and longitude coordinates respectively and t_i is the timestamp such that $t_{i+1} > t_i$ into a sequence of temporally ordered points of interest, $s(t) = \langle (poi_1, t_1), (poi_2 t_2) ... (poi_n, t_n) \rangle$, where poi_i is the point of interest at index i. The mobility prediction task was thus formulated as: given a sequence $s(t)$ up to a timestamp n, predict the next point of interest at timestamp $n+1$. The prediction accuracy was then estimated by following the approach stated by Lu et al. [154].

Predictive algorithms. We estimated the empirical predictability using seven different approaches: (1) Markov chains [76] (order 1-5), (2) hidden Markov model [219] (HMM), (3) vanilla recurrent

neural network [94] (Vanilla-RNN), (4) recurrent neural network with long short-term memory [108] (RNN-LSTM), (5) dilated recurrent neural network [37] (Dilated-RNN), (6) recurrent highway network [273] (RHN), and (7) pointer sentinel mixture model [161] (PSMM). We use the standard implementations of the predictive algorithms as described in their respective papers. Markov chains [76] and hidden Markov models [219] are implemented using the standard python libraries (hmmlearn). We use hyper-parameters stated in these works (Table 2.4). Vanilla-RNN [94], RNN-LSTM [108] and dilated-RNN [37] are based on predicting the next character (language modeling) in the text, whereas RHN [273] and PSMM [161] model the prediction task as multivariate classification. For dilated-RNN [37] we used the dilations of 1, 2, 4, 8, 16, 32 and 64 and provided the results for dilation 32 after which we observed a drop in the accuracy.

Table 2.4 – Hyperparameters selected for each recurrent neural networks (RNN) variant for the prediction accuracy measurement experiments.

RNN Variant	Hidden-Layer Size	Unroll Steps	Learning Rate	Activation Function	Optimizer	Dropout Rate
Vanilla-RNN	100	25	$1.0e-1$	tanh	Adam	0.2
RNN-LSTM	100	50	$1.0e-8$	ReLU	Adam	0.2
Dilated-RNN	100	32	$1.0e-6$	ReLU	Adam	0.2
RHN	100	50	$1.0e-8$	ReLU	Adam	0.2
PSMM	100	50	$1.0e-8$	ReLU	Adam	0.2

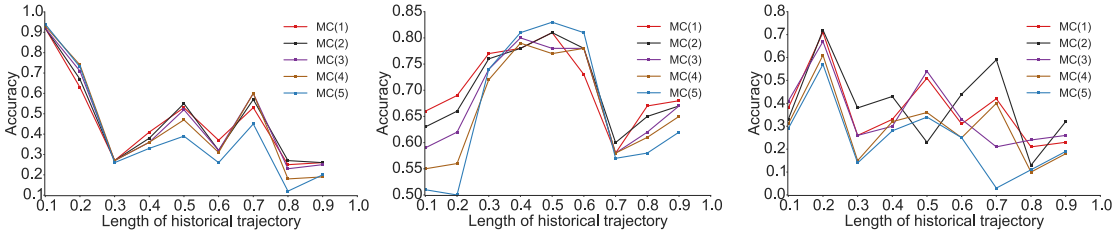
Prediction accuracy. We computed the prediction accuracy (π_{acc}) of a dataset by estimating the average accuracy across all the individuals present in that dataset. Along the lines of [154], we measured the individual prediction accuracy by the proportion of accurate predictions over all days of that individual (users who were not active on a day are excluded in the prediction). The accuracy of a model is given by Equation (2.8).

$$\pi_{acc} = \frac{\sum_{t=1}^{T} \mathbb{1}_{poi_t = poi_t^*}}{T}, \tag{2.8}$$

where poi_t is the true next point of interest of an individual at time t, poi_t^* is the predicted next point of interest and T is the total number of prediction time-steps. The data is split into 10 windows consisting of 10% training set and the subsequent 10% as test set as performed in [154, 226]. The training was performed in a cumulative manner such that the previous training instance was not lost. Such an approach also highlights the accuracy variations across the trajectory length in order to analyze the location dependencies and interaction distance.

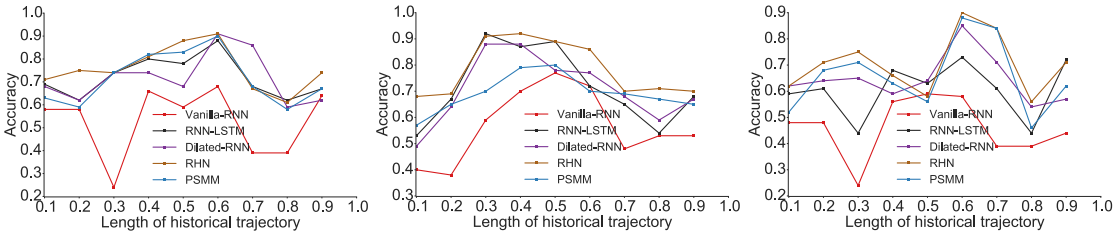
2.3.2 Confirming the Predictability Upper Bound Discrepancy

We found that higher order Markov chains (typically > 3) do not contribute to increased prediction accuracy, as also observed by Lu et al. [154]. The prediction accuracy for Markov chain models and the recurrent-neural architectures for all datasets is shown in Figures 2.1 and 2.2, respectively. The experimental results show the proportion of accurate predictions for each day (in terms of dataset duration) based on the length of the historical trajectory accounted for to train the predictive model.



(a) PrivaMov dataset. (b) NMDC dataset. (c) GeoLife dataset.

Figure 2.1 – Prediction accuracy for Markov models (order 1–5). The x-axis signifies the proportion of trajectory length considered for the train-test split and y-axis signifies the precision of the prediction model.



(a) PrivaMov dataset. (b) NMDC dataset. (c) GeoLife dataset.

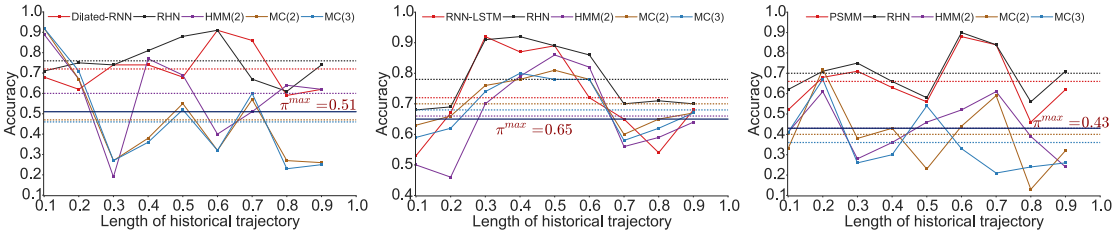
Figure 2.2 – Prediction accuracy for recurrent-neural architectures. The x-axis signifies the proportion of trajectory length considered for the train-test split and y-axis signifies the precision of the prediction model.

We observed that the accuracy of Markov models (π_{acc}) lies in the vicinity of π^{max}. It was also clearly evident that recurrent-neural architectures significantly outperform Markov models with respect to their average accuracies. In Table 2.5, we show the maximum predictability achieved by using the best performing models from each algorithm, and in Figure 2.3 we compare their performance with the theoretical upper bound.

Table 2.5 – Prediction accuracy achieved using the best performing models for each dataset.

Datasets	π^{max}	$\pi_{acc}(MC(2))$	$\pi_{acc}(MC(3))$	$\pi_{acc}(HMM(2))$	$\pi_{acc}(RHN)$	$\pi_{acc}(RNN)$
PrivaMov	0.50	0.47	0.46	0.60	0.76	0.72 (Dilated-RNN)
NMDC	0.65	0.70	0.68	0.66	0.78	0.72 (RNN-LSTM)
GeoLife	0.43	0.40	0.36	0.43	0.70	0.66 (PSMM)

We also observed that in addition to the prediction model, the dataset characteristics significantly impacted the average accuracy. The average accuracy was the highest for the NMDC dataset [110], followed by PrivaMov dataset [112] and then GeoLife dataset [162]. We hypothesize that the accuracy values were governed by four key properties of mobility datasets namely; (a) number of unique points of interest, (b) average length of the trajectories, (c) number of interacting point of interests, and (d) the distance between these interactions. We systematically validated these assumptions in Section 2.4. We found that the NMDC dataset contains fewer points of interest, shorter average trajectory length (as shown in Table 2.3) and shorter interaction distance



(a) PrivaMov dataset. (b) NMDC dataset. (c) GeoLife dataset.

Figure 2.3 – Comparison of π^{max} with the maximum predictability achieved using models from each category. The dotted lines indicate the predictability by each approach (indicated with the same colour). x-axis signifies the proportion of trajectory length considered for the train-test split and y-axis signifies the precision of the prediction model.

between the points as compared to the other two datasets (see Figure 12). Such short distance dependencies can be captured conveniently by Markov models of order 2, resulting in comparable accuracies to recurrent neural architectures. More importantly, variants of recurrent networks tend to show overfitting characteristics over datasets with smaller dependencies resulting in dropping accuracy on the validation set. This is the prime cause behind deep learning models showing poor performance against Markov models. We thus argue that precise quantification of dataset characteristics can guide towards selection of appropriate prediction models. The variation in the accuracy for a particular dataset with respect to the trajectory length under consideration stems from fluctuation of the interacting points and the distance within those interactions.

The prediction accuracies of recurrent-neural architectures also surpass the theoretical upper bound for the respective dataset. This anomaly in computing π^{max} is puzzling, even more so considering the diversity of the datasets with respect to their collective time spans, visited number of locations, demographics and spatiotemporal granularity.

2.4 Revisiting the Underlying Assumptions

In this section we revisit the underlying assumptions listed in Section 2.1.3 involved in the upper bound derivation methodology and perform statistical tests to invalidate these assumptions.

2.4.1 Questioning the Markovian Nature of Human Mobility

Current mobility models [154, 226] are based on the assumption that human mobility is memoryless or Poissonian. Such an assumption implies that consecutive events follow each other at relatively regular time intervals without the presence of very long waiting times. This Markovian assumption lies at the basis of the methodology used in deriving the upper bound for mobility predictability. The discrepancy between π^{max} and π_{acc} lets us questions the assumption that human mobility follows a Markov process. In this section, we conduct extensive analysis to validate the true nature of human mobility. More precisely we analyze the the distribution associated with

Table 2.6 – Candidate distributions used for assessing the power law fit to the statistical tests.

Name	Density $p(x) = Cf(x)$	
	$f(x)$	C
Power law with cutoff	$x^{-\alpha}e^{-\lambda x}$	$\frac{\lambda^{1-\alpha}}{\tau(1-\alpha,\lambda x_{min})}$
Exponential	$e^{-\lambda x}$	$\lambda e^{\lambda x_{min}}$
Stretched exponential	$x^{\beta-1}e^{-\lambda x^{\beta}}$	$\beta\lambda e^{\lambda x_{min}^{\beta}}$
Log-normal	$\frac{1}{x}exp[-\frac{(lnx-\mu)^2}{2\sigma^2}]$	$\sqrt{\frac{2}{\pi\sigma^2}}[erfc(\frac{lnx_{min}-\mu}{\sqrt{2}\sigma})]^{-1}$

Table 2.7 – Kolmogorov–Smirnov goodness-of-fit test for location rank-order distribution.

Rank Order	Power Law p	Log-Normal		Exponential		Stretched Exp.		Power Law + Cutoff		Support for Power-Law
		LR	p	LR	p	LR	p	LR	p	
Privamov	0.00	−12.72	0.00	−30.12	0.00	−11.42	0.00	−113.1	0.00	with Cutoff
NMDC	0.00	−11.28	0.00	−27.23	0.00	−13.95	0.00	−320	0.00	with Cutoff
Geolife	0.006	−17.04	0.00	−19.21	0.00	−18.21	0.08	−560.78	0.00	with Cutoff

several parameters of human mobility to check for slowly decaying, heavy-tailed processes.

In the following experiments, we check the power law fit using a Kolmogorov–Smirnov (K-S) statistic [241] based on the methodology adopted by Clauset et al. [52]. In order to estimate the likeliness of the data to be having drawn from the power law, we compute the p value and check its significance level. We also check the goodness of fit with other candidate distributions shown in Table 2.6 to exclude the possibility that no alternative distribution fits the data better than power law. We adopt the same approach for binned data as suggested by Virkar et al. [245]. The tests provide the log-likelihood ratio between the two candidate distributions R. This number will be positive if the data is more likely in the first distribution, and negative if the data is more likely in the second distribution. The significance value for that direction is p.

Location Rank-Order Distribution

In order to gain insight into the datasets, we first analyze the rank distribution of the locations, according to the visit frequency at individual and aggregated levels. An individual visits different locations depending on a perceived priority attached to the location [16]; this results in a heterogenous location frequency distribution [265]. To study the location-rank distribution, we follow the approach stated in Zhao et al. [265] in order to rank locations according to their collective magnitude at the aggregate level. Figure 2.4 shows the rank distribution of visited locations in human mobility and Table 2.7 proves the existence of power law scaling (Zipf's law [180]). We also observe a convergence and robustness at the individual level, which clearly indicates non-uniform mobility behavior and its effect on entropy, hinting at the non-Markovian nature of human mobility [265].

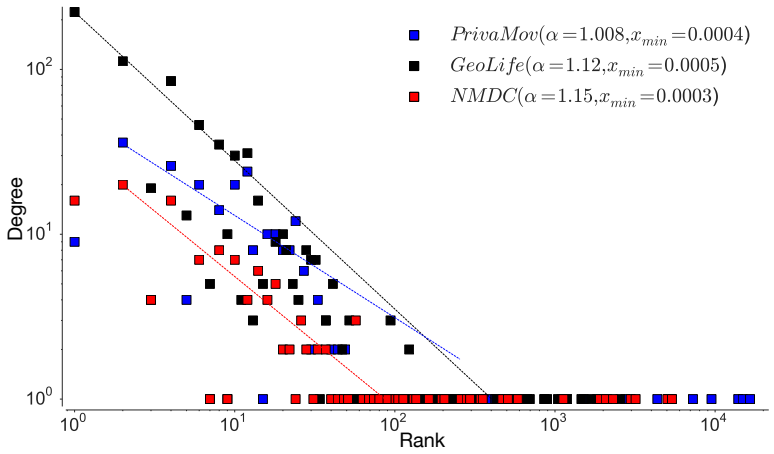


Figure 2.4 – Rank distribution of location visits at the collective level for aggregated dataset. The data is binned into exponentially wider bins and normalised by the bin width. The straight line represents the fitting through least squares regression (α and x_{min}, computed through maximum likelihood estimation).

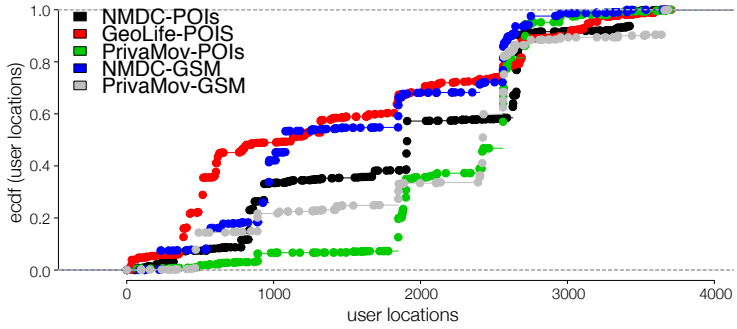


Figure 2.5 – Empirical cumulative distribution for dataset points of interests and GSM logs.

Maximum likelihood estimation and Kolmogorov–Smirnov test. Maximum likelihood estimation is a tool for estimating the parameters as a data mining model. It is a computationally tractable way to learn a model from the data. Herein, we perform such fits according to [52]. Kolmogorov–Smirnov test (K-S test) is a non-parametric methodology that compares an observed distribution to, $S(x)$ to a theoretical distribution $F*(x)$. In the above cases, the procedure consists of first forming the empirical cumulative distributions of $S(x)$ (see Figure 2.5) and $F*(x)$ and estimating the difference between the candidate distribution fits (Table 2.8). The test is based on the following statistic:

$$D = sup|F*(x) - S(x)|, \tag{2.9}$$

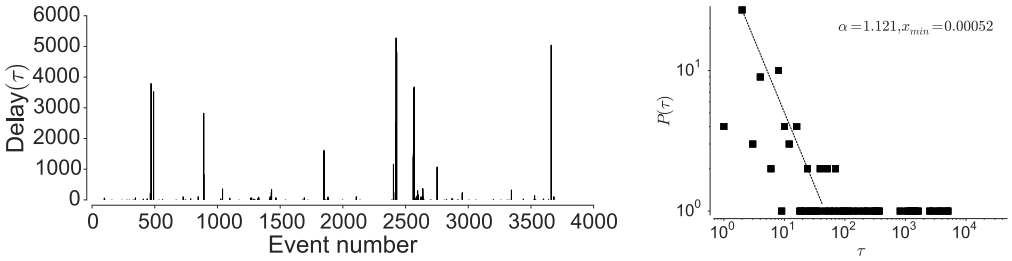
with smaller values of D indicating a better fit to the corresponding theoretical distribution.

Table 2.8 – Maximum likelihood and K-S test for the cumulative distributions (lower value in boldface indicates a better fit). We clearly observe that high granularity points of interest depict a power-law unlike the CDR logs which are a rough approximation of human mobility.

		MSE		
Measure	**Log-Normal**	**Exponential**	**Stretched Exp.**	**Power Law + Cutoff**
NMDC-POIs	0.04501	0.05648	0.02348	**0.00616**
GeoLife-POIs	0.00324	0.07306	0.00378	**0.00087**
PrivaMov-POIs	0.05824	0.09386	0.00739	**0.00114**
NMDC-GSM	0.25584	**0.00224**	0.00584	0.07268
PrivaMov-GSM	0.03655	0.00895	**0.00098**	0.00783
		K-S Test		
NMDC-POIs	0.65843	0.75615	0.07456	**0.00825**
GeoLife-POIs	0.63288	0.93644	0.04289	**0.00046**
PrivaMov-POIs	0.96752	0.69748	0.27896	**0.00116**
NMDC-GSM	0.56825	0.00987	**0.00967**	0.04568
PrivaMov-GSM	0.85621	0.00567	**0.00165**	0.00927

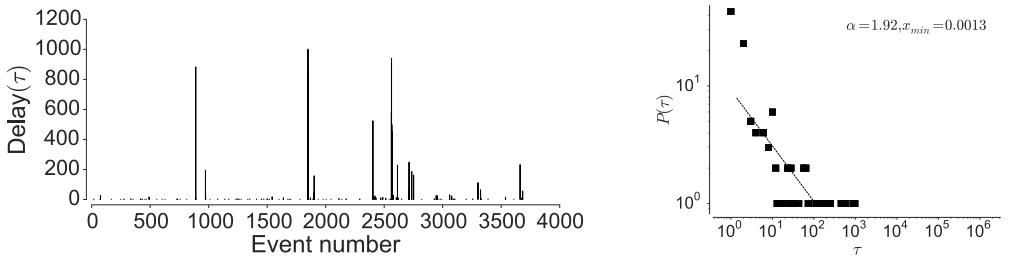
Inter-Event Time Distribution

To further confirm the non-Markovian nature, we check the distribution of the inter-event times associated with the individual locations. Here, visiting a particular location is considered as an event and hence time between two location visits is considered as inter-event time. The current mobility models are based on an assumption that human movements are randomly distributed in space and time, hence are approximated by a Poisson process [16, 180]. However, Barabasi [16] shows that human activities are non-Poissonian, by showing that inter-event timings depict long-tailed distribution. We observe a similar behavior when considering human mobility in all the datasets, when examining the inter-event and dwell times associated with each location; most locations are visited at high periodicity, while few locations encounter long waiting times. The current models assume that inter-event time follows exponential distribution [16], rather, we observe an emergence of power-law as seen in Figures 2.6–2.8 corroborated by the statistical tests shown in Table 2.9. The spikes in the plot correspond to delays and display the visit regularity, which indicates a long-tailed process. The delay-time distribution depicts the priority list model in human mobility, bearing similarity to other activities as remarked by Barabasi [16]. When an individual is presented with multiple events under the context of mobility, the next location is determined on a perceived priority, thus resulting in power-law dynamics in inter-location waiting times [16]. This shows that the dwell-times associated with human mobility are not memoryless, hence cannot be considered as Markovian. In the above analysis, we also observe a convergence between individual mobility patterns and aggregated datasets, which concurs with the observations of Yan et al. [253].



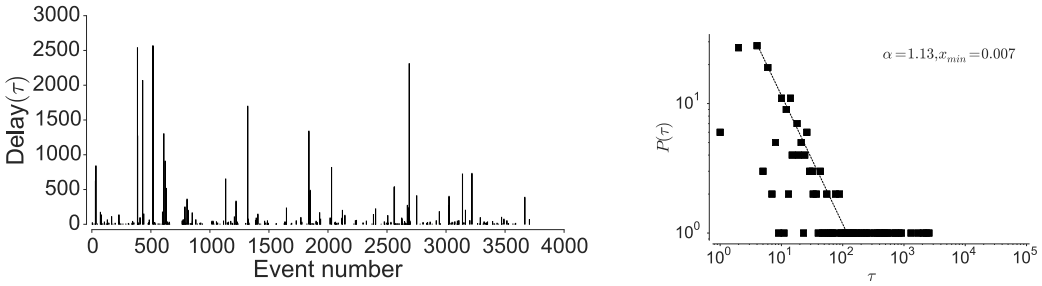
(a) Event distribution. (b) Delay time distribution.

Figure 2.6 – Distribution of the location visits and the delay between the visits in PrivaMov dataset.



(a) Event distribution. (b) Delay time distribution.

Figure 2.7 – Distribution of the location visits and the delay between the visits in NMDC dataset.



(a) Event distribution. (b) Delay time distribution.

Figure 2.8 – Distribution of the location visits and the delay between the visits in GeoLife dataset.

Mutual Information Decay

We validate Lin et at. [148, 149] observation on mobility data where they state that; I as a function of the number of symbols (locations) between any two symbols and state that it would decay with a power-law for any context-free grammar and hence must be non-Markovian. With respect to human mobility trajectories, I between two location instances is the realization of a discrete stochastic process, with separation τ in time [148]. In order to analyze the existence of power law decay indicating the presence of memory in mobility trajectories we first consider the GeoLife [266] which is collected at a uniform sampling rate (location/5 s). We first validate the emergence of power law at distinct sampling rates by undersampling and oversampling the dataset

Table 2.9 – Kolmogorov–Smirnov goodness-of-fit test for inter-event time distribution.

Inter-Event Times	Power Law p	Log-Normal		Exponential		Stretched Exp.		Power Law + Cutoff		Support for Power-Law
		LR	p	LR	p	LR	p	LR	p	
Privamov	0.12	−1.13	0.28	5.69	0.00	0.09	0.00	−0.34	0.74	with Cutoff
NMDC	0.08	−0.11	0.02	2.98	0.00	3.78	0.54	−2.87	0.31	weak
Geolife	0.86	−7.76	0.00	−20.43	0.00	17.87	0.08	−0.30	0.59	good

by a factor of two and four. We perform oversampling by using semivariance interpolation [97]; a commonly used spatial interpolation scheme that fits the missing points by modeling the similarity between the points as a function of changing distance.

Mutual information between two location symbols is computed the estimating entropy of the marginal distribution of discrete random variables X and Y, and the joint entropy of discrete random variables X and Y as in Equation (2.10).

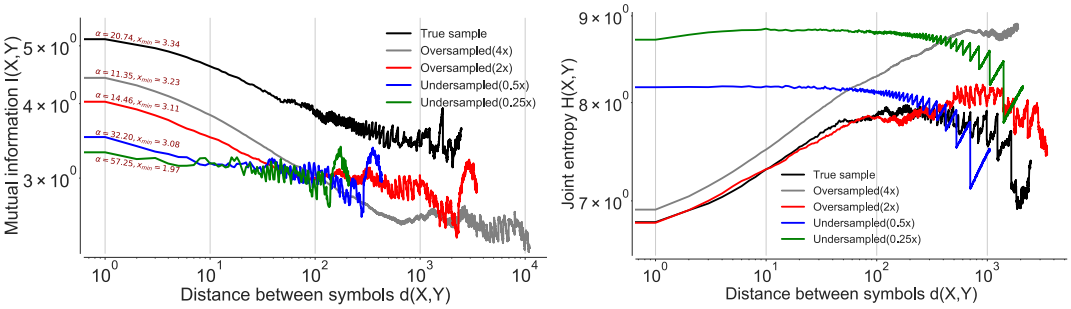
$$I(X,Y) = H(X) + H(Y) - H(X,Y) = D_{KL}(p(XY)||p(X)p(Y)), \tag{2.10}$$

where $H(X)$ is the entropy of a random variable X and $H(X,\ Y)$ is the joint entropy of X and Y. D_{KL} is the Kullback–Liebler divergence [191]. Thus, mutual information is same as the Kullback–Leibler divergence between distributions of X and Y. In order to compensate for insufficient samplings, we use the following adjustment proposed by Grassberger et al. [91] (Equation (2.11)) to compute $H(X)$, $H(Y)$, $H(X,Y)$.

$$H(X) = \log N - 1/N \sum_{i=1}^{k} N_{\mathrm{i}} \psi(N_{\mathrm{i}}). \tag{2.11}$$

Thus, we first estimate the distribution of X from index 0 followed by the distribution of Y at some index d, where the random variables X and Y are sampled from the individual trajectory sequence. d is then varied to compute long-distance dependencies at every separation by creating displacements between the random variables. Once the contextual dependence limit is reached, the process starts sampling noise, which sets the termination criterion and then the average similarity between the two symbols is quantified.

As shown by Lin et al. [148], we observe a power-law decay at all the sampling rates (see Figure 2.9 and Table 2.10). This experiment validates the presence of LDDs in location sequences irrespective of their sampling rates. However, contrary to what would be expected that I would increase and decrease by the factor of under/over sampling, we observe a decrease in I for all the contexts in which the true distribution of the data is altered. We also observe that the reduction is proportional to the Kullback–Leibler divergence [191] between their respective distributions. The reduction in I stems from the fact that a change in the distribution results in the alteration of the true correlation between the location pairs. The true distribution will therefore show maximum I, compared to the cases when either artificial pairs are introduced (oversampling) or true pairs are removed (undersampling) from the dataset.



(a) Mutual information decay

(b) Joint entropy

Figure 2.9 – Mutual information decay for the GeoLife dataset at different sampling rates of the raw GPS coordinates projected onto a grid through Google S2 [89]. The upsampling was performed by the semivariance interpolation scheme [135].

Table 2.10 – Kolmogorov–Smirnov goodness-of-fit test for mutual information decay of GeoLife dataset at varying sampling rates.

Sampling Rate	Power Law p	Power Law + Cutoff		Log-Normal		Exponential		Stretched Exp.		Support for Power Law
		LR	p	LR	p	LR	p	LR	p	
1X	0.51	5.43	0.19	0.278	0.47	9.89	0.96	4.32	0.12	good
2X	0.06	0.00	0.07	−1.25	0.08	2.89	0.11	10.08	0.00	with Cutoff
4X	0.46	−0.065	0.67	−0.072	0.87	1.89	0.87	1.78	0.07	moderate
0.5X	0.00	0.00	0.00	−5.54	0.01	8.66	0.38	11.88	0.00	with Cutoff
0.25X	0.00	0.00	0.02	−1.78	0.03	9.94	0.04	13.56	0.00	with Cutoff

To verify our hypothesis, we calculate the joint entropy for all the cases and observe an increase in $H(X,Y)$ for the altered distributions as shown in Figure 2.9b. We see that the increased entropy is due to an increase in the ratio between unique pairs in the dataset over the total number of pairs. The introduction of spurious pairs scrambles the true distribution as it leads to introduction of data points in the true sequence, thereby changing the random variables sampled at distance d, hence reducing I. This occurrence was confirmed after computing the area under the receiver operator characteristic (ROC), which was maximum for the true data distribution in the first quartile as compared to the rest as shown in Figure 2.10. This explains our observation of higher joint entropy for the oversampled and the undersampled case. This experiment also confirms that sampling rate of location coordinates would have a significant impact on the estimation of π^{max} as also identified by [222].

After validating the existence of power law at different sampling rates, we analyze and quantify the presence of long-range correlations in other datasets. We observe a power law decay across all the datasets and their respective joint entropy, as shown in Figure 2.11a,b and corroborated by the statistical tests shown in Table 2.11. This information also serves as basis for the difference in accuracy for each dataset and the performance difference between the prediction algorithms. We further explore the Markov transition matrices for these datasets and observe that they are reducible and periodic, resulting in the decay of I to a constant. It has been shown that such a characteristic of the transition matrix cannot result in an exponential decay by Lin et

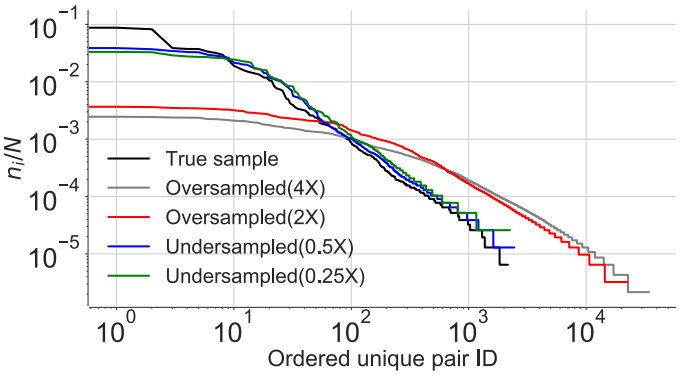
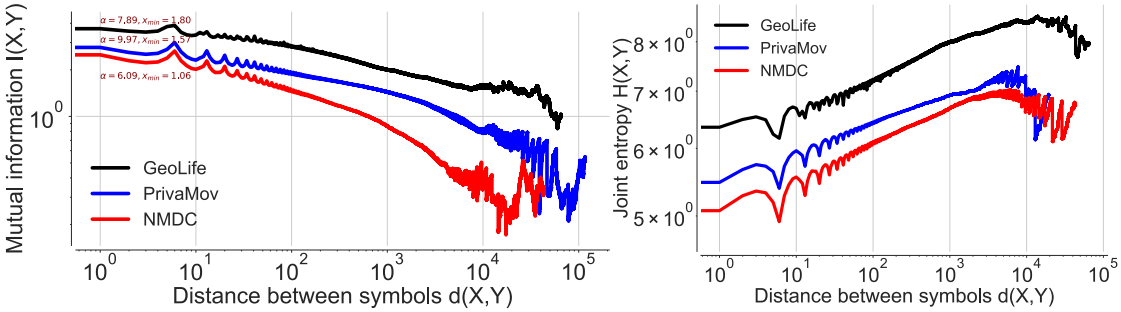


Figure 2.10 – Location pair occurrences across all the sampling rates of the true sample. The x-axis represents the unique pair ID in the descending order of their frequency of occurrence. The y-axis is the ratio between the unique pairs and the total number of pairs contained in the an individual trajectory.



(a) Mutual information decay.

(b) Joint entropy.

Figure 2.11 – Mutual information decay and joint entropy estimated for all the datasets. The dataset consists of stacked sequences of temporally arranged individual points of interest.

Table 2.11 – Kolmogorov–Smirnov goodness-of-fit test for mutual information decay across all the datasets.

Dataset	Power Law p	Power Law + Cutoff		Log-Normal		Exponential		Stretched Exp.		Support for Power Law
		LR	p	LR	p	LR	p	LR	p	
Privamov	0.43	3.25	0.69	1.78	0.28	6.28	0.83	4.89	0.34	good
NMDC	0.27	1.82	0.11	−0.27	0.10	2.47	0.65	2.21	0.16	moderate
Geolife	0.51	5.43	0.19	0.278	0.47	9.89	0.96	4.32	0.12	good

al. [148, 149]. This phenomenon is seen in a number of cases, including hidden and semi-Markov models [148, 149].

In the literature, such behavior is superficially dealt with by increasing the state space to include symbols from the past, which does not address the main issue [149] with Markov models; lack of memory. This analysis shows that GeoLife dataset consists of considerably higher number of long-range correlations, compared to the PrivaMov dataset and the NMDC dataset. This should be self-evident from their respective data collection durations. However, the lower dependencies

in the NMDC dataset, compared to PrivaMov, is due to the smaller area of the data collection region, which generally results in lower entropy of movement [154, 226].

Here, we reason about the accuracy variation within and between the datasets and about the performance differences between the prediction algorithms. We observe that the NMDC dataset provides higher accuracy as compared to the other datasets, and witness a lower variation within the accuracies of different algorithms. This stems from the presence of very short dependencies in the individual trajectories present in the dataset, as seen in Figure 2.11a. The lower correlations also result in roughly equivalent prediction accuracies within the predictive models. The lower accuracies of recurrent-neural architectures, compared to Markov chain at some time-steps are due to the models tendency to actively seek for long-range dependencies. However, if the dataset does not contain such dependencies, the model underperforms, unless it is weighted to account for such an existence. This underperformance is evident from the behavior of dilated-RNN's, where an increase in dilations (to account for longer dependencies) results in dropping accuracy. Such a phenomenon has also been observed in language modeling tasks, which suggests that this is not a domain specific occurrence [122]. The performance drop in the recurrent-neural architectures at different time steps is due to capturing the long-distance dependencies to different degrees, resulting in either under/over fitting. An additional reason for higher accuracy in NMDC dataset is due to a lower number of unique locations and smaller variations in the dwell-times, as compared to the PrivaMov and GeoLife datasets, as shown in Figures 2.4 and 2.7. These aspects directly correlate with the entropy and thus affect predictability [226]. We also observe that PSMMs perform better on GeoLife dataset, compared to other two, due to its ability to search for dependencies at longer distances.

Our analysis of all the tests in this section, provides a compelling evidence that human mobility is characterized by a non-Markovian nature. More specifically, the presence of power law decay in these tests indicate a presence of memory which cannot be modeled by Markov processes. Furthermore, the diversity of the considered datasets with respect to the data collection region, duration, radius of gyration and sampling rate shows that this phenomenon is observable across disparate mobility behaviors. We thus invalidate the long held assumption that human mobility is Markovian by several prior works and confirm our first hypothesis which could have resulted in the inaccurate estimation of the predictability upper bound. In the next section, we analyze the impact of this assumption on the derivation of mobility entropy S^{real} and consequently the predictability upper bound.

2.4.2 Questioning the Asymptotic Convergence of the Entropy Estimate

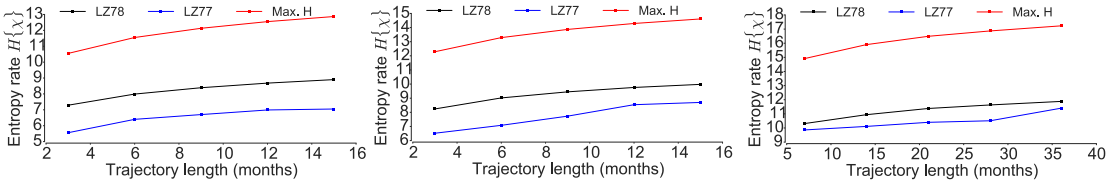
In this section, we investigate whether the entropy estimation schemes used in the current works provide an accurate characterization of the mobility entropy. Entropy estimation is the most crucial step towards computing the upper bound on mobility predictability using the Fano's inequality [80, 200]. We compare two significantly different variations of the Lempel–Ziv encoding algorithms with respect to their entropy estimates.

To this end, we check the scaling behavior of two variants of the Lempel–Ziv algorithm, the LZ78 [275] and the LZ77 scheme [274]. The current works [111, 154, 226, 265] rely upon LZ78 data compression scheme [275] to compute the mobility entropy. The LZ78 scheme segments the complete trajectory sequence into substrings, where a substring is defined as the shortest subsequence in terms of its length, yet to be encountered. Song et al. [226] estimate entropy rate of an individuals trajectory according to Equation (2.7).

Theoretically, for a Markov process (of any order) Lempel–Ziv compression algorithms are optimal in achieving the compression limit put forth by Shannon and thus can be leveraged to estimate the entropy rate [211, 225]. As it is non-trivial to estimate the entropy rate of information sources with strong long range correlations, we compare the two approaches with respect to their convergence. Assuming a binary sequence $S = (1,0,1,1,0,1,0,1,1,0,1,1,0)$, LZ78 coding will break words $w_1, w_2...$ in a sequence S such that $w1 = s1$ and w_{k+1} is the shortest new word immediately following w_k. Thus S will be broken down into $(1),(0),(11),(01),(011),(0110)$, here each word w_k with $k > 1$ is an extension of w_j with $j < k$ by one single symbol $s' \in A$. LZ77 coding on the other hand, does not necessarily break w_k as an extension of a previous word w_j, but can be an extension of any substring S, starting before w_k and may even overlap it. Therefore, LZ77 will break down the sequence S as $(1),(0),(11),(010),(11011)$.

The LZ77 scheme uses string-matching on a sliding window; whereas the second, LZ78, uses an adaptive dictionary. Furthermore, LZ77 coding does not necessarily break a substring as an extension of a previous subsequence and may therefore overlap it. Here, the average word length increases faster and the algorithm can make better use of long-range correlations. This stems from Grassberger's [90] result, which states that as the block length increases more correlations are taken into account as a result of information/symbol decreasing with the number of elements in a block.

As seen in Figure 2.12, LZ77 clearly results in lower entropy as compared to the LZ78 scheme as observed by Storer et al. [229] who shows that LZ78 cannot truly capture long-range dependencies present in the sequence. One of the reasons for this as Schurmann [211] points out; LZ78 scheme based on shorter words is more efficient in the case of Bernoulli sources. However, in the case of the logistic map, the convergence of LZ77 scheme is faster then for the memoryless case. Thus, although LZ77 operates in the ignorance of the source statistics, it compresses the sequence better as compared to LZ78. However, we emphasize that it is still not the optimum scheme to compute the entropy as the information carriers of the sequence lie in its structural origin. We simply show that, the entropy measure provided by LZ78 scheme adopted by Song et al. [226] does not attain convergence. The maximum entropy here is computed by $log_2(k)$, where k is the cardinality of trajectory sequence. Grassberger [90] furthermore points out that LZ78 [275] and LZ77 [274] attain their claimed asymptotic behavior only when applied to Markov sequences. However, as previously established human mobility is not memoryless and therefore Markov property is not applicable in this case.



(a) PrivaMov dataset. (b) NMDC dataset. (c) GeoLife dataset.

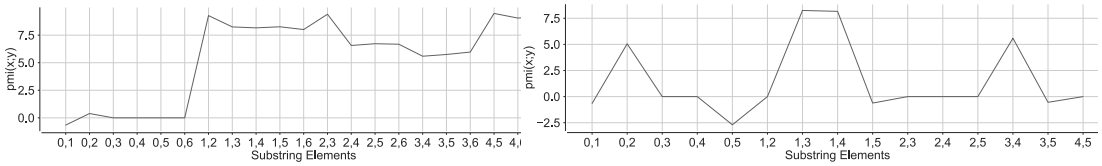
Figure 2.12 – Comparison of entropy derived using LZ78 and LZ77 encoding algorithms. The red curve is the maximum entropy.

2.4.3 Questioning S^{real} as a Relative Entropy Estimate for Human Mobility

Next, we inspect whether the current methodology ignores the presence of any long-range correlations present in the mobility sequence. In order to perform the above step, we compute the pointwise mutual information of an individual mobility trajectory. In order to analyze the long-distance dependencies between the elements of the individual substrings extracted by LZ78, we compute the PMI. This serves as a measure of the dependencies missed when the elements are grouped in distinct substrings as PMI computes the information provided by a symbol about another at a given distance d. Thus, we provide empirical evidence that the current entropy estimation scheme does not account for all the dependencies present in sequence.

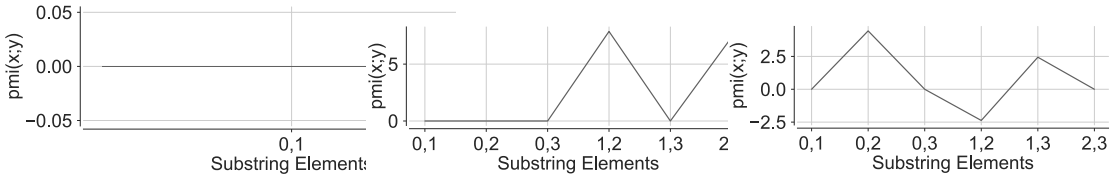
We first see that a vast majority of substrings are of length one or two, which are dominant contributors to the entropy as also observed by Lesne et al. [144]. The estimated entropy is thus the outcome of finite-size fluctuations; and the total count of the substrings and of the elements in a substring does not represent the true probability distribution. As evident from Figure 2.13 the structural correlations between the elements of the individual substrings are ignored in case of long substring (number of elements > 5) but more surprisingly even in the case of short substrings (number of elements < 4) as seen from Figure 2.14. These correlations are ignored based on the argument that the probability of joint occurrences is weak [144]. This argument stems from the reasoning that the parsed substrings are independently and identically distributed according to Gaussian distribution, that does not apply to mobility trajectories. Finally, the correlated features can be compressed only by memorizing all the cases of intervening random variables between the correlated instances. [229]. It has thus been proved that Lempel-Ziv approach fails to capture redundancies in the data sources with long-range correlations [144].

Furthermore, as evident from Equation (2.7) the Lempel-Ziv approach limits the entropy estimation process at the sub-string level. Given that entropy is the complete quantitative measure of the dependency relations (including many point correlations), the computation of higher-order entropy is non-trivial. Therefore, it is flawed to assume that the π^{max} derived from such an approximate estimation of S^{real} should act as an upper bound of predictability on trajectories compiled for long time-spans. He shows that these bounds are tight if the sequence has no long-range correlations or more precisely, $h_n = h$ for all $n \geqslant N$ is the sequence is an N^{th} order Markov chain. However, in case of mobility traces, these correlations are very strong (our experiments based



(a) PMI substring 1. (b) PMI Substring 2.

Figure 2.13 – Pointwise mutual information across longer substrings in a user trajectory. The x-axis denotes the index's of element pairs in a substring derived from a user trajectory using LZ78 encoding algorithm. The y-axis denote the pointwise mutual information between the element pairs.



(a) Zero correlation. (b) Two-pair positive correlation. (c) Two positive, one negative.

Figure 2.14 – Pointwise mutual information across short substrings in a user trajectory. The x-axis denote the index's of element pairs in a substring derived from a user trajectory using LZ78 encoding algorithm. The y-axis denote the pointwise mutual information between the element pairs.

on mutual-information (MI) decay verify this claim) and h_N converges very slowly. We thus present an empirical evidence that the current approach used to estimate S^{real} is not the true entropy associated with the mobility trajectory. But, it is in fact the entropy estimate derived by ignoring all the dependencies present within the individual elements of the substring extracted by the Lempel–Ziv encoding schemes. Ignoring the dependencies inflate the S^{real} estimate and thus lowers the π^{max}.

After estimating the mobility entropy S^{real} as described in above, it is used to compute the predictability upper bound using Fano's inequality [80]. We now present this notion to understand their inter-relation. Consider estimating a random variable X, by an estimator $\widehat{X}$ under the assumption that $\mathbb{P}(\widehat{X} \not\equiv X)$=$\varepsilon$.

Joint entropy $H(X|\widehat{X})$ is the average number of bits required to be transmitted in order to estimate X with the knowledge of $\widehat{X}$. Fano's inequality upper bounds this notion of estimating X given $\widehat{X}$. Now, consider that we utilize some bits to communicate if X is $\widehat{X}$ or not. The distribution for this is $P_e, 1-P_e$, i.e. we need to transmit $H(P_e)$ bits on an average to successfully execute this task. If X is not $\widehat{X}$, then it could be any one of the other $|\chi|-1$ symbols in the alphabet (location point in the set of all possible locations). As a result the worst case length is $log(|\chi|-1)$ with a probability of P_e. Therefore Equation (2.12) quantifies this notion.

$$H(X|\widehat{X}) \leqslant H(P_e) + P_e.log(|\chi|-1) \leqslant H(P_e) + P_e.log(|X|). \tag{2.12}$$

Fano's inequality, rooted in information theory [8], is intended for a data source with a well known probability distribution [80] which may not apply for mobility trajectories due to sampling, discretization and filtering schemes. Furthermore, the estimation of entropy by using Lempel–Ziv coding [275] was originally constructed to provide a complexity measure for finite sequences, i.e., input sequence displaying exponential decay in long-range correlations (memoryless structure). In this section, we thus demonstrate that when a mobility trajectory is further split in to smaller subsequences, the true distribution of the data is altered. This increases the associated entropy; and the derived π^{max} thus acts as a limit on the Markov model.

2.5 Discussion

Even though large strides have been made in the field of human mobility modeling, much has to be done to understand the underlying dynamics of mobility behaviors. Through analyzing three large scale datasets containing mobility trajectories of individuals from several different countries, we still observe the discrepancy between π^{max} and π_{acc}. Beneath this observation lie analysis for several assumptions made by previous works which has led to this inaccurate upper bound for mobility modeling. We have explored these assumptions from the lens of various information theoretic approaches such as mutual information, coding technics along with long-distance dependencies. In this Section, we discuss the observations and provide a nuanced explanation of these observations.

Why do RNNs perform better? A key step in modeling mobility behavior is to interpret the characteristics of LDDs present in the mobility trajectories. As for a Markov process, the observations at t_n depends only on events at previous time step t_{n-1} or on previous n time-steps for an n-order Markov chain. Under such a context, the maximum possible predictive information is given by the entropy of the distribution of states at one time step, which is in turn bounded by the logarithm of the number of accessible states. Unlike Markov chains, the recurrent-neural architectures, such as RHN's, approach this bound while maintaining the memory long enough so that the predictive information is reduced by the entropy of transition probabilities. Furthermore, the characteristics of LDDs depend on the number of interacting symbols and the distance between each interacting symbol, which is non-trivial to be modeled by a Markov process. In order to quantify LDDs, we use mutual information due to its simplicity and domain independence. As shown by Lin et al. [148], the mutual information decay offers some insights into why recurrent-neural architectures exceed probabilistic models in terms of capturing LDDs lying at multiple timescales. The ability of RNN's to reproduce critical behavior stems from its architecture, where a long short-term memory (LSTM) cell will smoothly forget its past over a timescale of approximately $\log(1/f) \equiv \tau_f$. However, as described by [148] for timescales $\geqslant \tau_f$ the cells are weakly correlated and on timescales $\leqslant \tau_f$ the cells are strongly correlated. Therefore, a cell can remember its previous state for τ_f time steps and then grows exponentially with the depth of the network. At each successive layer, the gradient flow becomes exponentially sparse, which governs the growth of the forget timescale [148]. It has been recently shown that understanding the characteristics of LDDs can lead towards selection of better hyper-

parameters for a model [157]. For instance understanding the scale of the dependencies can aid in selecting a suitable network depth, or the dilations of the dilated-RNN. In this work, we do not perform hyper-parameter tuning, which could have resulted in even higher estimates of π^{max}. Although estimation of a true upper bound is impractical, we hypothesize RNN models such as hierarchical-multiscale RNNs [50] could potential provide a very good π^{max} estimates by capturing dependencies existing at several timescales.

Systematic bias. The wide range of the upper limit of mobility prediction in the previous works arise mainly due to the difference in the dependencies in their respective datasets collected for varying timespans. Other factors such as demographics, spatiotemporal resolution, radius of gyration, filtering and discretization schemes have a minor impact for longer duration datasets, typically exceeding three to four years. These factors gain importance in determining upper bounds and interpreting results of predictive performance for short duration datasets lasting one to two years. The previous research [154, 226] estimated S^{real} and pi^{max} by using CDR datasets spanning a period of three to five months. Such datasets do not truly capture features such as the total number of unique locations visited by an individual, due to its low granularity (typically 4–5 km [111]). This results in a dataset with a masked entropy and mobility patterns ignoring long-range correlations. An important point to note is that for very short distances, power-law decay and exponential decay may not be trivial to differentiate [180]. This was due in part due to the fact that previous works [154, 226] were only studied for short distances of human mobility and not due to unavailability of high granularity GPS datasets. Therefore, the assumptions underlying the computation of S^{real} and π^{max} would have been fairly easy to overlook.

Reinforcing this bias. The aforementioned inadequacies would reinforce the empirical validation of π^{max} using Markov chains. However, as mentioned above, this would result in an error-prone estimation of predictability. As seen in other domains of sequential-data modeling such as natural language processing, Markov chains are fundamentally unsuitable for modeling such processes [47]. Our empirical observations, backed by theoretical foundations, indicate that human mobility will be poorly approximated by Markov chains. This is particularly true for trajectories that satisfy criteria of long time-span of collection.

Non-triviality of entropy estimation. It is non trivial to estimate the true entropy of mobility trajectory as the dependencies lie at several structural levels. Furthermore, the repeating patterns are typically hierarchical and they lie at various timescales. These scales depend on the mobility behaviors of the individual and therefore challenging to formulate a generic model. A more sophisticated description of these structures determining the mobility characteristics can be provided as more of the trajectory is observed. This results in an increase in the number of parameters in the model. That is, when we examine trajectories on the scale of individual coordinates, we learn about the rules of combining these points into points of interest and the transition paths between them. At the next level, if we consider several of these points of interest and the paths, we learn the rules for combining these points into semantic patterns. Similarly, when we look at semantic patterns, we learn about the visitation periodicities and circadian rhythms associated with the mobility behaviors. Therefore, longer traces have an increasing

number of long range structural correlations that are non-trivial to be captured by the currently available entropy measure. One consequence of ignoring these structural properties is that the missed regularities are converted to apparent randomness. We empirically showcase this by computing the pointwise mutual information of the trajectories under consideration. We demonstrate that this problem arises particularly for small data sets; e.g., in settings where one has access only to short measurement sequences. Moreover, the current approximation implies that the substrings have the same compressibility factor [275], hence the results derived from this approach would coincide with the average. Thus, the current computation will result in higher estimates of entropy, consequently resulting in a lower predictability bound.

Effect of dataset characteristics on accuracy. As is clear from the accuracy charts that different datasets result in different accuracy values. Furthermore, we also observe variations in the average accuracy across the length of the trajectory. We highlight that determining the key characteristics of the dataset that affect the accuracy is not trivial. However, based on our experiments we find that the following factors indicate a correlation:

- number of unique locations present in the trajectory,
- length of the trajectory and the size of the dataset,
- number of interacting locations within a long-distance dependency,
- distance between the interacting locations.

We argue that precise quantification of the above characteristics could provide insights regarding the accuracy variations. More importantly the quantification of dataset characteristics can guide towards selection of appropriate prediction models.

All is not lost for Markov processes. Even though Markov models tend to underperform in modeling human mobility, their use in human mobility prediction is not entirely without interest. In fact, considering their low computational complexity, it might be advantageous to opt for a Markov model when a dataset contains short-distance dependencies and low number of unique locations. However, in datasets exhibiting LDD characteristics, long-range correlations appear in the vicinity of the system critical point, which can benefit from recurrent-neural architectures to accurately model human mobility. Therefore, quantifying the LDD characteristics of a dataset can aid in inferring where Markov models are applicable.

2.6 Conclusions

In this work, we scrutinized the methodology behind the upper bound estimation of human mobility prediction upon confirming the discrepancy of this limit with extensive experimentation. To this end, we revisited all the steps involved in the derivation of the upper bound. We first confirmed the discrepancy between π_{acc} and π^{max} by analyzing three mobility datasets and

seven widely contrasting prediction models. We then systematically analyzed the assumptions underlying the derivation of π^{max} and highlighted their shortcomings. We demonstrated the non-Markovian character in human mobility by conducting the statistical tests which confirmed the emergence of scaling laws in the distributions of dwelling times and inter-event times. We showed that mobility trajectories contain scale-invariant long-distance dependencies similar to natural languages unaccounted for by the upper bound computation methodology. We further quantified these dependencies measured by a power-law decay of mutual information and we claim that these assumptions culminate into the computation of an inflated entropy measure. We also showed that the exponent characterizing this decay is well defined for infinite sequences, however for mobility trajectories the accuracy of the analysis is restricted by the length of the substrings and their entropy. This explains why the empirical accuracy results surpass the theoretical upper bound in several previous research works and in our own experiments. Finally, we argued that the precise estimation of the predictability upper bound can be determined only when all the long-distance dependencies present in human mobility trajectories are accounted for by an entropy estimation scheme. However, we emphasized that usage of Markov models for modeling human mobility is still sometimes justified considering their low complexity for datasets containing short dependencies.

3 Extracting Hotspots without A-priori by Enabling Signal Processing over Geospatial Data

Abstract

The proliferation of mobile devices equipped with internet connectivity and global positioning functionality (GPS) has resulted in the generation of large volumes of spatiotemporal data. This has led to the rapid evolution of location-based services. The anticipatory nature of these services, demand exploitation of a broader range of user information for service personalization. Determining the users' places of interest, i.e. *hotspots* is critical to understand their behaviors and preferences. Existing techniques to detect hotspots rely on a set of *a-priori* determined parameters that are either dataset dependent or derived without any empirical basis. This leads to biased results and inaccuracies in estimating the total number of hotspots belonging to a user, their shape and the average dwelling time. In this paper, we propose a parameter-less technique for extracting hotspots from spatiotemporal trajectories without any *a-priori* assumptions. We eliminate parameter dependence by treating trajectories as spatiotemporal signals and rely on signal processing algorithms to derive hotspots. We experimentally show that, our technique does not necessitate any spatiotemporal or behavior dependent bounds, which makes it suitable to extract hotspots from a larger variety of datasets and across users having disparate mobility behaviors. Our evaluation results on a real world dataset, show accuracy rates exceeding 80% and outperforms traditional clustering techniques used for hotspot detection.

Keywords: Spatiotemporal hotspots; Clustering parameters; Signal processing.

3.1 Introduction

An integral aspect of location-based services (LBS) is to extract meaningful information from the location trajectories recorded by their users. For example, mobility prediction services rely on clustering algorithms to extract user specific points of interest from raw GPS trajectories. LBS typically depend on mobility prediction as a means to improve quality of service by pushing context-aware information to users ahead of time. Other services such as traffic management, urban planning and consumer profiling, heavily rely on their ability to identify the hotspots where moving entities spend a considerable amount of time. Hotspot detection is therefore a key aspect of LBS for user mobility modeling.

Several techniques for extracting hotspots from trajectories were inspired by well-known clustering algorithms such as k-means [102] and DBSCAN [64]. Other methods span the domain of scan statistics, fingerprinting, gradient-based or eigenvector-tensor based techniques [67, 236, 264]. Overall, these techniques rely on a set of parameters that reflect *a-priori* assumptions about the mobility behavior of users by imposing bounds on distance, speed, time, number of points and/or visitation repeatability rates. These techniques require multiple steps, involving several iterations over the dataset to extract all the hotspots resulting in an increased latency. Furthermore, the hotspots are assumed to fit a predefined shape (mostly circular) which rarely reflects the reality, leading to erroneous estimations of hotspot area and dwelling time.

To address these problems, we propose a hotspot detection technique that is independent of such *a-priori* assumptions. We treat user mobility trajectories as spatiotemporal signals (see Figure 3.1) and apply filtering modules to iteratively extract and enhance the quality of the detected hotspots. We show that, these signals preserve all the key knowledge contained in the trajectories, and our system is able to accurately detect the hotspot occurrences, the time of hotspot entry and exit and a precise representation of the total area and time spent at each hotspot. We evaluate the extracted hotspots in terms of precision and recall rates and compare its efficacy with respect to popular clustering techniques used for hotspot detection. Applying signal processing algorithms also has the added benefit of exploiting the digital-signal processors (DSP) embedded in modern smartphones. This in turn preserves the privacy of users by restricting the computations on their smartphones by not sharing the raw data with untrusted third-party services.

The rest of the paper is structured as follows. The related work on hotspot detection and the associated drawbacks are presented in Section 3.2. We present the problem statement addressed in this paper and the translation process from trajectories to signals in Section 3.3 and Section 3.4 respectively. The system design and implementation is described in Section 3.5. The evaluation results and discussion is presented in Section 3.6. We finally conclude the paper in Section 3.7.

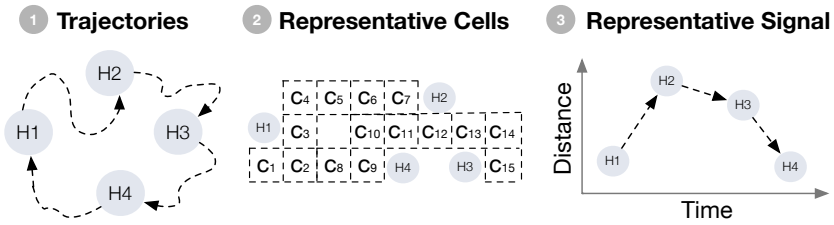
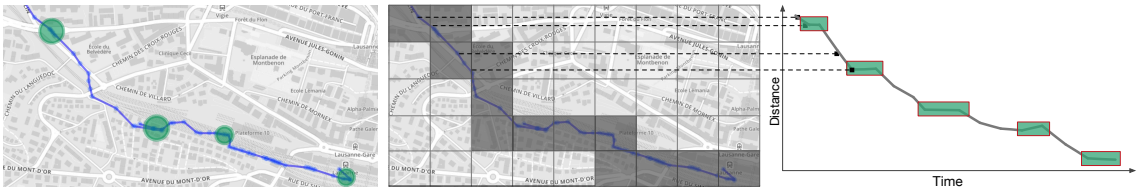


Figure 3.1 – From 3-D traces to 2-D signals



(a) A typical user trajectory. The green circles indicate user hotspots (b) The trajectory after being projected on to the google S2 cells (c) The signal generated after plotting the cell IDs' against timestamps

Figure 3.2 – Translating geospatial trajectories to space-time signals (left to right)

3.2 Related Work

In this section, we review the existing techniques to mine hotspots from spatiotemporal data. The premier contribution in adopting the data clustering techniques for hotspot detection was made by Ashbrook et al. in [11]. They propose an iterative approach to extract the hotspots and improve their granularity by imposing spatiotemporal bounds. These bounds are derived by analyzing their variance with respect to certain values they affect. Montoliu et al. proposed a two-level clustering approach in [170], where the geolocated points are first clustered in the temporal domain to discover the stay points that are used to derive stay regions using a grid-based clustering approach. Several other popular clustering algorithms such as Density-Joinable clustering [271], Density-Time clustering [102] and Time-Density clustering [77] have also been adapted to detect clusters in geospatial datasets, which are then considered as hotspots. Detecting hotspots by leveraging realtime location data streams has been proposed in [136, 173]. This scheme extracts the most frequent and recent hotspots of users in realtime. Zheng et al. [268] propose a variation of DBSCAN, wherein the input parameters are estimated by observing the distribution of movement density. Thomason et al. [236] proposed a gradient-based technique that combines the advantages of both; k-means and DBSCAN.

Farrahi et al. proposes a fingerprinting-based approach [68] by analyzing the temporal regularities and patterns of local transitions over time. The fingerprint is essentially a vector of visible cell towers, as described in the works of BeaconPrint [106] and the hotspots are detected based on the repeatability rate of an associated vector. Another set of techniques are based on scan-statistics, wherein a cylinder of varying radii and height is moved over the spatiotemporal space. The surface of the cylinder covers the space dimension and its height covers the time dimension. The cylinders are then sorted depending on a parameter called *p-value*, which is then used as a threshold to consider the detected regions as hotspots. Louail et al. propose a technique to extract

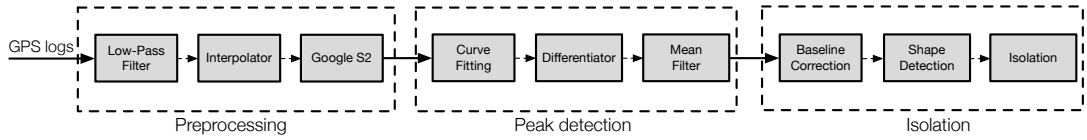


Figure 3.3 – From preprocessing to peak detection and hotspot extraction

hotspots from trajectories belonging to a group of users without relying on the commonly used spatiotemporal bounds [153]. However, they consider a group of geolocated points at a particular time t, as a hotspot, if the density of users at that location is greater then a predefined threshold δ.

The above techniques use several bounds to classify a particular region as a hotspot. Some of the parameters include, maximum distance between the collected locations, maximum and minimum time bound, cluster shape, grid size and maximum number of points per cluster. However, different users are characterized by different mobility profiles, which result in varying optimal values of these *a-priori* chosen parameters. This task of estimating the parameters and their values is challenging due to the inherent number of possible parameter combinations, different mobility behaviors, duration of the available dataset, rate at which locations are sampled and the distribution of noise in the recorded data. Often, the parameter values are derived based on logical reasoning and lack exhaustive empirical basis. This could result in a possible bias when generalizing and comparing results obtained with different techniques on different datasets. This has resulted in conflicting views regarding the significance of some parameters, such as maximum time bound between two coordinate points as seen in [170] and [236]. In this paper, we propose a solution to address the above discussed problems.

3.3 Problem Statement

Our work identifies and addresses the problem of extracting hotspots from user location trajectories without relying on any rigid parameters. Our solution to address the problem is to treat user mobility trajectories as space-time signals and process these signals to extract the hotspots. We thus have a two-fold problem statement as described below:

1. Given a user's trajectory $T = \langle ..., t_i, ... \rangle$, a sequence of spatiotemporal points, where each point t_i is a three item tuple, $\langle lat_i, lon_i, t_i \rangle$, where lat and lon is the latitude-longitude coordinate pair and t is the timestamp, translate it into a 2-dimensional continuous signal, $s(t)$, modeled as a function of changing distance with respect to time, retaining the spatial locality between the discretized points.

2. Given the spatiotemporal signal $s(t)$ of a user u, extract all the distinct hotspots and their properties, namely; the area and dwelling time.

3.4 From Trajectories to Signals

The geolocation sensors result in noise and non-uniformly sampled data points due to the hardware imperfections and network failures. Thus, the data points need to be de-noised and resampled to generate a continuous signal. We first use a standard convolution-based low-pass filter to remove the noisy components residing at high frequencies. Next, in order to get a uniformly sampled location points we apply a semivariance interpolation scheme, which fits the missing points by modeling the similarity between the points as a function of changing distance [136].

The output of the preprocessing stage is a uniformly sampled and de-noised coordinate points. In order to discretize space, we use the Google S2 library[1]. The library projects a spatial region on to the face of a cube which encloses the sphere. It performs a hierarchical decomposition of the sphere into compact cells and superimposes the spatial region/point on to the cells. It then constructs a quad-tree on each face and selects a quad-tree cell containing the projected region. Each cell is represented by exactly the same area and provides sufficient resolution for indexing the geographic features. The cells are enumerated on the Hilbert curve, which preserves the spatial locality of the points. The resulting spatiotemporal signal can be denoted as $s(t) = \langle ..., (c_i, t_i), ... \rangle$, where c_i is the cell ID and t_i the timestamp as shown in Figure 3.2.

3.5 System Design

In this section, we present our system design and implementation. The mobility signal $s(t)$ has two main components: (1) a static element which corresponds to the place having maximum user time occupancy, (2) local maxima/minima, which correlate with the user's frequently visited places. The user movements oscillate around the static element with a significant deviation, generating several local maxima/minima. This can be viewed as the presence of basic noise with a general mean which makes the hotspots identifiable . Therefore, the problem of hotspot discovery is essentially a matter of detecting the local maxima and minima (or simply 'peaks') contained in the signal. In order to determine the hotspot properties, we design our system to heuristically compute the peak start and end positions, peak height and width to estimate the hotspot visit entry and end time, the total distance travelled and the total area. The steps involved in hotspot detection are illustrated in Figure 3.3.

In order to make the peaks distinct, we perform curve fitting over the discretized location traces, $s(t)$. However, the peak shapes are not identical throughout the signal and differ according to the visited place. We therefore perform a non-linear iterative curve fitting, ensuring that the peaks do not shift or are missed. The peaks can then be detected by taking the first differential of the curves. The detection procedure operates by checking for the point of downward/upward going zero-crossing at the peak-maximum/minimum, for the peaks and the valleys. In order to make the peak detection robust, we apply a mean filter and smooth the first differential prior to checking the upward/downward-going zero-crossings. The detected peaks in turn contain two components,

[1] Google S2: https://pypi.python.org/pypi/s2sphere/

the travel path to the hotspot and the hotspot region itself. Thus, in the next step we split the ravel and the hotspot component in the peak, which requires a correct estimation of the peak shape. This step requires automatic adjustment of the baselines so as to adapt constantly to the changing user behaviors. To address this, we keep a track of the standard deviation of the points and analyze the points deviating from the moving mean, which iteratively sets the baseline and operates precisely irrespective of the peak shape. The peak shape is finally detected by taking the successive derivatives, as different peak shapes have distinct derivative shapes.

Finally, for isolating the peak components, we monitor the average rate of change of slope of the detected peaks. Once a user arrives at the hotspot, the slope changes to zero or to an infinitesimally small value, as compared to the slope of the travel component. Thus the two parts can be separated depending on the average rate of change of the slope along the maxima or minima. After the peak is isolated, the cells belonging to the hotspot are extracted and the remaining cells belong to the travel component. The representative path connecting the hotspot is constructed by selecting the cells common to both the edges. The cells, when inverted back to the location coordinates, represent the spatial locations.

3.6 Evaluation and Discussion

In order to evaluate the accuracy of the detected hotspots, we validate our results with the ground truth and perform a comparison with three popular clustering techniques commonly used for hotspot detection. As the publicly available datasets are devoid of the ground truth, we collect a dataset to validate the efficacy of our approach and to confirm our findings regarding the correlation between the spatiotemporal components and the signal elements. The mobile application provided to the users logs their latitude, longitude, timestamp, acceleration, altitude, horizontal and vertical accuracy of the GPS coordinates for a period of 11 weeks. The data points are collected at a sampling rate of 5 seconds with a granularity of resolution up to 5 meters. The ground truth is captured by periodically attesting the visited hotspots. The hotspots were selected with a clear definition: *'any place where the subject visited with an intentional purpose'*. These regions include places such as cafeterias, restaurants, bus/train/metro stops, sports arenas, bookstores, office and work places and excursions. The ground-truth evaluation was performed by computing the precision, recall and the accuracy values.

We configure the Google S2 library to map each coordinate pair to a cell of dimension $38m^2$. It could be argued that the cell size involves a arbitrarily chosen parameter in the process. However, real-world hotspots typically spread over areas larger than $38m^2$. Furthermore, this choice is motivated by the localization accuracy of a typical GPS sensor and the performance complexity involved when subdividing the cells to the leaf level.

We consider Density Joinable Cluster (DJ Cluster) [271], Density Time Cluster (DT Cluster) [102] and ZOI Detect [136]. DJ Cluster computes hotspots, based on the number of points within a certain radius and merges these clusters if they share at least one point in common. Furthermore,

Clustering algorithm	Parameters
DJ Cluster	Min_{speed}: 0.4 (km/hour) / $Cluster_{radius}$: 60.0 (meters) / Min_{points}: 10
DT Cluster	Max_{dist}: 60.0 (meters) / Min_{time}: 900 (seconds)
ZOI Detect	Max_{dist}: 60.0 (meters) / Min_{time}: 900 (seconds) / Min_{visit}: 6

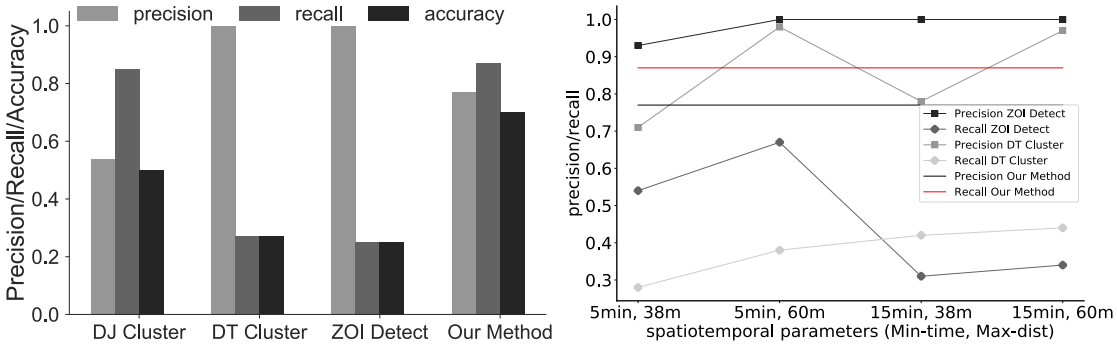
Table 3.1 – Clustering algorithms and their default parameter values

the points are also clustered if they satisfy the min_{speed} bound. DT Cluster aggregates points lying within a predetermined spatiotemporal bound. These clusters are treated as valid hotspots. ZOI Detect follows a similar strategy as DT cluster but involves an additional parameter min_{visit} as a threshold and merges the clusters upon intersection. The parameters selected by these techniques and their values are shown in Table 5.3. These values selected in published works are either based on dataset trends [136] or on user mobility behavior [271].

We see that DT Cluster and ZOI Detect have a high precision and low recall and accuracy rate as seen in Figure 3.4a. This indicates that, these techniques discover a large number of hotspots that are not contained in the true hotspot set. This is clearly due to the spatiotemporal bounds being too rigid, which results in considering arbitrary clusters as valid hotspots. DJ cluster, however, has higher recall and low precision. Here, we see that the Min_{speed} eliminates the occurrences of false negatives, whereas, the Min_{points} creates high false positives. Increasing the Min_{points} can address such occurrences, as it requires a higher density of points, thus creating only valid hotspots. In case of our method, we have a few false positives due to the high sensitivity for the slop change and only three false negatives. Closely examining the false-positives reveal that, these regions are visits without any purpose attached, such as delays at metro and bus stops. This creates additional hotspots which are not based on user intent. The false negatives are the stops where the user does not have to wait. These cases occur due to planned time synchronization by the user between the transportation mode switches, resulting in a constant average slope.

To better understand the parameter influence, we consider four different parameter sets for the values of Min_{time} and Max_{dist} as seen in Figure 3.4b. We see that the parameter Min_{visits} always correctly classifies a region as a hotspot, thus leading to high precision rates. We can also see that larger values of Max_{dist} results in higher precision and recall in DT Cluster. Max_{dist}, thus plays a vital role in determining precision, compared to Min_{time} parameter in the considered dataset. These results highlight the importance of selecting the parameter space which is challenging to determine *a–priori*.

In general, we observe that DJ Cluster and DT Cluster detect a significantly high number of hotspots in both the cases. In case of DJ Cluster, we find that the parameter Min_{points} creates a large number of hotspots. However, we argue that if the sampling rate of the dataset is high, Min_{speed} could play an important role in further increasing the clusters. Whereas, in DT Cluster Min_{time} bound parameter results in a higher frequency of visit separations, increasing the total number of hotspots. These factors contribute to a higher number of hotspots, which



(a) Ground truth validation.

(b) Parameters impact on accuracy.

Figure 3.4 – Performance evaluation.

is not typical for an average user. We observe that the number of hotspots discovered by ZOI Detect [136, 173] is lower than DT and DJ Cluster. This is due to the merging of individual clusters upon intersection, in addition to extracting the most frequent clusters governed by the Min_{visit} parameter. In general, if the parameters satisfy cluster merging, multiple clusters merge and form a large hotspot; hotspot division occurs if this bound is missed by even an infinitesimal small value. This results in the fluctuation of the number of hotspots solely due to the parameters.

The hotspot area in our case corresponds to $38m^2 \times cellnumbers$. We find that, this results in a significantly smaller areas compared to the clustering techniques and overlaps with the ground truth area. This is due to the set of cells in the hotspot, which corresponds to a cell where a user was actually present, and which is smaller than the actual area of the hotspot.

3.7 Conclusion

In this paper, we have proposed a technique to detect hotspots from user trajectories without relying on any *a-priori* assumptions. We have depicted the bias resulting due to the stringent parameter bounds while extracting user hotspots. We have also depicted the problems arising from such bounds that are based on non-empirical calculations and extended to operate on some other datasets and on users having different mobility behaviors. We have addressed this problem by treating user movements as spatiotemporal signals, effectively converting it to a peak-detection problem by using signal-processing algorithms. The evaluation results show that our approach outperforms the popular clustering techniques used for hotspot detection. We have also validated our results with the ground truth and achieved precision and recall rates exceeding 80%.

Predicting Human-Mobility Part II

4 MobiDict – A Mobility Prediction System Leveraging Realtime Location Data Streams

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 184 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.

Keywords: Realtime mobility prediction; Mobility behaviour; Location-based services.

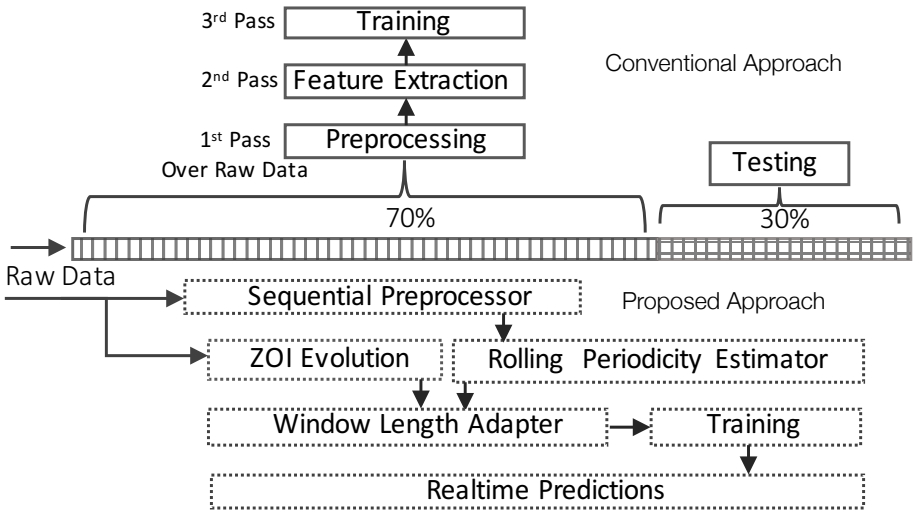


Figure 4.1 – Traditional Prediction Systems *vs.* MOBIDICT. The process on the top depicts the traditional mobility prediction approach, while the process chain shown at the bottom gives an overview of our technique.

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 [7, 65, 140] 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 [138, 266]. 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

[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

on the hindsight only results in statistical prediction models without truly capturing the inherent nature associated with user movements [57].

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 [77] and identifying the user [55]. 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 [84].

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 4.1 presents our system model and introduces formal definitions and notations used in the paper. Section 4.2 describes our approaches of

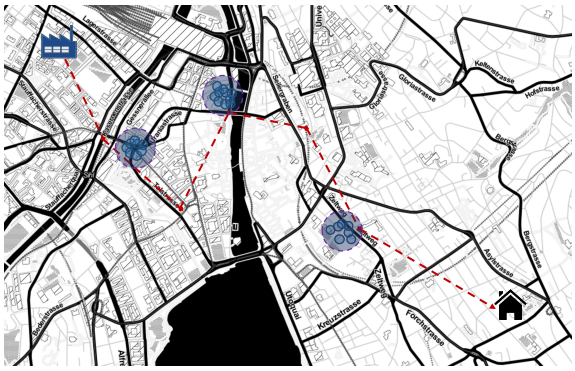


Figure 4.2 – ZOI Construction from Cluster of Location Points.

quantifying user behaviour. In Section 4.3, we present the different prediction techniques used in the MOBIDICT system. Section 4.3 then 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 4.4, based on mobility traces collected on our campus by Nokia Research, from 184 users, between October 2009 and March 2011. Finally, Section 4.5 discusses research efforts similar to ours and Section 4.6 concludes the paper by sketching future research directions around MOBIDICT.

4.2 Mobility behaviors

4.2.1 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.

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. A subset $l \subseteq L$ becomes a cluster iff the following conditions in Equations 4.1, 4.2 and 4.3 are satisfied[4]:

[4]This clustering process is inspired by a technique called *DT cluster* and presented in [77].

$$\forall loc_i, loc_{i+1} \in l : \\ distance(centroid(loc_1, \cdots, loc_i), loc_{i+1}) \leqslant \Delta d_{max} \tag{4.1}$$

$$loc_n.t - loc_1.t \geqslant \Delta t_{min} \tag{4.2}$$

$$\nexists l' \neq l : l \subset l' \tag{4.3}$$

A cluster is a 4-item tuple $c = (\phi, \lambda, \Delta r, l)$, where $\phi \in \mathbb{R}$, $\lambda \in \mathbb{R}$, $\Delta r \in \mathbb{R}$ and l are a latitude, a longitude, a radius in meters and a subset of locations respectively. Here, $\Delta d_{max} > 0$ and $\Delta r > 0$. The mean of all ϕ and λ of the locations contained in the subset l is the centroid (ϕ, λ) of the cluster, which is designated as $c.centroid$. We consider the set C, containing all user's clusters, where $C = \{c_1, c_2, \ldots\}$. Equations 1, 2 and 3 do not guarantee disjointness of clusters which 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:

$$distance(c_i.centroid, c_j.centroid) \\ - \ (c_i.\Delta r + c_j.\Delta r) \ < \ 0 \tag{4.4}$$

A cluster group is a 4-item tuple $g = (\phi, \lambda, \Delta r, \{c_1, c_2, ...\})$, where $\phi \in \mathbb{R}$, $\lambda \in \mathbb{R}$, $\Delta r \in \mathbb{R}$, $\{c_1, c_2, \ldots\} \in C$ 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 $G = \{g_1, g_2, \ldots\}$.

ZOI. A ZOI is a frequently and recently visited zone by a user in everyday life. The 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, while $minVNB \in \mathbb{N}$ is a variable representing the minimum number of visits. Then, $size(g)$ is a function which computes and returns the number of clusters of the cluster group g, $meanVNB(G)$ 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 which returns the duration between the current date and the last visited date of the cluster group contained in G. $meanVNB(G)$ returns values which dynamically change over time according to the mobility behaviour of the user due to the realtime nature of the process. $minVNB$ is equal to the value returned by $meanVNB(G)$ 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 in Equation 4.5 and 4.6 are satisfied:

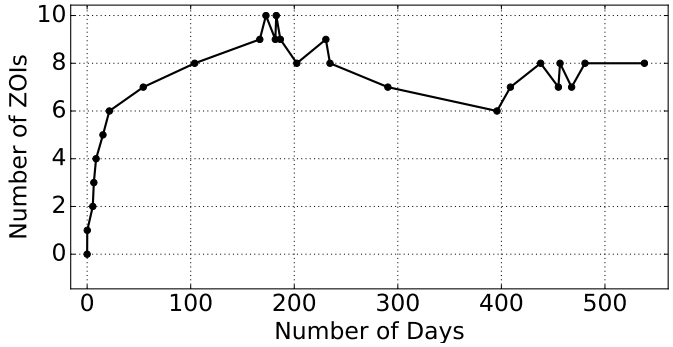


Figure 4.3 – ZOI Evolution Over Time.

$$
\begin{aligned}
& size(g) \geqslant minVNB \\
\vee \;\; & minVNB = meanVNB(G) \\
\wedge \;\; & minVNB <= visitThreshold \qquad (4.5) \\
& timeDuration(G) <= maxTimeDuration \qquad (4.6)
\end{aligned}
$$

A ZOI z is formally, a four item tuple $z = (\phi, \lambda, \Delta r, g)$, where $\phi \in \mathbb{R}$, $\lambda \in \mathbb{R}$, $\Delta r \in \mathbb{R}$ 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.

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.

4.2.2 Periodicity of Movement

Human mobility is characterised by a high degree of periodicity, contrary to the popular assumption that the mobility patterns are highly stochastic. Detecting these periodic behaviours can assist to generate quick predictions, evading the complex training procedure. However, one of the challenges is to identify periods which 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.

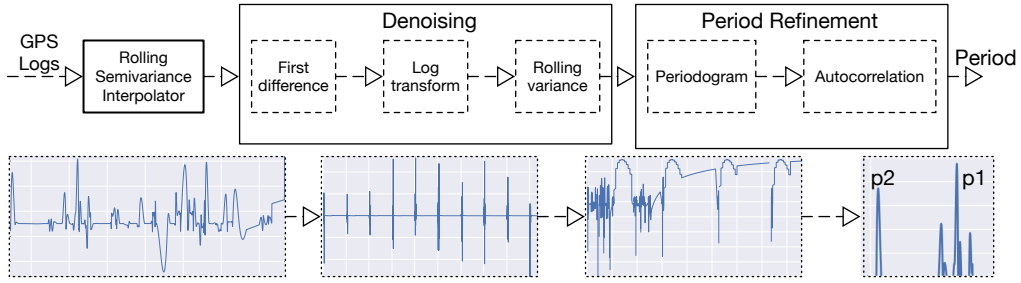


Figure 4.4 – Realtime Periodicity Estimation Chain.

We describe the steps involved to accurately detect the movement periodicity.

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.

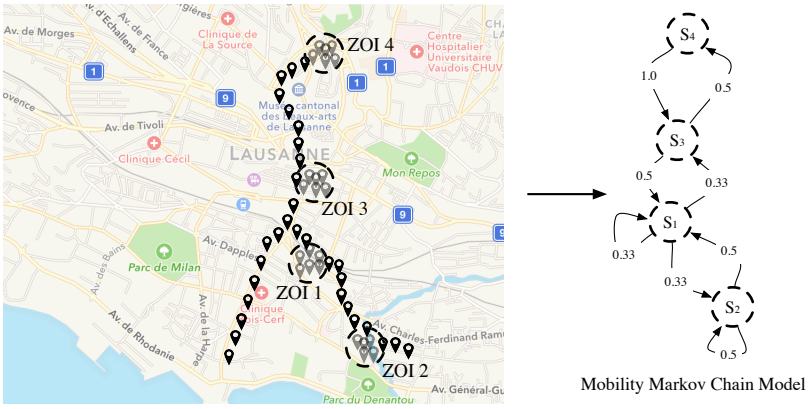


Figure 4.5 – From User's ZOIs to MMC Model.

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 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.

4.3 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 4.2, 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 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

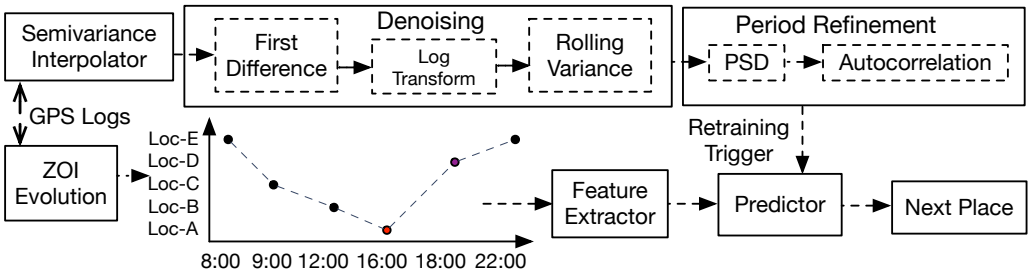


Figure 4.6 – Realtime Periodicity Aware Training Process.

behaviours to produce next place predictions. The common goal being, formulation of a robust predictive system on streaming location data.

4.3.1 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 freshness 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 4.8 as below:

$$t_{next} = date_c + u_{mean} + \frac{u_{mean}}{2} \tag{4.8}$$

where $date_c$ is the current date of the system. 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.

4.3.2 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, which in our case are the user movements and the variations, link to changing periodicities. We empirically determine that the model accuracy is affected for an autocorrelation index change of 0.2 and greater. This severs 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 4.2, 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 4.9 as below:

$$P(\frac{C_k}{N}) = \left|\left| \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x(n).e^{\frac{-j.2\pi.c.n}{N}} \right|\right|^2, c = 0, 1 ... \frac{N-1}{2} \tag{4.9}$$

where, C_k are the strength encodings at a given frequency k, $x(n)$ are the spectral coefficients associated with the sinusoids.

We track the periodicity continually, following the above approach. Regarding the training phase, as depicted in Figure 4.6, 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 4.2.2. 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.

4.4 Experimental Evaluation

In this section, we demonstrate the experimental results of our approach based on the Nokia data set [138] consisting of mobility traces, collected from 184 users in Switzerland from October 2009 to March 2011. The participants consisted of university students and professionals with a

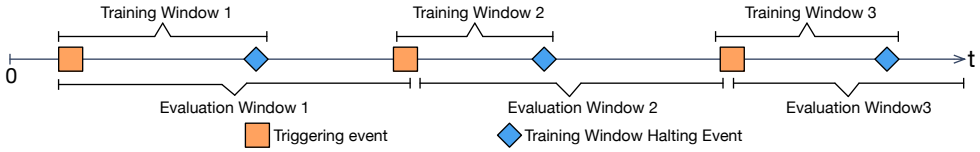


Figure 4.7 – Realtime Evaluation Scheme.

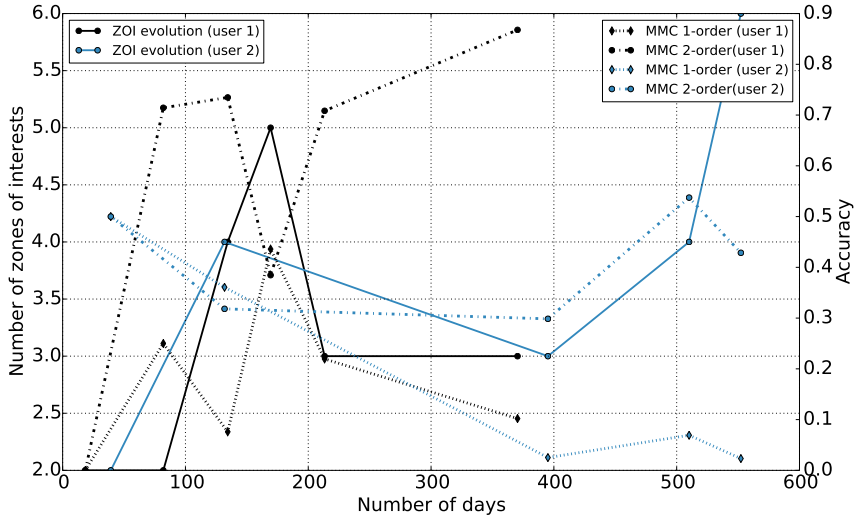


Figure 4.8 – Evolution of ZOIs and Prediction Accuracy Over Time of 2 Users According to 1-order and 2-order MMC.

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.

4.4.1 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 points of interest with respect to space and time. The $visitThreshold$ parameter is set to 6 visits and the $maxTimeDuration$ of three 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.

4.4.2 Real-time Evaluation Scheme

Figure 4.7 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 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 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 4.2.2. 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.

4.4.3 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 4.8 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, contained 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 unaffected 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

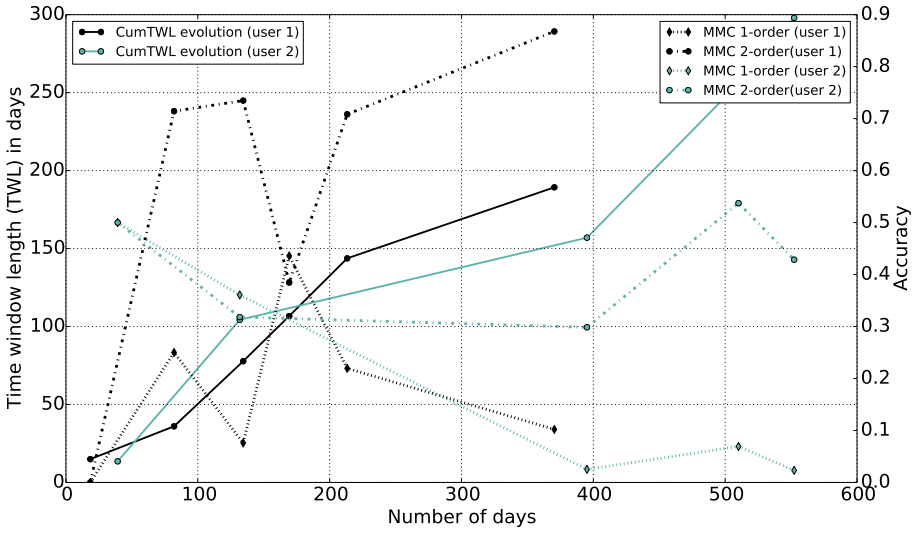


Figure 4.9 – Evolution of Cumulative Time Window Length and Prediction Accuracy Over Time of 2 users According to 1-order and 2-order MMC.

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

probabilities contained within them.

In Figure 4.9, 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 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 ore 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 4.10, we see a clear correlation between the periodicity

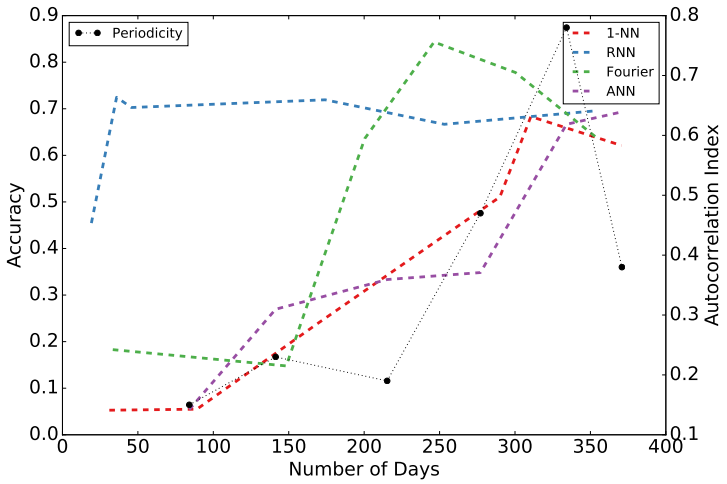


Figure 4.10 – Variation of accuracy with time and the movement periodicity.

	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.

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 into account before making a prediction, effectively combining, high level direction with low level modelling that results in high accuracy, maintained almost stable with time. On the other hand, the classification and neural network based approach weighs 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.

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 4.11. As we see, the accuracies are in general lower than the baseline accuracies however, in most of the cases represent satisfactory accuracy level above 50%. After update 3, the accuracies of 2-MMC, ANN, RNN and Fourier based predictors are often higher as compared to the baselines. Regarding

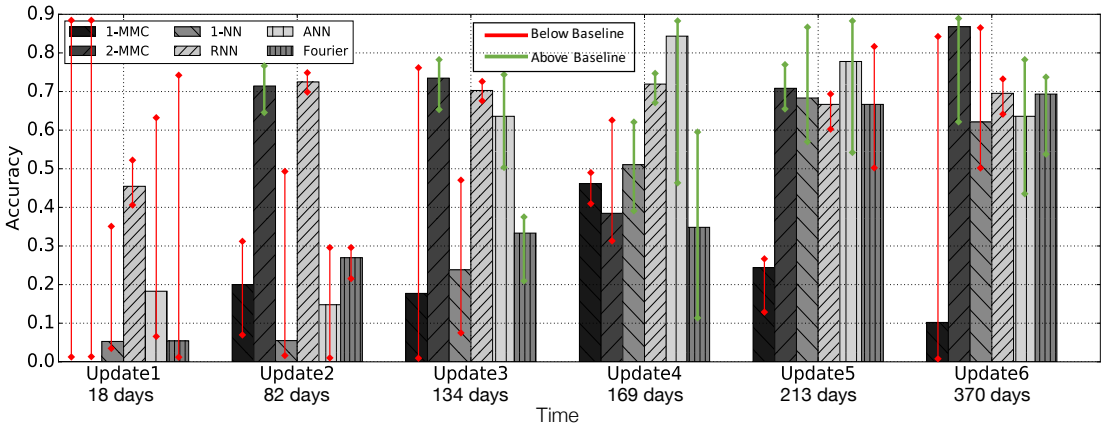


Figure 4.11 – Comparison of MOBIDICT accuracy for individual predictors against the baseline accuracies.

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, involves 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 [145]. 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. $n = t_{n1} + t_{n2} \ldots + t_{nN}$, since we account for the user behaviour, which is constant for some time periods the total number of data-points will be lower than D, thereby having a lower complexity. The same goes for the training time.

4.5 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 [41, 169]. The above approaches rely on clustering visits to form a visit region. We reviewed several location based clustering works [12, 60, 77, 118, 259]. 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 [42, 100]) 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) [150], Levy-flight nature [204] and daily activity analysis and dissimilarities within them as presented in[115]. Although the above models aid in predicting human mobility, the mobility models are 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. [199] 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 [34, 147]. 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 [57, 57, 87, 140] 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 [143, 193]. 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 [7, 206, 208]. [56] 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 [139]. 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 the mobility behaviour.

4.6 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 LBS 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 RRN 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 to quantify 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.

5 Capstone: Mobility Modeling on Smartphones to Achieve Privacy by Design

Abstract

Sharing location traces with context-aware service providers has privacy implications. Location-privacy preserving mechanisms, such as obfuscation, anonymization and cryptographic primitives, have been shown to have impractical utility/privacy tradeoff. Another solution for enhancing user privacy is to minimize data sharing by executing the tasks conventionally carried out at the service providers' end on the users' smartphones. Although the data volume shared with the untrusted entities is significantly reduced, executing computationally demanding server-side tasks on resource-constrained smartphones is often impracticable. To this end, we propose a novel perspective on lowering the computational complexity by treating spatiotemporal trajectories as space-time signals. Lowering the data dimensionality facilitates offloading the computational tasks onto the digital-signal processors and the usage of the non-blocking signal-processing pipelines. While focusing on the task of user mobility modeling, we achieve the following results in comparison to the state of the art techniques: (i) mobility models with precision and recall greater than 80%, (ii) reduction in computational complexity by a factor of 2.5, and (iii) reduction in power consumption by a factor of 0.5. Furthermore, our technique does not rely on users' behavioral parameters that usually result in privacy-leakage and conclusive bias in the existing techniques. Using three real-world mobility datasets, we demonstrate that our technique addresses these weaknesses while formulating accurate user mobility models.

Keywords: Location privacy; Mobility modeling; Signal processing; Behavioral parameters; Mobility dynamics.

5.1 Introduction

A large amount of geolocation data is being ubiquitously collected, due to the advent of location-based services (LBS) and the pervasive nature of smartphones. The personally identifiable information (PII) of users extracted from this data is crucial from the service providers perspective for offering personalized services. The accumulated data is used to constitute user specific, as well as collective mobility models, that encapsulate mobility behaviors. Such models are used for a variety of applications such as location-based advertisements, traffic management and urban planning. However, when users share their location traces with third-party service providers, it exposes them to several privacy risks [129]. Simple heuristics can be applied by curious adversaries to derive PII for blackmailing or stalking purposes [77]. Recent regulations such as the EU General Data Protection Regulation (GDPR)[1], however, have placed stringent data acquisition and retention policies. Article 25 (*data protection by design and by default*) lays out strict clauses for service providers, regarding the localization of computations and storage at the user's end whenever possible.[2] A recent report claims that about 55% of mobile applications do not currently comply with GDPR [2]. Therefore, user privacy consideration will be a key factor to determine the success and adoption of context-aware services in the coming years.

Several solutions have been proposed to address this issue in the context of LBS, including spatial cloaking [95], k-anonymity [78] and cryptographic primitives [73]. Such techniques account for the optimization of the privacy/utility trade-off, where utility is often quantified in terms of the accuracy of the disclosed location traces [217]. However, such measures are still inefficient in deriving user mobility models with practically usable tradeoff [260]. Another category of solutions investigate data concealment, for example, Laplace perturbation, which encodes the trajectories with their Fourier transform coefficients [203]. However, data concealing and aggregation techniques are also exploitable due to the regularity and uniqueness of human mobility as shown by Xu et al. [251].

Orthogonal to the above solutions, a drastic privacy-preserving approach is to deploy the learning models directly on user's smartphones to train on their data without having to send it to the cloud. An example of this approach is Google federated learning [160]. This model reverses the client/server relationship by enforcing the service providers to query for the required data from the model present on the user smartphone. Finally, the query response can be processed in a trusted computing environment as illustrated in our previous work [131]. Judicious scheduling in such systems ensures that learning occurs only when the device is completely idle [160]. Thus, computational complexity and power consumption are the main concerns in making such a system practical.

In this paper, we adopt this type of approach consisting in restricting the mobility modeling task on the user's smartphone. Our approach in making such a system feasible is to treat spatiotemporal trajectories as signals. To this end, we leverage the following key properties of spatiotemporal

[1] GDPR: www.eugdpr.org

[2] Article 25 GDPR: gdpr-info.eu/art-25-gdpr

signals: (i) lower data magnitudes due to the reduced dimensionality, which is a direct application of Johnson–Lindenstrauss lemma (low-distortion embeddings of points from high-dimensional into low-dimensional Euclidean space), (ii) compressed representation, as the information is concentrated in a few spectral coefficients, and (iii) the ability to offload computationally intensive tasks to the digital-signal processors (DSPs) present in many smartphones.

Traditionally, a mobility model is represented in terms of a directed graph, where the nodes correspond to the user's regions of interest (ROIs) and the edges correspond to the representative paths between the ROIs, weighted by the respective transition probabilities [181]. Mobility modeling task is therefore composed of computing the ROIs, representative paths and the transition probabilities as depicted in Figure 5.1. The current techniques used to perform the above tasks rely on an individual's behavioral parameters representing their mobility dynamics. However, these parameters act as side channels that can be used by malicious adversaries to infer an extended view of the whereabouts of a user appearing in an anonymous trajectory [188]. Commonly used parameters such as minimum time period and maximum distance between two location coordinates can be used to de-anonymize aggregated spatiotemporal data [251]. We also show that reliance on these parameters result in conclusive bias and unfair comparisons of the efficacy of different techniques. We eliminate the dependence on the rigid parameter space and implement the proposed approach on a DSP chip to practically demonstrate the advantages. Our contributions in this context are as follows:

- We present **Capstone**, a technique to construct a user mobility model using space-time signals. We divide our contributions in three distinct parts: (i) translating the noisy and non-uniformly sampled GPS trajectories into a continuous space-time signal, (ii) establishing a systematic relationship between the fundamental components of human mobility and the temporal-spectral units of the space-time signal, and (iii) a signal processing pipeline to extract user mobility model.
- We highlight the parameter curse present in the current techniques resulting in a strong conclusive bias and privacy leakage through experimental evaluation. We demonstrate the effectiveness of **Capstone** in addressing such drawbacks in addition to its suitability across a large variety of mobility datasets and disparate user mobility behaviors.
- Finally, by using three real-world mobility datasets, we show that **Capstone** achieves higher precision, lower complexity and reduction in power consumption as compared to the existing techniques, demonstrating its suitability to function on smartphones.

We describe the privacy model and the problem statement in Section 5.2 and Section 5.3. The three key contributions of **Capstone** are presented in Section 5.4, 5.5 and 5.6. The drawbacks associated with the behavioral parameters are presented in Section 5.7 followed by the evaluation results and discussion in Section 5.8. Finally, we present the related work in Section 5.9 and conclude our paper in Section 5.10.

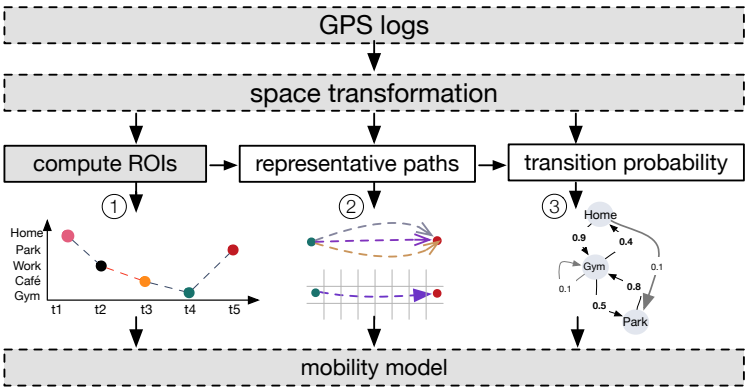


Figure 5.1 – Mobility modeling tasks: (1) computing ROIs, (2) estimating representative trajectories, and (3) computing transition probabilities.

5.2 Privacy and Attack Model

Our work focuses on privacy concerns associated with user mobility data, aggregated at the LBS provider's end. We adopt the privacy by design principle, which demands inclusion of data protection convention from the onset of the system design. More specifically, we base our model to comply with the EU data protection regulation.[3]

Location-based services are typically divided into two types: continuous and sporadic, depending on the exposure of user locations [217]. In our case, we consider a continuous location exposure-based service, where the provider is assumed to be a passive adversary (i.e. honest-but-curious). We focus on converting such a continuous case, wherein the adversary can track users over time and space into a sporadic case, where a user explicitly grants location access to the adversary, only at discrete time instances. Thus, the adversary will know the geographical distribution of users over a considered region, but not their exhaustive movements. Therefore, by constructing the mobility model locally, our approach adopts a privacy by design principle, i.e., only sharing a summary/sketch of movements that is sufficient to use the service. We emphasize here that, we do not perform the space transformation (dimensionality reduction) as a means to encode the data in a format which directly preserves privacy. Instead, we simply leverage it to lower the computational complexity and power consumption. We also do not consider man-in-the-middle attacks or code injection attacks.

5.3 Problem Statement

The key idea behind our work is: computation on spatiotemporal data using signal processing has inherent complexity and privacy benefits. To this end, the central problem is construction of mobility models using the space-time signals in order to process and store user data locally. Hereafter, we split this main problem statement into three sub-problems for clarity and set forth

[3] Art. 23: www.privacy-regulation.eu/en/article-23-restrictions-GDPR.htm

the requirements and challenges associated with each problem.

Problem 1: Mobility Signal Generation. Given a trajectory T_u of an individual u, a temporally ordered sequence of tuples, such that, $T_u = \langle (l_1,t_1),(l_2,t_2)...(l_n,t_n)\rangle$, where $l_i = (lat_i, lon_i)$, the latitude-longitude coordinate pair and t, the timestamp such that $t_{i+1} > t_i$, translate T_u into a 2-D signal $S_u(t)$, modeled as a function of changing distance with respect to time.

Requirements and Challenges. (i) Constructing a continuous graph from the noisy and non-uniformly sampled location trajectories, (ii) preserving all the key knowledge contained in the trajectory samples, and (iii) retaining the spatial locality between the discretized points.

Problem 2: Signal Interpretation. Given a user's spatiotemporal signal $S_u(t)$, interpret and model the distinct signal elements in the temporal and spectral domain, i.e. local maxima/minima, rising/falling edges, static signal component, candidate frequencies, spectral coefficients and harmonics, with respect to human mobility behaviors.

Requirement and Challenge. In order to facilitate inter-domain switching, attach and validate a semantic meaning to each of the above signal components.

Problem 3: Mobility Modeling. Given the signal $S_u(t)$ and the valid interpretation of each element, construct the user's mobility model in terms of a graph $G_u(ROI, Tr)$, where $ROI = \{ROI_1, ROI_2...ROI_n\}$ is the set of all the regions of interests belonging to u and $Tr = \{(R_{12}, p_{12}),(R_{23}, p_{23})...\}$ is the set of tuples where R_{ij} and p_{ij} denotes the representative path and the transition probability from ROI_i to ROI_j.

Requirement and Challenge. To extract all the distinct ROIs and the transitions without relying on any behavioral parameters to eliminate conclusive bias, privacy leakage and facilitate applicability across diverse datasets.

5.4 From Trajectories to Signals

In this section we address **Problem 1**, i.e., translating the noisy GPS trajectories into a continuous signal, that can be processed.

5.4.1 Preprocessing

The imperfections in the geolocation sensors and network failures often result in noisy and non-uniformly sampled trajectory points, thus hindering the process of generating a smooth and continuous signal. Therefore, to make the scheme robust, we first filter and de-noise the incoming location traces. Since the noise is not symmetrically distributed, applying averaging and median techniques does not solve this problem. Additionally, the noisy components reside at high

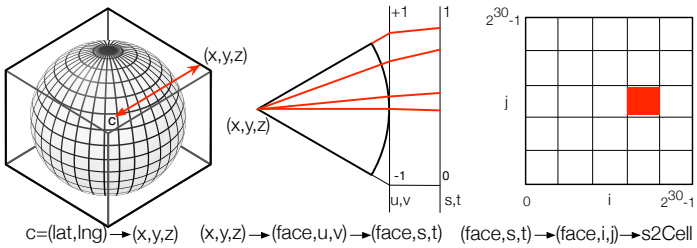


Figure 5.2 – Projecting a coordinate pair onto the grid with Google S2.

frequencies and do not contain any sharp pulses, hence we apply a standard convolution-based low-pass filter. Next, we employ semivariance interpolation to obtain uniformly sampled location points. This interpolation scheme uses a moving average construction and conceals the incoming data about the spatial variance of the past trajectory points[118].

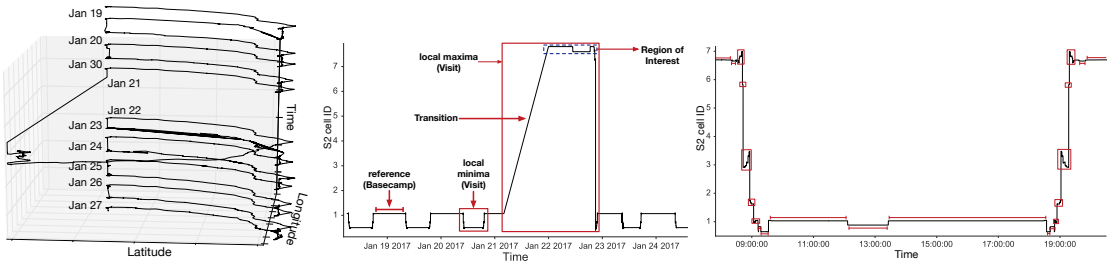
5.4.2 Space Discretization

The filtering and interpolation results in uniformly sampled de-noised trajectories. The process eliminates any bursty coordinates, however, some amount of white noise can still be present. The next step is space discretization, for which we rely on the Google S2 Library.[4] It performs a hierarchical decomposition of the earth sphere into compact cells and superimposes a spatial region/point on to one of the cells. Each cell is represented by exactly the same area and provides sufficient resolution for indexing the geographic features. In short, the library operates by first enclosing the earth in a cube. It then projects the spatial region onto the face of the cube, builds a quad-tree on each face and selects the quad-tree cell that contains the projection of that region. In the first step, the point $c = (lat, lon)$ in Figure 5.2 is transformed into (x,y,z) after projecting it on the cube. As the cells on the cube have different sizes when mapped back to the sphere, a non-linear transform is performed, i.e., (u,v) is transformed to (s,t) before discretizing the point by superimposing it on the grid and retrieving the respective *Cell ID*. The cells are then enumerated on a Hilbert curve, preserving the spatial locality of the points [172]. The 64 bit *Cell ID* has 3 bits that encode the cube's face and the remaining 61 bits encode the position of this cell along he Hilbert curve. The resulting spatiotemporal signal can be denoted as $S(t) = \langle (c_1,t_1),(c_2,t_2)...(c_n,t_n)\rangle$, where c_i is the *Cell ID* and t_i the timestamp. The *Cell IDs* ensure that, each of the discrete point connects to the other to obtain a continuous graph, thus preserving the spatial locality between the individual points. The 3-D trajectories, shown in Figure 5.3a thus translate to 2-D space-time signals depicted in Figure 5.3b.

5.5 Signal Interpretation

In this section, we address **Problem 2**, i.e., interpreting the distinct signal elements in time and frequency domain. The theoretical constructs proposed in this section are validated using three

[4]Google S2: https://s2geometry.io/



(a) Ten-day trajectory (b) Space-time signal (c) one day's space-time signal

Figure 5.3 – Visualizing the user's movements as a trajectory and as space-time signal. The red rectangles and lines denote the ROIs.

mobility datasets (discribed in Section 5.8).

5.5.1 Temporal Domain

A periodic signal $S(t)$ is typically represented as $S(t+n.T)$ for all time t and the periodic component T, where $n.T$ is the period of the signal. Although, human mobility is characterized by distinct regularity [226], the space-time signal $S(t)$ is not perfectly periodic. As the mobility patterns do not have the same mean periods, $S(t)$ can be considered as almost periodic [239] and represented as $S(t) = S(t+n.T(t))$, where the fundamental period T, can change over time. It has two main components: (i) a static element and, (ii) rising/falling edges as seen in Figure 5.3b. In this work, the static element is treated as a *reference*, that corresponds to the user's *basecamp*. We refer to the *basecamp* as a place having maximum user time occupancy (typically the home or work place). It is represented in the time domain as $Mo(S(t))$, i.e., simply the *Mode* value of the signal, that correlates with the most frequent location in the user's trajectory.

This reference signal is accompanied with local maxima and minima. The user movements revolve around this reference with an element of deviation. This is viewed as the presence of basic noise with a general mean. A user's ROI visit, thus corresponds to the local maxima/minima present in the signal and their amplitude correlates to the distance from the *basecamp* (or another ROI). A set of distinct ROIs can thus be obtained by selecting only the maxima/minima with distinct amplitudes. The maxima/minima significantly deviate from the noise and the reference element, and therefore are distinctly identifiable. A ROI visit can be expressed in terms of the local maxima at time t_m as given in Equation 5.1. Here, $t_m - t_e$ and $t_m + t_e$ correspond to ROI visit outset and end times.

$$Visit = \bigcup_{t_e=-e}^{e} \{S(t_m + t_e) \mid t_m, t_e \in t, S(t_m \pm t_e) > S(t)\} \tag{5.1}$$

Each maxima/minima can be decomposed into its constituents: (i) rising/falling edge, and (ii) local static element. It can be represented as $R_{edge} + LS_{static} + F_{edge}$, where R_{edge} and F_{edge}, the rising and the falling edges correspond to the transition component and LS_{static}, the local static

element maps to the user's ROI. A ROI visit can therefore be represented as $Visit = Tr + ROI$. With respect to the spatial region, Tr (transition) contains the knowledge of the trajectory traversed between the ROIs. Whereas, the *ROI* holds the information regarding the spatial extent of the region of interest.

5.5.2 Frequency Domain

The spatiotemporal signal is inherently non-stationary, i.e., the frequency content of the signal changes over time. Therefore, it needs to be processed as a short-term signal where it can be assumed as quasi-stationary. Applying autocorrelation and power spectral density (PSD) analysis of such a signal extracts the candidate visitation periods. As this signal can be viewed as a superposition of multiple periodic elements, each one corresponds to the visitation cycle associated with a distinct ROI. Applying a discrete cosine transform (DCT) further isolates the signal into fine grained constituents that correspond to the different movement patterns of a user. The periodicity associated with a single ROI visit corresponds to the frequency of one complex sinusoid and is represented as Equation 5.2.

$$periodicity = \sum_{n=0}^{N} C_{i,n}.e^{ji\theta_n} \tag{5.2}$$

In this equation, $C_{i,n}$ is the spectral coefficient that can change over time with respect to a time-dependent parameter θ_n bounded by N and associated with a single frequency component i. Therefore, the complete set of periodicities associated with all the ROIs of a user can be derived by using Equation 5.3. Here, the number of frequencies are restricted to an unknown but finite number P, the fundamental frequency θ_n and the spectral coefficients C_p, which can drift with time.

$$ROI_{periodicity} = \bigcup_{p=0}^{P} \sum_{n=0}^{N} C_{p,n}.e^{jp\theta_n} \tag{5.3}$$

An important property of DCT, which makes processing large magnitudes of trajectories viable on smartphones is its high degree of compaction. A DCT can provide a representation of the original signal by using a relatively small set of coefficients [198]. This is a highly desirable property when it comes to computing on resource constrained platforms, as it reduces the data storage requirements by storing only the coefficients that contain significant amounts of energy. These coefficients retain the key signal information in a compressed state, such as the visitation periodicity and distance associated with the transitions.

In order to interpret the frequency domain components, we performed a modified discrete cosine transform (MDCT) over a fixed-size window (24 hours). We correlate the scaled coefficients with the time domain signal and deduce that the low frequencies best describe the cycles and periods

in the mobility traces. On the other hand, the high frequencies mostly contain noise, and can be eliminated. The abscissa represents the frequency of visitation (dominant periods), whereas the ordinate corresponds to the distance. Furthermore, we also infer that the lower frequencies reside at higher distances and depict a periodic behavior and the harmonics represent the time-shifted versions of the ROI visits.

5.6 Mobility Modeling

In this Section, we address **Problem 3**, i.e., mobility modeling and describe **Capstone's** system design and implementation. A mobility model consists of three main components: (i) ROIs, (ii) representative paths, and (iii) transition probabilities. Once all the ROIs are computed, the process of obtaining the representative paths and the probabilities is detailed in our previous work [39, 136]. Constructing a representative path is essentially a procedure to efficiently extract the set of *Cell IDs* that best describes the trajectory between the two ROIs. The transition probabilities can be computed by using mobility Markov chains [74].

Following the discussion in Section 5.5, it is evident that the problem of constructing the mobility model, is essentially detecting the local maxima and minima (henceforth referred to as a 'peak') contained in the signal. This step is followed by isolating a peak into its constituents i.e., the set of cells associated with the ROI and the set of cells constituting the representative transition path. The latter is provided as an input to our technique [39] that extracts the path from this set. The system should be able to heuristically compute the following components upon which we base our design:

1. peak start and end positions, to determine the ROI visit entry and end times;
2. peak height, through which the distance travelled from the basecamp is calculated;
3. peak width, to compute the total area and time spent at a given location;
4. peak separation into travel time and stay time.

5.6.1 Visit Detection and Isolation

This section focuses on obtaining the individual visits to a ROI and isolating them into the constituent components. Following the preprocessing steps to obtain the de-noised signal described in Section 5.4 and depicted in Figure 5.4, we perform two more operations to make the peaks in the signals distinct. Note that, a visit is synonymous to a peak (upward or downward going) in the signal.

The first three steps involving the low-pass filtering, interpolation and the generation of a discrete signal are already discussed in Section 5.4. The step after discretizing the location traces is curve fitting. As the peak shapes are not identical throughout the signal, a predefined, shape-dependent

Figure 5.4 – Preprocessing steps.

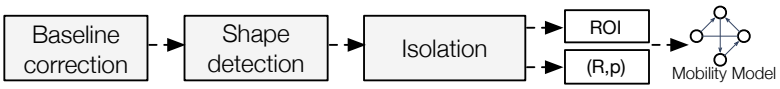


Figure 5.5 – Visit-detection and mobility modeling procedure.

curve fitting could not be used to fit a curve to the cell IDs. We observed that the peak shapes differ according to the visited place. For example, the movement is restricted to a relatively small area in a work/home/gym place, whereas it is dispersed over a larger area in a shopping mall or in a park. This does not affect the transition component of the peak, rather the cap (local static signal) of the peak representing the actual region. The peak shapes observed in the considered datasets can be represented in terms of convolution functions, i.e., $rectangular * Gaussian$ or $rectangular * Lorentzian$ or $triangular * Gaussian$ function. We do not make any assumptions regarding the shapes and perform a non-linear iterative-curve fitting with selectable peak-shape models. The curve fitting is applied to the whole signal, ensuring that the actual peak parameters are not distorted during the subsequent processing steps. This step is crucial as it facilitates measurement of the slope to isolate the peak into its components (ROI+representative path). Furthermore, it is also necessary to accurately estimate the ROI area and the visit duration. The iterative fitting ensures that the peaks do not shift or are missed, which might result in inaccurate cell ID retrieval of user movements.

An elementary technique for peak detection is to take the first differential of the points whose peaks have a downward going zero-crossing at the peak maximum. However, the peaks can also lie below the basecamp that can be viewed as valleys. In this case, the upward-going zero-crossings are checked, and the local minima is accounted for, instead of the maxima. The presence of white noise might result in false positives, leading to failure in obtaining the correct ROIs and estimating accurately the repeated visits. It can also alter the derived features of the peak. To address this, we apply a mean filter and smooth the first derivative prior to checking for the upward/downward-going zero-crossings. Smoothing and differentiation can result in degrading the signal-to-noise ratio, which disturbs the peak shape and hence the peak entry and end times. This is addressed by comparing the successive peaks against the previous peaks, assuming that no two peaks will be overlapped or directly adjacent to one another. This assumption is valid because a user cannot be physically present at two distinct ROIs at any given time, and there must be a sufficient time gap (travel duration) between two successive ROI visits.

Next, we pass the signal through the visit-detection and mobility modeling module depicted in Figure 5.5. Here, we perform the baseline correction, peak-shape detection and isolation. The baseline correction is performed to remove the background noise and to make the peaks distinct. An important question is: How to automatically adjust the baselines so as to adapt constantly to

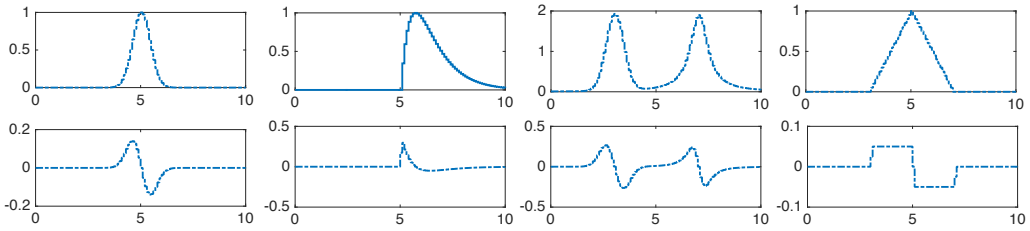


Figure 5.6 – Peak Shapes (top row) and their respective derivatives (bottom).

changing user behaviors? As our goal is to estimate the ROIs without mobility parameters, we do not perform flat-or quadratic-baseline corrections because such methods assume a complete view of the signal. To address this, we keep track of the standard deviation of the incoming points and analyze the points that deviate from the moving mean and the previous degree of standard deviation. This sets the baseline that works irrespective of the peak shape. Furthermore, we need to correctly determine the peak shapes to accurately estimate the location, distance and time spent at the ROI. The shape of the peaks can be detected by taking the successive derivatives, as different peak shapes have distinct derivative shapes as shown in Figure 5.6.

For example, a rising signal has a positive derivative, a signal that slopes down has a negative derivative, and a flat signal has a derivative that is zero. For the peaks associated with human movements, the accidence point coincides with the maximum of the first derivative, and it corresponds with the zero crossing point in the second derivative. If Equation 5.4 is satisfied, we consider the signal as a peak.

$$\frac{d(S(t+1)-S(t))}{d(t)} - \frac{d(S(t)-S(t-1))}{d(t)} > 0 \tag{5.4}$$

This process is not precisely instantaneous as we miss the peak by one $d(t)$, but this delayed detection compensates for false positives in the noisy data. Each *Visit* can be separated into its constituent components by monitoring the average rate of change of slope. Upon arriving at a ROI, either the slope changes to zero or to an infinitesimally small value, as compared to the slope associated with the transition path component for some arbitrary slope $m \in \mathbb{R}$. The two parts are separable, depending on the average rate of change of the slope along the maxima or minima, such that $Tr = Visit \mid \frac{\delta S(t)}{\delta t} = m$ and $ROI = Visit \mid \frac{\delta S(t)}{\delta t} \neq m$. Once the cells belonging to the ROI are extracted, the remaining cells of the visit belong to the rising edge and the falling edge. In order to construct the representative path connecting the ROI, we rely on [39]. Our technique captures the practical nature of human mobility, by considering the fact that, users can move between two ROIs through different paths. We finally extract the best possible path amongst several options to represent the most significant trajectory of the user.

The positions where the slope changes also identifies the ROI entry and exit instants and are used to compute the area. Computing the zero-crossing in the first derivative gives the signal

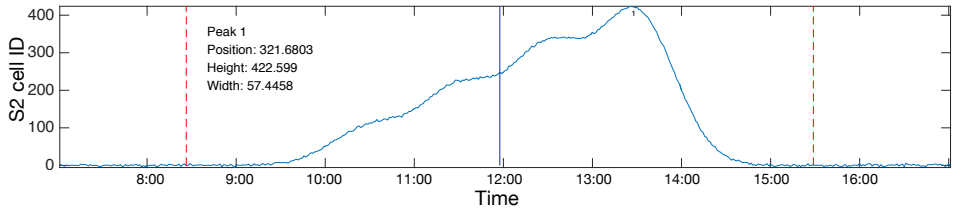


Figure 5.7 – Peak detection and peak detail computation.

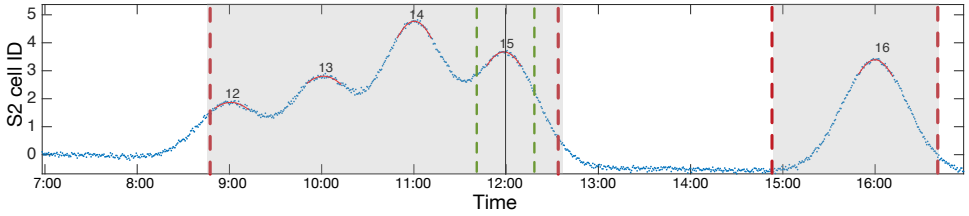


Figure 5.8 – Repeated visit with a changed behavior creates a new ROI.

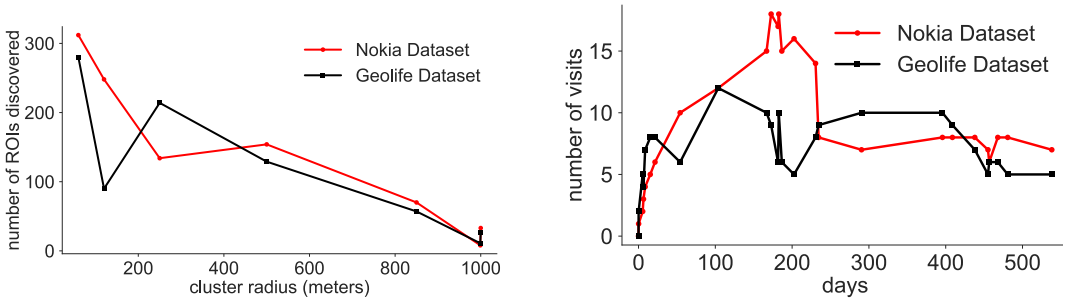
peak-point, irrespective of the signal type, hence the location of this point is an estimation of the mean visit time of the ROI. We attain the values of peak start and end in the process of peak-shape detection, i.e, the first derivative detects the time of the peak start and the second derivative gives the time of peak end as depicted in Figure 7.4d. This process finds the cells corresponding to the individual ROI of a user and the cells associated with the transition path component.

5.6.2 Sub-ROI Discovery

If the frequently visited region of a user is large, it might consist of a combination of smaller ROIs that we term as Sub-ROIs. For example, a university can be dissected into smaller regions such as the math building, engineering building and the cafeteria. Unlike the clustering techniques that require hierarchical clustering to dissect an extracted ROI to find these smaller locations [12], we follow a different approach. A ROI can be visited by a user in two ways. Either the user follows a regular routine of visiting each Sub-ROI (e.g., math building to cafeteria) included in the main ROI (e.g., university) or simply visits one of the sub-ROIs (math building) and returns. A challenge here is to not characterize them as two different ROI visits, when both are essentially part of the same bigger ROI (university).

A solution to address this issue is to check if a new peak/valley lies within the time frame of an already commenced peak/valley. The time bounds can be used to classify the minor peaks as a part of the major. However, if the behavior of the user changes as to take a different start and end route, the peaks can be classified as a new ROI, as shown in Figure 5.8.

To address this, we check if either of the peak-start and peak-end positions match, and we use Dice coefficient to check the similarity of the ROIs. The Dice's coefficient can be expressed as in Equation 5.5 and represents the similarity measure of two sets in range $[0, 1]$, where 0 indicates



(a) Number of ROIs vs. radius

(b) Number of visits vs. time

Figure 5.9 – Trends across two different datasets for parameter estimation.

no overlap.

$$QS = \frac{2|A \cap B|}{|A| + |B|} \tag{5.5}$$

Here, A is the set of cells contained in the main ROI and B is the set containing the cells associated with the sub-ROI. Two ROIs cannot share the same cell, hence any value of QS greater than 0 indicates an overlap. This ensures that we classify the different patterns of visits in a ROI as repeated visits to the same. A mobility model is thus formulated by linking all the distinct ROI's, the representative paths and the transition probabilities. The transition probabilities are estimated based on a mobility Markov chain (MMC) model which accounts for the state-transition matrix as described in [136].

5.7 The Parameter Curse

A key advantage of **Capstone** is the independence from the *a–priori* selected user behavioral parameters. To highlight this advantage, we demonstrate the bias and the privacy leakage resulting from the parameter space. Given a user's spatial trajectory, the ROI discovery problem in the conventional setting is formalized as finding all the distinct ROIs, where each ROI_i is a three-item tuple (lat_i, lon_i, r_i), where r_i is the radius of the region (assuming circular ROIs). Here, the maximum distance and the minimum time between two spatiotemporal points are bounded by fixed thresholds before assigning them to a particular cluster. These clusters are then merged, depending on their spatiotemporal similarity to form a ROI.

In general, users are characterized by distinct mobility profiles, which results in different optimal values of these parameters. Their estimation is also challenging due to the large number of possible combinations, the duration of the available dataset, the sampling rate of the locations and the noise distribution in the recorded data. The parameter values are selected by analyzing the trends in the considered dataset or are derived by logical reasoning, which lacks exhaustive empirical basis. For example, the techniques that use cluster radius as a parameter [11], select

Parameter	Abbreviation	Parameter	Abbreviation	Parameter	Abbreviation
1. Max. Distance	Max_{dist}	6. Max. Point Separation	Max_{dp}	11. Grid Size	S_g
2. Min. Time	Min_{time}	7. Gradient Threshold	Tsh_g	12. P Value	Val_p
3. Max. Time	Max_{time}	8. Seed number	Num_s	13. Height	H
4. Num. Eigenvectors	Num_{ev}	9. Vector Length	Len_{vec}	14. Cluster Radius	C_r
5. Min. Num. Points	Min_{points}	10. Minimum Visits	Min_{visit}	15. Minimum Speed	Min_{speed}

Table 5.1 – Parameters used by existing ROI discovery techniques.

it at a point, which results in a 'significant' change in the slope between the number of clusters vs. the cluster radii, known as a *knee* in the plot. We use the same technique proposed in [11] to extract clusters and then derive the parameter values across two geospatial datasets as shown in Figure 5.9a. To select the value of *minimum visit*, either the knee in the plot of time vs. number of visits to a particular ROI is selected [136] as shown in Figure 5.9b, or the duration of the collected dataset is taken into account [68]. We clearly see different trends followed by the two datasets. This results in a possible bias when generalizing and comparing the results obtained with different techniques or the same technique on different geospatial datasets.

Therefore, a comparison performed with a partial knowledge of the effect of altering the settings of the clustering algorithm will result in arbitrary conclusions. This has led to different parameter values in the published works that are dependent either on the application scenarios as in [170], or on the behavior as stated in [11], which results in inconsistent derivations of comparative results. We also notice their disagreement about the significance of the effects of certain parameters. For example, the results highlighted in [170] and [236] lead to conflicting conclusions regarding the importance of the *maximum time*, between two coordinate points.

Finally, these techniques extract clusters, characterized by fixed shapes (mostly circular). In the literature, the circular cluster shape is based on the diffusion theory in Kulldorff's spatial scan statistics [117]. The techniques that assume a pre-defined shape provide assurance of a complete enumeration of all the regions of that shape in a given area. However, clustering techniques, when applied for ROI detection assume a circular shape that might not represent reality. Setting predefined circular windows to define the potential cluster areas will result in difficulties in correctly detecting actual noncircular ROIs [234] and furthermore to estimate the areas and total time spent. These issues show the importance of not relying on prior assumptions regarding either the parameters or shapes for devising generalizable and conclusive ROI-detection algorithms. We present an exhaustive list of predetermined behavior dependent parameters used by popular ROI discovery techniques in Table 5.1.

These parameters are measures of individual mobility dynamics [188]. Therefore, in a situation where the adversary requests a data provider for aggregated/sparse mobility data, a knowledge of these parameters can increase the background knowledge to carry about membership inference attacks. Parameters such as radius of gyration, mobility entropy and average number of visits of an individual, have been shown to de-anonymize users from aggregated databases [251]. A recent work to estimate privacy risk of individuals based on the individual mobility features show that

several parameters such as the maximum distance/time between locations, total distance traversed per day, number of distinct locations and others increase an individual's risk of identification against location sequence construction attacks, home and work place attacks, location probability attacks, etc. [188]. Therefore, we argue that a technique to extract user mobility models even from sparse data without relying on user's mobility parameters is beneficial.

5.8 Evaluation and Discussion

In this section, we evaluate **Capstone's** effectiveness in mobility modeling without any parameters and its operational efficiency on smartphones based on the implementation of the proposed technique on a DSP chip. We also perform privacy analysis, by quantifying the accuracy and risk of two popular attacks performed on the user's exposed locations. All the evaluations are performed using the Nokia dataset [124], Geolife [266] and a third dataset annotated with the ground truth. These datasets contain geospatial trajectories of more than 370 users, collected in Switzerland and China. ROIs being the key component of the mobility model, our focus is on their validation. The extraction of representative trajectories from the set of transition paths, provides precision and recall rates exceeding 80% as shown in [39].

We configure the google S2 library to project each coordinate pair onto a cell of dimension $38m^2$. It could be argued that the cell size involves a arbitrarily chosen parameter in the process. However, our choice is motivated by the localization accuracy of a typical GPS sensor and the performance complexity involved when subdividing the cells to the leaf level.

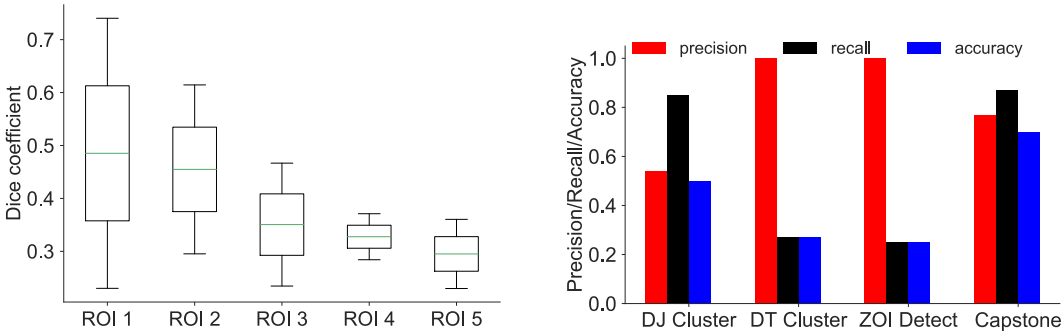
The publicly available datasets are devoid of the ground truth due to privacy concerns. Therefore, we collect an additional dataset by providing a mobile application to one of the co-authors of this paper as a part of our data collection campaign.[5] The application logs the latitude, longitude, timestamp, acceleration, altitude, horizontal and vertical accuracy of the GPS coordinates. The data points are collected at a sampling rate of 5 seconds with a granularity of resolution up to 5 meters for a period of 15 weeks. The ground truth is captured by periodically attesting the visited regions of interest, average time spent and the approximate area.

5.8.1 Visit Consistency

Here, we perform a qualitative evaluation by using the mobility datasets to guarantee consistency of the discovered ROIs using metrics derived in published works based on the same datasets [236].

The task of ROI extraction is synonymous with unsupervised clustering. Therefore, we first validate our results by relying on the knowledge about the data and the properties of ROI visits. To this end, we use the properties derived by Thomason et al. [236], which hold for a majority of the Nokia dataset users. They comprise of: (i) A typical user makes an average of 2 to 15

[5]Mobility data collection campaign: bread-crumb.github.io



(a) Dice index distribution

(b) Ground truth validation

Figure 5.10 – Comparison of the ground truth analysis.

Num. ROIs	2-5	6-9	10-12	**Max. Stay Time**	5-8 Hrs	9-10 Hrs	11-26 Hrs
User %	37	51	12	**User %**	67	26	7
Short Visits	<10 min	<15 min	<30 min	**Stay Time v/s Travel time**	3:2	4:1	2:3
User %	22	36	42	**User %**	35	54	11

Table 5.2 – Visit accuracy evaluation based on Nokia dataset.

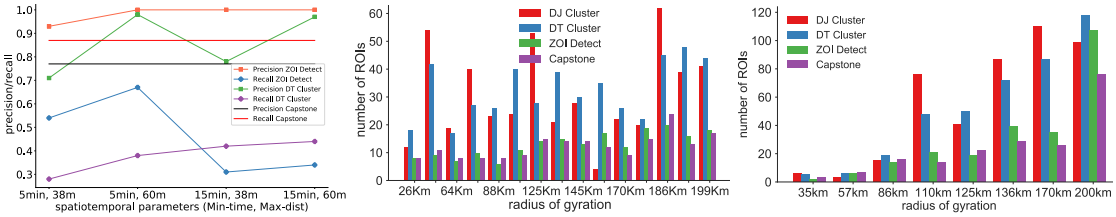
distinct ROI visits a day, (ii) a visit does not exceed a period of 2 days, (3) a user spends 60% of the time at the ROI and not more than 40% traveling to the location. Our results (see Table 5.2) corroborate these properties.

A drawback of our approach is its high sensitivity to even small stoppages occurring on the path, due to which unintentional delays can be classified as ROIs, e.g., a bus stop on the way of an intentional destination. Furthermore, we do not rely on the Min_{visit} parameter, which classifies even a single visit as a ROI and results in some false positives. This parameter is often selected depending on the duration of the available dataset, which does not reflect the true periodicity of visiting a particular place. If the periodicity is very low, this will be reflected through the learning algorithms if the extracted ROIs are utilized for applications such as mobility prediction [35]. Thus, although our technique might result in some false positives, it ensures that none of the ROIs are filtered out either based on the dataset duration or the mobility behavior. The additional outliers occur due to property (iii) which does not hold for bus/metro/train stops.

Next, we examine if the repeated ROI visits with a differing user behavior (different entry or/and exit points) results in creation of new ROIs. To analyze this, we use the Dice coefficient to assess the similarity, and show its distribution across the considered set of ROIs spanning distinct areas. As depicted in Figure 5.10a, marginal deviations in user behavior associated with repeated visits, does not lead to creation of new ROIs.

Clustering algorithm	Parameters
DJ Cluster	Min_{speed}: 0.4 (km/hour) / C_r: 60.0 (meters) / Min_{points}: 10
DT Cluster	Max_{dist}: 60.0 (meters) / Min_{time}: 900 (seconds)
ZOI Detect	Max_{dist}: 60.0 (meters) / Min_{time}: 900 (seconds) / Min_{visit}: 6

Table 5.3 – Clustering algorithms with the default parameter values.



(a) Ground truth validation (b) Number of ROIs: Nokia dataset (c) Number of ROIs: Geolife dataset

Figure 5.11 – Comparison of the ground truth with different parameter settings and the number of ROIs using two different datasets.

5.8.2 ROI Accuracy

In this section, we perform quantitative analysis using the dataset annotated with the ground truth and compare our results with three popular time-space-density-based clustering algorithms commonly used for ROI extraction.

Here, we validate the accuracy of the discovered ROIs with respect to the ground truth and compare our results with three clustering techniques. We consider Density Joinable Cluster (DJ Cluster) [271], Density Time Cluster (DT Cluster) [102] and ZOI Detect [136]. DJ Cluster computes ROIs based on the number of points within a certain radius and merges the clusters if they share at least one common point. The points are also clustered together if they satisfy the Min_{speed} bound. DT Cluster aggregates points lying within predetermined spatiotemporal bounds. These clusters are then treated as valid ROIs. ZOI Detect follows a similar strategy as DT cluster but relies on an additional parameter Min_{visit} as a threshold and merges the clusters upon intersection. The parameter space of these techniques and their values are shown in Table 5.3. These values selected in published works are based either on the dataset trends [136] or on the mobility behaviors [271].

For the ground-truth comparison, the 34 ROIs of the considered subject were selected with a clear definition: 'any place where the subject visited with an intentional purpose'. These regions include places such as cafeterias, restaurants, bus/train/metro stops, sports arenas, bookstores, office and work places, and excursions. The ground-truth evaluation was performed by computing the precision, recall and accuracy. As the set of true negatives is infinite (ROIs not visited by the user and not discovered by the algorithm), $Accuracy = \frac{TP}{TP+FP+FN}$. A comparison of **Capstone** with the clustering techniques, with respect to the ground truth, is shown in Figure 5.10b. To ensure consistency, the ROIs with the clustering techniques are computed with the default

parameter values given in Table 5.1.

We see that DT Cluster and ZOI Detect have a very high precision and low recall and accuracy. This indicates that these techniques detect a large number of ROIs that are not contained in the true ROI set. This is clearly due to the spatiotemporal bounds being too rigid, which results in considering arbitrary clusters as ROIs. DJ cluster, however, has higher recall and low precision. Here, we see that the Min_{speed} eliminates the occurrences of false negatives, whereas, the Min_{points} creates high number of false positives. Increasing the Min_{points} can address such occurrences, as it requires a higher density of points, thus creating only valid ROIs. In case of **Capstone**, we have a few false positives due to the high sensitivity and only three false negatives. The false negatives are the transportation stops where the user does not have to wait due to planned time synchronization, resulting in a constant average slope.

To better understand the parameter influence, we consider four different parameter sets for the values of Min_{time} and Max_{dist} as seen in Figure 5.11a. We see that the parameter Min_{visit} always correctly classifies a region as a ROI, thus leading to high precision rates. We can also see that larger values of Max_{dist} results in higher precision and recall in DT Cluster. Max_{dist}, thus plays a vital role in determining precision, compared to Min_{time} parameter in the considered dataset. These results highlight the importance of selecting the parameter space which is challenging to determine a–priori.

To present qualitative results we evaluate the Nokia and Geolife dataset. In the absence of ground truth, choosing relevant metrics for comparison is a challenging problem. To address this, we explore the number of ROIs discovered as it directly influences the accuracy of the technique. A lower number may signify the merging of multiple ROIs leading to the loss of information, such as the total area and the time of entry and exit from the respective ROI. Whereas, a large number indicates a higher number of false positives. We first show the results for the Nokia dataset in Figure 5.11b and the Geolife dataset in Figure 5.11c. In order to consider different mobility behaviors, we select users with distinct activity areas captured with respect to the radius of gyration of movement.

DJ Cluster and DT Cluster detect a significantly high number of ROIs, not typical for an average user. In case of DJ Cluster, we find that the parameter Min_{points} creates a large number of ROIs. However, we argue that if the sampling rate of the dataset is high, the Min_{speed} could play an important role in further increasing the number of clusters. Whereas, in DT Cluster Min_{time} parameter results in a higher frequency of visit separations increasing the total number ROIs. We see that the number of ROIs discovered by ZOI Detect is lower than DT and DJ Cluster. This is due to the merging of individual clusters upon intersection, in addition to extracting the most frequent clusters governed by the Min_{visit} parameter. In general, if the parameters satisfy cluster merging, multiple clusters merge and form a large ROI; ROI division occurs if this bound is missed by even an infinitesimal small value. This results in the fluctuation of the number of ROIs solely due to the parameters. We cannot validate the accuracy of the ROIs detected by **Capstone** in this case, however, we observe a consistency between the distance and the ROI

number. We also do not observe an alarming number of ROIs. We exclude DBSCAN [64] and TD clustering [77] from the comparison, as they are similar in the parameter space to the techniques already considered.

The ROI area in our approach corresponds to $38m^2 \times cellnumbers$. This results in a significantly smaller areas compared to the clustering techniques and overlaps with the ground truth area. This is due to the set of cells comprising the ROI, which corresponds to the actual ROI area relative to the user movement. In contrast, the clusters encompass the area not covered by the user, due to reliance on bounding circles, where the centroid corresponds to the mean of the points and the radius corresponds to the maximum distance between the centroid and the points. The same reasoning holds for the time spent in the ROI. This representation method can introduce a large area of dead space, and we argue that convex hulls would be more suitable than bounding circles for representing ROIs in the clustering techniques.

5.8.3 Complexity and Power Consumption

Next, we evaluate and compare the techniques with respect to their computational complexity and power consumption. We implement Capstone, DJ Cluster, DT Cluster and K-means (as a reference) on a TI OMAP-L138 C6000 DSP+ARM Processor (Figure 5.13a) present in many smartphones.[6] The Dual-Core SoC contains an ARM9 general purpose processor (GPP) and a C674x DSP core. As the performance and scaling is also dependent on the actual implementation of the algorithms, we do not optimize any techniques and derive only the asymptotic performance. A typical workflow between the GPP and the ARM processor is depicted in Figure 5.12.

We benchmark the performance at various dataset sizes and consider the average time after 10 runs on each dataset size (see Figure 5.13b). We see that, capstone reduces the runtime latency as compared to the rest by a factor of approximately 2.5. The key reasoning behind the performance is: (i) the stackable non-blocking signal-processing pipelines, (ii) maximizing the computations per clock cycle by mapping these pipelines to the DSP architecture, (iii) efficient execution of all the filtering and peak-detection stages by utilizing the five multiply-add-accumulate units (MAC) in parallel, and (iv) space-transformation which facilitates carrying out all the operations on integers rather than 3-dimensional floating points. The performance of K-means rapidly deteriorates as the execution depends on the disk IO bound, and continually paging the RAM to access the distance array dramatically increases the runtime. Similarly, the agglomerative/hierarchical clustering techniques suffer through the same drawback. An additional drawback of such algorithms is due to the fact that they operate in several steps [11]. This is done, by first clustering the points in the temporal domain and consequently in spatial domain, or by extracting locations that span large areas and dissecting them into smaller regions in the second iteration over the dataset. This results in increased time and computational complexity, hindering the possibility of operating them in real-time scenarios on resource-constrained devices.

[6] TI C6000: www.ti.com/product/OMAP-L138

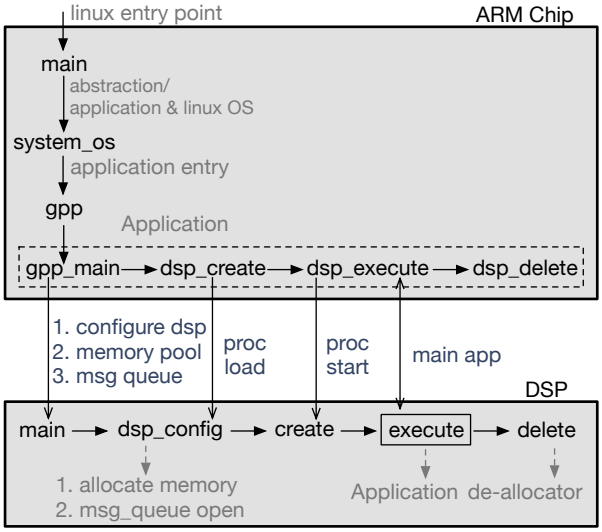
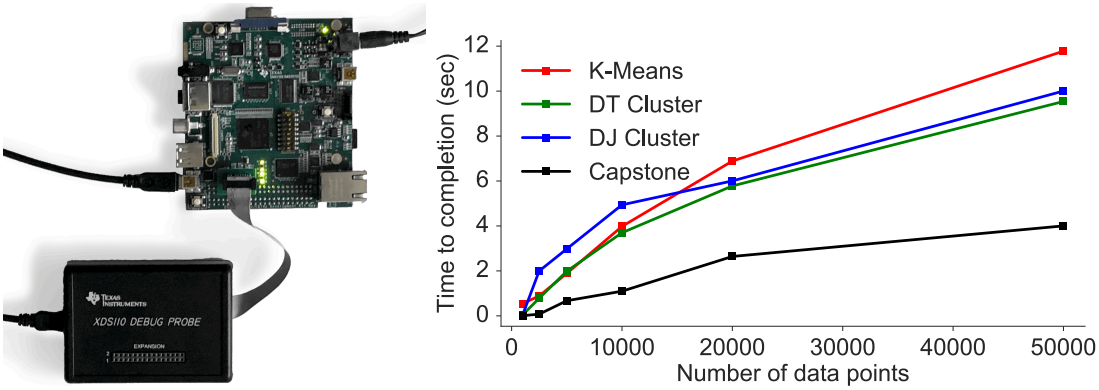


Figure 5.12 – A typical workflow of a GPP-ARM based SoC.

In order to theoretically compute the complexity bounds, we consider a total of n coordinate points from which the ROIs have to be extracted. We assume k unknown ROIs, as we do not have a-priori knowledge on the number of clusters that will be detected. In the case of space-time-density based clustering techniques, there are multiple blocking steps involved. For each coordinate assignment to a *stay region*, the Euclidian distance and time bounds are computed, and checked with the neighboring points. Once the stay region is estimated, the centroid of the region is computed. This step has an overall complexity of $O(kn)$. The next step involves iterative merging of clusters based on distance bounds and is characterized by a complexity of $O(\sum_{i=0}^{k-1}(k-1)^2)$. Scalar product of both the steps measures the total complexity. In case of Capstone, the preprocessing and peak-detection steps, the low-pass filtering, curve fitting, and the mean filtering contribute to a complexity of $O(2n)$; and the differentiation and baseline corrections contribute to $O((2n)^2)$, which results in a total complexity of n^2. We can consider the operations as a $n \times n$ scalar matrix C multiplying a scaler vector v of length n; these operations result in a total of n^2 multiplications and $n(n-1)$ additions. These multiplications and additions are parallely executed across the five MAC units in a non-blocking fashion contributing to the runtime improvement.

Next, we compare the power consumption at various dataset sizes as shown in Table 5.4. The power drawn by a process can be categorized in to baseline and active power. The former includes the static power (leakage), phase-locked loop, oscillator power and various subsystem components that cannot be turned off through the on-chip power management module. Active power is the consumption due to the active parts of the SoC, which is dependent upon the frequency, utilization, read/write balance and switching (GPP-DSP). We consider the total power as the sum of these individual power consumptions measured using the TI's EnergyTrace tool.[7]

[7]Energy Trace: www.ti.com/tool/ENERGYTRACE



(a) Experimental setup (b) Execution time comparison

Figure 5.13 – Experimental setup and performance comparison.

# data points	1000	10000	50000	100000
K-Means	**0.27***mW*	**1.67***mW*	6.76*mW*	11.93*mW*
DT Cluster	0.33*mW*	2.33*mW*	8.23*mW*	14.66*mW*
DJ Cluster	0.39*mW*	3.14*mW*	8.95*mW*	15.28*mW*
Capstone	3.13*mW*	3.78*mW*	**4.04***mW*	**6.79***mW*

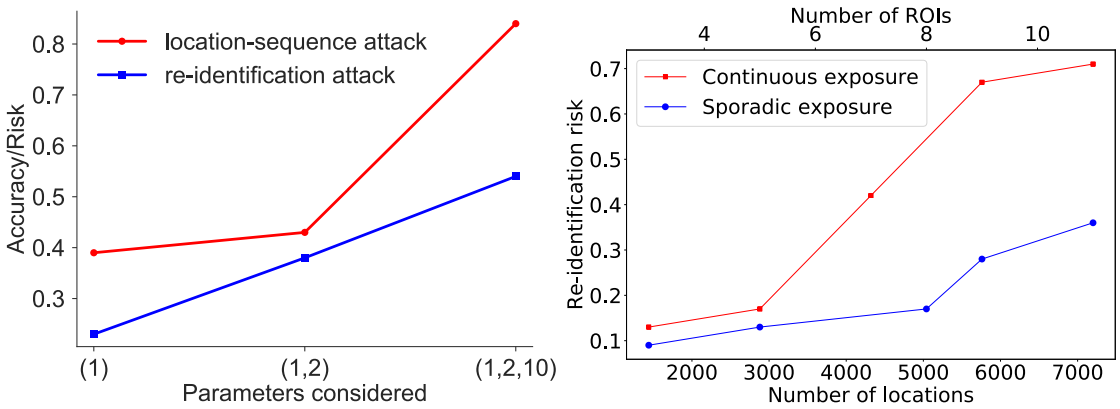
Table 5.4 – Power consumption comparison (baseline + active power).

Capstone; inherently a DSP implementation draws a higher baseline power as compared to the GPP implantation of the clustering techniques. This is due to the power consumed in configuring the DSP chip and setting up the shared memory pool, the message queue between the GPP and the DSP and the real-time operating system (RTOS). We clearly see in the results that, as the dataset size increases the power consumption of the clustering techniques rapidly escalates. However, the DSP implementation leverages the efficient power management capabilities of the RTOS that uses the chip power-efficiently, while still providing high performance.

5.8.4 Privacy Analysis

The privacy by design approach cannot rely on measures such as differential privacy to perform privacy analysis, unlike the data concealing approaches. We follow the methodology specified by Shokri et al. [217] involving construction of a schedule consisting of an application, a LPPM (location privacy preserving mechanism), an attack and the evaluation metric. In our case, an application can be any continuous exposure LBS at the user's end and our LPPM is the minimization of the exposed locations via on-board processing. We consider two commonly used attacks: (i) location-sequence attack, and (2) re-identification attack. The success of these attacks depends on the adversary's prior knowledge, i.e. access to some traces of users or public information such as visited locations. Finally the user's privacy is quantified in terms of the correctness/incorrectness of the attacks by using the Location-privacy and mobility meter[8] and

[8] *LPM*^{2}: icapeople.epfl.ch/rshokri/lpm/doc/



(a) Adversarial attack accuracy

(b) Adversarial attack risk

Figure 5.14 – Privacy analysis based on select users from Nokia dataset.

privacy-lib.[9] In case of a privacy by design based system, we can clearly see (Figure 5.14b) that by minimizing the locations shared with the third party services, we lower the adversaries prior knowledge, hence the risk of the attacks resulting in an increased user privacy as stated in [217]. Furthermore, by not relying on behavioral parameters, we lower the adversaries background knowledge contributing to enhanced user privacy as depicted in Figure 5.14a. Here, we consider three parameters for evaluation: (1) Max_{dist}, (2) Min_{time}, and (3) Min_{visit}. We do not take service utility in to account as we assume a system based on a trusted computing environment, which does not compromise on service utility [131]. However, in case of techniques such as obfuscations or anonymization, pseudo-locations are used for the last hop, in which case the accuracy depends on the amount of distortion added to the user's true location.

5.9 Related Work

We present a new perspective on treating mobility trajectories as space-time signals to model human mobility in a computationally efficient manner. However, as a major aspect of our technique is ROI and transition discovery from the signals without relying on any behavioral parameters, we review the state of the art in this field. The existing techniques and their dependent parameter space is depicted in Figure 5.15.

Clustering. An iterative approach for extracting the ROIs using clustering was proposed by Ashbrook et al. [11]. The ROI granularity was improved by setting the spatiotemporal bounds derived by analyzing their variance with respect to the dependent values. Montoliu et al. [170] proposed a two-level grid-based clustering approach, where the points are successively clustered in the temporal and spatial domain. Algorithms such as k-means [103], neighbor density-based clustering (DBSCAN) [64] and time-density clustering [77] are used to detect clusters in spatiotemporal datasets, which are then considered as ROIs. Zheng et al. [268] proposes a

[9] privacy-lib: github.com/pellungrobe/privacy-lib

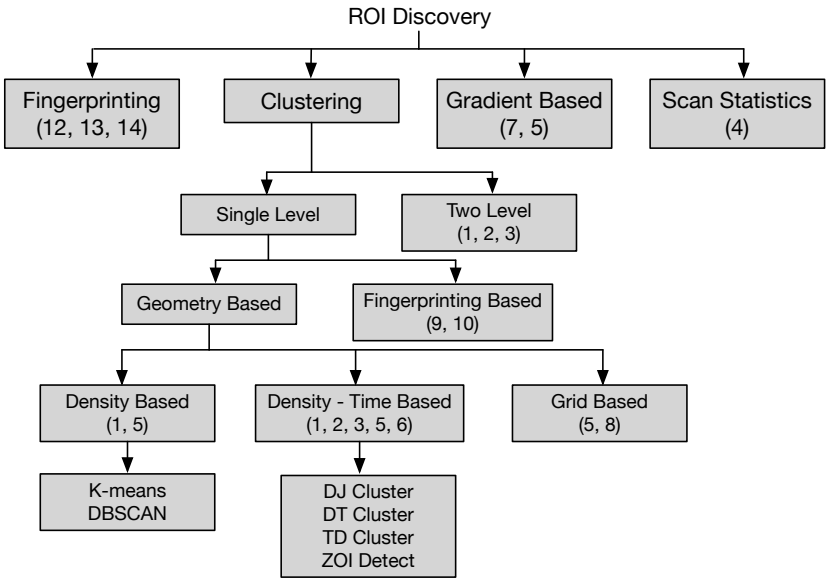


Figure 5.15 – Summary of ROI discovery techniques and their categorization. The numbers represent the dependent parameters from Table 5.1.

clustering-based ROI extraction technique where the parameters are estimated by observing the distribution of movement density.

Fingerprinting. This technique relies on estimating a user's location fingerprint based on the detection of stable radio environments indicating a ROI. Farrahi et al. [68] extracts ROIs by first forming a vector of the visible cell towers and then using repeatability rates and location transitions over time to filter out insignificant places. This technique is particularly applied to identify places, such as work and home.

Scan Statistics. This technique proposed by Fanaee-T et al. [67] moves a cylinder of varying radii and height over a spatiotemporal space, where the surface covers the spatial dimension and height covers the temporal dimension. The cylinders are then sorted depending on the F-score, and an additional parameter called p-value is used as a threshold to filter insignificant places. The authors show the ability of this technique to handle complex data and reduce noise, at the cost of loosing the original shape of the ROI. However, spatial scan statistics is based on a frequentist view and does not depend on other priors like in Bayesian statistics.

Gradient Based. Thomason et al. [236] proposed a technique that combines the advantages of both k-means and DBSCAN. It does not place a bound on visit duration or enforce any delay on trajectory points being considered, thus it can be operated in instantaneous time. The thresholds used in this work are empirically derived and the results achieved have minimum bias due to the parameters. Louail et al. [153] propose a technique to extract ROIs from trajectories belonging to a group of users without relying on the commonly used spatiotemporal bounds. However, they consider a group of points at a particular time, as a ROI if the density of users at that location is

greater than a predefined threshold.

5.10 Conclusion

The paradigm shift towards cloud computing has encouraged LBS providers to deploy their infrastructure on untrusted cloud providers. This context has created several privacy and confidentiality issues by aggregating large amount of user location information in third-party datacenters. Although, several techniques have been proposed to curb the location privacy leakage, a large gap still exists between the theoretical body of knowledge and the real-world applications. Furthermore, the new EU data protection regulations have imposed stringent restrictions on the volume of data aggregated, processed and stored at the service provider's end.

In order to address these issues and facilitate the trend of on-board processing at the user's end, we have proposed a novel perspective on spatiotemporal computation by treating trajectories as space-time signals. We have leveraged the properties of these signals to reduce the computational complexity and power consumption. We have presented **Capstone**, that illustrates this approach on mobility modeling task and shows that, not only do the signals preserve all the key knowledge contained in the trajectories but also formulate the mobility models with a high accuracy. We have evaluated in depth the proposed technique by first analyzing it only from the signal processing perspective, and then verifying whether it satisfies already proven measures of human mobility. Our validation with the ground truth achieves precision and recall rates exceeding 80% and achieves results on par with the conventional clustering approaches. We have performed the complexity and power consumption analysis by implementing **Capstone** on a DSP chip commonly present in many smartphones. Furthermore, we have experimentally depicted the bias resulting from the stringent parameter bounds in the mobility modeling process and the associated privacy leakage. We have also demonstrated the suitability of our technique to extract ROIs from a larger variety of datasets and across different mobility behaviors.

6 Privacy-Preserving Location-Based Services by using Intel SGX

Abstract

We are witnessing a rapid proliferation of location-based services, due to the useful context-aware services they provide their users. However, sharing sensitive location traces with untrusted service-providers has many privacy implications. Although, user-data monetization is the core economic model of such services, offering private services to concerned users will be a beneficial functionality in the coming years. Existing solutions include location perturbation, k-anonymity and cryptographic primitives that trade service accuracy or latency for enhanced user privacy. We introduce a novel approach for privacy preserving location-based services by using the Intel Software Guard eXtensions (SGX). We implement a simple location-based service using SGX and gauge its performance in terms of efficiency and effectiveness, in comparison with its bare-metal implementation. Our evaluation results show that SGX contributes a marginal overhead but also provides near-to-the-perfect results in contrast to spatial cloaking with k-anonymity whose performance deteriorates as the degree of desired privacy increases. We show that hardware-based trusted execution-environments are a promising alternative for offering proactive and de-facto location-privacy in the context of location-based services.

Keywords: Location privacy; Intel SGX; Privacy-preserving LBS.

6.1 Introduction

The ubiquitous nature of mobile phones equipped with internet connectivity and global positioning functionality (GPS) has led to the widespread development of location-based services (LBSs). Such services collect and store a large amount of user-location data in the untrusted cloud, which exposes users to several privacy risks. Data breaches and unlawful exchanges [4] can enable curious adversaries to derive personally identifiable user information (PII) by applying simple heuristics [74]. This information can be used for blackmailing or stalking purposes [3]. Thus,

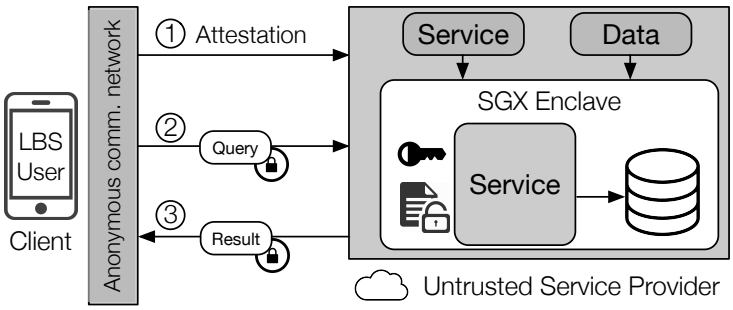


Figure 6.1 – Private-LBS: The client can verify the application security by performing attestation. The application and database is embedded in the enclave. The query is encrypted, which can only be decrypted and processed inside the enclave. The result sent by the service-provider can only be decrypted by the client. Thus ensuring end-to-end-to-service-provider encryption.

user privacy consideration will be a key factor in determining the success and adoption of LBSs in the coming years.

Service provider's (SP) application usually resides in the cloud where the user data is processed to yield the desired result. The SPs rely on virtual machines or containers to insulate the underlying platform from the users and to offer isolated execution. Although, such measures safeguard the SPs against the users, the latter have to implicitly trust the SPs and the execution platforms. Several solutions have been proposed to address the privacy concerns in LBSs, such as spatial cloaking [96], k-anonymity [78] and cryptographic primitives [72]. However, such techniques are not widely adopted in practice, either due to their low accuracy or high latency.

We propose an architecture for Private-LBS, it relies on Intel's next generation hardware-based trusted execution-environment called **Intel SGX**[1]. Intel SGX provides a reverse sandbox that enables independent software vendors to run a software module on an untrusted cloud. It designates a container that isolates the program and data from all the other software, potentially malicious OSs and the hypervisor. Furthermore, it offers a verification mechanism for authenticating the remote hardware platform and its state. We use these features to implement a **Point-of-Interest Locator** (POI-Locator) application that imposes anonymity and indistinguishability to enforce user privacy. We quantify the overheads involved in such a system, with respect to its bare-metal counterpart. We also compare its performance with a popular hybrid location-perturbation algorithm: spatial cloaking with k-anonymity. An architectural overview of our system is depicted in Figure 6.1.

6.2 Background

In this section, we present the background information regarding the features offered by Intel SGX. These features are leveraged in our system design to offer private location-based services.

[1]Intel SGX: https://software.intel.com/en-us/sgx

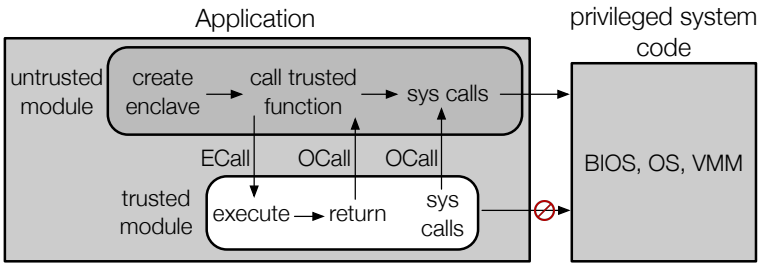


Figure 6.2 – Trusted & untrusted modules of SGX application

6.2.1 Intel Software Guard eXtensions (SGX)

SGX is Intel's new architecture extension for providing a strong and provable isolation of binary code that runs concurrently and shares resources. It enables an application to construct protected regions of memory at a virtual address space called **enclave**. The enclave can be created and destroyed using certain privileged instructions. An application code and the data can be embedded into the enclaves and is ensured protection from the outside world. SGX provides guarantees that no privileged software, even with the root access, can view the contents of the enclave. Furthermore, all the contents belonging to the enclave that lie outside the enclave are encrypted. As an enclave has a limited size, we can create multiple enclaves that are isolated from one another and distribute data using shared keys.

As these features can also be used to create super malware, the enclave is prohibited from executing any privileged instructions, including systems calls and I/O operations. Additionally, the enclave code can only run in the user-mode and not in the kernel-mode. A typical SGX application consists of two modules: the untrusted module that executes security uncritical code and the trusted module that executes critical code inside the enclave as shown in Figure 6.2. These two modules communicate via two function calls: ECall (trusted) and OCall (untrusted). An ECall function enters an enclave and the OCall leaves the enclave. Therefore, an OCall is made every time the enclave wants to execute a privileged instruction. Evoking an OCall triggers the CPU to switch from the enclave mode to the user mode. The switching results in a certain overhead and can open up the enclave to various attacks. Thus, the OCalls for I/O operations are only used during the enclave debugging phase.

SGX also provides a remote attestation scheme to attest to the security offered by the untrusted cloud-provider. This feature enables a remote user to verify whether an application is running inside a legitimate enclave and does not leak any information, thus, leaving only the processor operation and the security keys printed on the die to be trusted by the user.

Memory Encryption. All the enclave data and code is transparently encrypted in the memory by the SGX Memory Encryption Engine (MEE). The MEE uses a combination of Merkle trees and a 56-bit AES counter, producing a 128-bit integrity key and a 512-bit universal hash key to encrypt the enclave pages. The keys are generated at boot time and are placed in the privileged

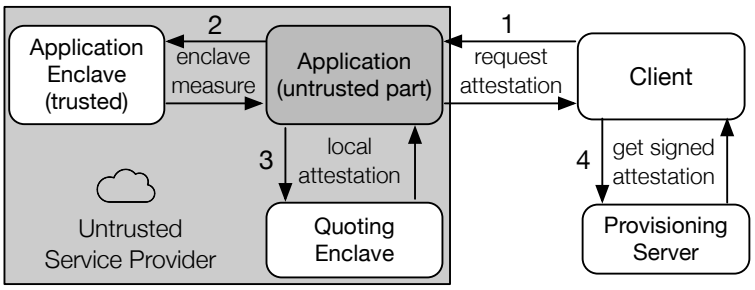


Figure 6.3 – Remote attestation procedure

MEE registers that are destroyed at system reset. Thus, access to this protected memory region called Enclave Page Cache (EPC) is restricted at the hardware level. The Enclave Page Cache Map (EPCM) restricts the pages that the enclave is permitted to access in the EPC. The MME therefore also protects the data in the RAM from unauthorized access.

Remote Attestation. Typically, the clients have no assurance regarding the software running at the remote server. To address this, SGX implements the remote attestation mechanism that guarantees that the application is not tampered with, and is transparent about how the private user data is treated. Every enclave, when initialized, generates a certificate containing its measurement, vendor ID, product ID and other enclave attributes. The remote attestation procedure is depicted in Figure 6.3. During the initialization phase, the measurement of the enclave contents is taken by performing a hash over its memory pages. First, the enclave obtains a signed attestation for itself from a specific Intel enclave, known as the quoting enclave, through a local attestation procedure. The attestation is performed over the enclave's measurement to create a report. The quoting enclave checks the report and generates a signed attestation quote, that, intern is sent to the remote user.

When the remote user demands an attestation, a loader process which connects to the Intel's provisioning server is initiated. The purpose of the service is to verify the signed attestation quote of the enclave. Intel burns two secret keys into the CPU: a provisioning secret burnt during manufacturing and a sealing secret burnt at the boot time. The provisioning secret is shared with Intel for the attestation service, whereas the latter is not accessible outside the CPU. The provisioning service checks the key, derived from the provisioning secret to attest the enclave. In the case of successful attestation, a report is created and digitally signed, attesting that the CPU is indeed running in a secure mode where the memory is encrypted. The loader process now sets up a secure channel to the provisioning server and can download the intended software and data into the encrypted RAM for execution. The software module and the data can also be encrypted and saved to the disk. When the SGX-enabled processor connects to the provisioning server via the enclave, it also receives an attestation key, that can be sealed and stored to create further attestations. This reduces the entities to be trusted to only Intel's remote attestation service, as all other infrastructure is locked out by the encryption.

Sealing. As discussed above, when the enclave is instantiated, the MEE provides the data integrity and confidentiality. However, the enclaves are stateless, i.e., upon terminating the enclave process, the data stored within the enclave will be lost. Sealing is a special feature provided in order to store data outside the enclave, if the data is meant to be re-used at a later stage. When invoked, the data is sealed using persistent sealing keys derived from the CPU to encrypt and integrity-protect the data. The sealed data block can be unsealed either by the enclave that sealed it or by the software vendor, depending on the key used for sealing. This ensures the confidentiality, integrity and authenticity of the data.

6.2.2 SGX in Practice

SGX has been used to enforce privacy in the smart-grid infrastructure, to secure the energy consumption traces of users [130]. Here, to perform analyses over the user data, SGX is used as an intermediary entity between the smart metering devices and the SP. Only the meta records created by the intermediary entity are then sent to the SP for billing purposes. The security features offered by SGX have also been used to implement a content-based routing (CBR) engine inside the enclave [197]. CBR is not very popular as it necessitates routers to view the data in plain-text, which poses security threats. However, when message filtering is performed inside the enclave, the routers remain oblivious to the messages. [98] uses SGX to make secure two-party function evaluation more efficient as compared to traditional cryptographic operations that are too slow for practical applications. Along similar lines, [196] proposes the use of SGX for providing privacy guarantees for MapReduce operations. In all the above applications, it was found that porting the applications inside the SGX enclaves results in a superior performance compared to the cryptographic mechanisms. These results motivated us to exploit the privacy guarantees offered by SGX and apply it to a domain where privacy is of extreme importance: LBSs.

6.3 System Description

In this section, we present our system model, the adversary model and the protocol design.

6.3.1 System Model

Our system model consists of the following three entities.

- The **Client** is the end user having a subscription to the LBS. In our application, the client sends her current location to the POI-Locator service provider to receive information regarding the nearby (user-configurable distance) points of interest (POI) that can include restaurants, pubs, cafes, gas stations etc.
- The **Service Provider** receives the client's current location, computes the nearby points of interest using a local database and returns the result back to the client. The result should

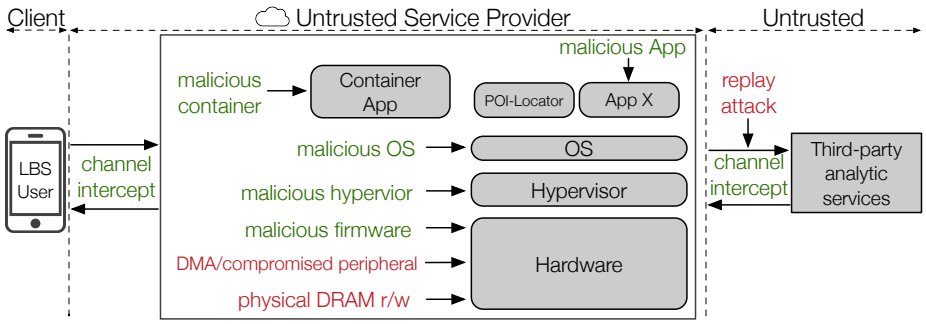


Figure 6.4 – Adversary Model

include the name and category of the place, address and distance from the user's current location.

- The **Infrastructure Provider** hosts the POI-Locator application and provides the SGX-based machine to run the service provider's application: for example, a public cloud platform offering cloud services.

6.3.2 Adversary Model

Considering the above system model, the adversaries from the client's perspective are both the SP and the infrastructure provider (IP). We also assume that the SP does not trust the IP. Hence essentially, the SP wants to keep its code confidential and the client wants to keep her location-coordinates private.

The hardware platform and the system software running at the client's end are assumed to be trusted. The IP is treated as untrusted and malicious or compromised, capable of executing any arbitrary software or modifying the OS or the bootloader. The attacker is assumed to be able to control all the privileged software, including the hypervisor, firmware and the entire management stack. As the resources in a public cloud domain might be shared amongst multiple SPs, we also assume that all the other services/applications running at the IP's end are malicious. The IP administrators are not trusted and are assumed to be curious or malicious. The SP is assumed to be honest but curious, i.e. the application always computes and returns correct results to its clients, however they can use the information regarding a user's identity and/or location traces for any kinds of activities. The SPs can also leverage analytical services from other third party entities such as Azure[2]. We also assume that such services are honest but curious.

We also consider that the processors equipped with the SGX functionality are to be trusted and an attacker is not capable to physically tamper with them. Figure 6.4 shows the possible attacks considered by our system. SGX provides implicit protection against the attacks marked in green but not those marked in red. Table 6.1 shows a detailed list of hardware attacks, against which

[2]Microsoft Azure: azure.microsoft.com

Attack	Description
Denial of Service	Host machine physically taken off the network
Port attack	Malicious software running via the debug ports
Bus tapping attack	Tap motherboard bus's to track or modify traffic
Chip attack	Power and timing analysis to reverse engineer code
Side channel attack	Reverse engineering via performance monitoring
Cache timing attack	Learn correlation between memory access & time
Microcode attacks	Reprogram the machine code functionality

Table 6.1 – SGX is vulnerable against above hardware attacks

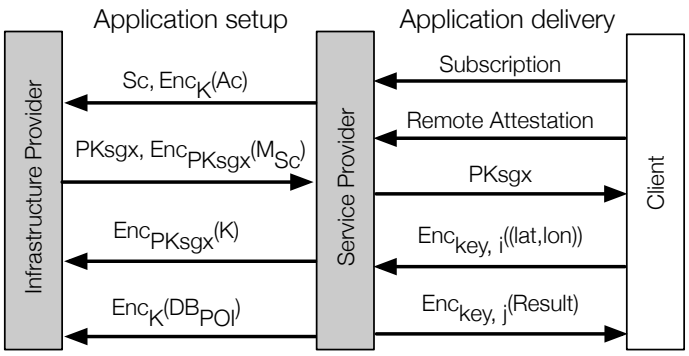


Figure 6.5 – Communication protocol in POI-Locator

SGX does not offer implicit protection. We do not consider these attacks, as they require physical access to the hardware, which is easier to detect in most cases.

6.3.3 System Design

Our system and protocol design focuses on the computation and communication security between all the entities involved.

Application Setup Phase. In order to setup and launch the POI-Locator application, the SP first transfers a setup code, *Sc* to the IP. *Sc* is not confidential and can be sent in plain text. Next, the application code, *Ac* is encrypted with a confidential key, *K* and sent to the IP. Upon reception, IP will setup and instantiate an enclave and run the *Sc*. After running the *Sc*, a log called enclave's measurement, M_{Sc} is created, which attests that *Sc* is running in isolation within a legitimate enclave. This log is encrypted using the public key of the SGX core, *PKsgx* and sent to the SP. The SP verifies the log with the provisioning server and sends the key *K* encrypted with *PKsgx* back to the IP. After receiving the *K*, *Sc* can decrypt *Ac* and can initiate the application inside the enclave. The interaction between the SP and the IP is depicted in Figure 6.5.

Next, the database file containing the POI's is sent to the IP, encrypted with *K*. This file is later decrypted inside the enclave and is sealed using the enclaves TweetNaCl keypair[3].

[3]TweetNaCl: https://tweetnacl.cr.yp.to/

Service Provisioning. Once the service is running, and before serving the clients the enclave generates TweetNaCl keypairs . Upon the successful generation of keypairs, the enclave outputs a public key that is provided to the client after subscribing to the service. This key can be used by the client to encrypt the requests and authenticate the results.

First, the client runs the remote attestation service to verify that the SP is running the promised program securely. The client views it as a public-key certificate, where the SP, along with the Intel provisioning server, endorses the application. After this step, both the parties have the enclave public key, $PKsgx$. We rely on the SGX re-encrypt[4] mechanism to establish a secure communication protocol between the client and the SP. In order to gain access to the service, the client sends a request with the current location and range encrypted with a key ID, i. Along with this message, the client also sends a key ID, j: it is the encryption key ID of the result to be returned by the SP.

The SP receives the request and decrypts the ciphertext using the key ID, i, providing the user location coordinates (lat, lon) and the range. The application then retrieves all the POI's lying within the range of the user's location, encrypts it using the key ID j and sends the result to the client. To view the result, the client decrypts the it using the the key j. All the plaintexts are encrypted using AES-GCM with 128-bit keys[5] and elliptic curve schemes over p256, which provides 128-bit security. In order to preserve anonymity, we rely on the anonymous routing component, Tor[6] at the client's end. This is simply achieved by using a Onion Proxy mobile client [7] to connect to the service provider; this mobile client uses a type of source routing to achieve communication anonymity between the client and the SP.

6.4 Evaluation and Results

To evaluate the system performance, we base our results on a database of points of interest in Switzerland retrieved from the Open Street Maps[8]. Our implementation is run on a 64-bit, Intel 4-Core i5-7500 CPU clocking at 3.40GHz and running Ubuntu-16.04. We use the Linux 2016-06 SGX SDK[9], and the Enclave Page Cache was set to the maximum available size of 128 MB.

6.4.1 Benchmarking SGX Overhead

In order to quantify the overhead involved due to the SGX, we benchmark the latency of basic enclave operations. The execution time of enclave creation, enclave entry and exit (ECall and OCall), encryption, generating the keypairs, measurements and tokens, copying and sealing

[4]SGX re-encrypt: github.com/kudelskisecurity/sgx-reencrypt
[5]AES-GCM: tools.ietf.org/html/rfc5084
[6]TOR Proxy: www.torproject.org
[7]Onion Proxy: www.torproject.org/docs/android.html.en
[8]OSM Switzerland: planet.osm.ch
[9]Linux SGX-SDK: https://github.com/01org/linux-sgx

Enclave Task	Execution Time	Enclave Task	Execution Time
Create	$22.41\mu sec$	Copy (128 Bytes)	$0.155\mu sec$
Entry	$0.752\mu sec$	Seal (128 Bytes)	$0.137\mu sec$
Exit	$0.631\mu sec$	Keypair	$13.445\mu sec$
Encrypt (128 Bytes)	$0.0154\mu sec$	Hash (128 Bytes)	$0.264\mu sec$
Token	$24.9944\mu sec$	Quote	$15.39\mu sec$

Table 6.2 – Micro-benchmarks of enclave tasks

data is shown in Table 6.2. All the results are derived after taking into account the average and variance over 100 runs. We rely on SGX-log [119] to implement and quantify the latency of the micro-benchmarks.

The latency due to the enclave creation, copying and sealing is a one time cost involved during the service initialization phase. Every new client also has to bear the initiation cost of retrieving the measurement quote and generating the SGX public keypair. The other tasks, such as encryption, ECalls and OCalls, are recurring costs and contribute to the core of the overhead involved due to the SGX.

6.4.2 Bare-Metal Comparison

Here, we compare the overhead contributed by the SGX to the bare-metal implementation of the same application. We select a random coordinate lying within the POI dataset and select a range of 1000 meters in the query. These two parameters are kept constant for this evaluation. We quantify the overhead in terms of number of total instructions executed as the size of the POI-dataset increases, as shown in Figure 6.6. SGX results in a modest 10-12% rise in the number of total executed instructions. A majority of these additional instructions result due to transferring the execution between the enclave and the non-enclave modules. More specifically, the OCalls that the enclave has to initiate in order to execute system calls. Additional overhead is contributed by the instructions that need to be executed to encrypt and decrypt the array that contains the result and the user's location. However, these costs are marginal and do not lead to noticeable service delays.

6.4.3 Precision Comparison

Next, we compare our SGX-based approach to a popular location-privacy preserving technique: spatial cloaking with k-anonymity. The central idea of cloaking is to perturbate and anonymize the user's true location by creating cloaked regions. Spatial cloaking typically requires a trusted third party, called a location anonymizer, responsible for generating the cloaked regions. The anonymizer has to ensure that the cloaked region contains the number of users greater than or

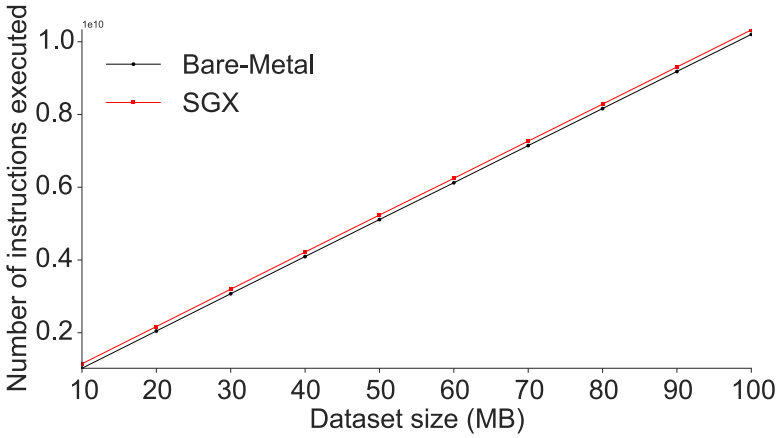


Figure 6.6 – Comparison of number of instructions executed

equal to *k*. Here, k refers to a privacy parameter that can be chosen by the user and corresponds to the desired degree of privacy.

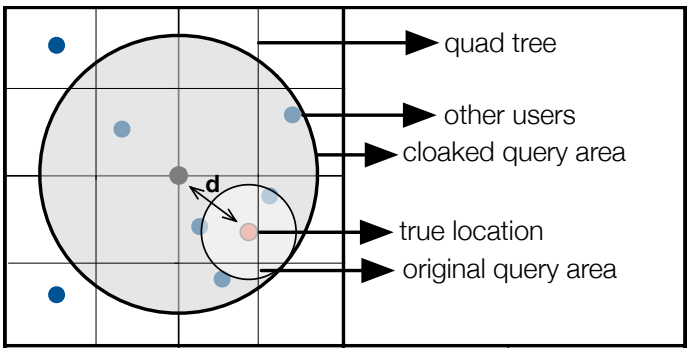


Figure 6.7 – Spatial cloaking with k-anonymity

The POI-Locator application can be queried with a location and a desired range. Hence, for comparison, we assume a scenario wherein a set of queries are performed by different users who lie within the POI-dataset range. The cloaking implementation for this experiment is based on a simple spatial perturbation with k-anonymity technique described in [96]. The algorithm first indexes the locations of all the users in a quadtree. Given the location of a user, it then searches for the first cell that contains this location and less than k queries. The parent of this cell is guaranteed to contain a number of queries greater than or equal to k and is returned as the cloaked region. The algorithm then computes a new range by adding the distance between the initial location and the center of the cloaked region, *d* to the initial range specified by the user as shown in Figure 6.7. Thus, the center of the cloaked region and the cloaked range is used to send the query to the SP. The anonymizer does a good job at cloaking the user-location and range, however, this comes at a great cost in terms of precision.

In Figure 6.8, we show the relationship between the actual user query range and the cloaked query range as *k* increases. We consider three query ranges: 100, 500 and 1000 meters and

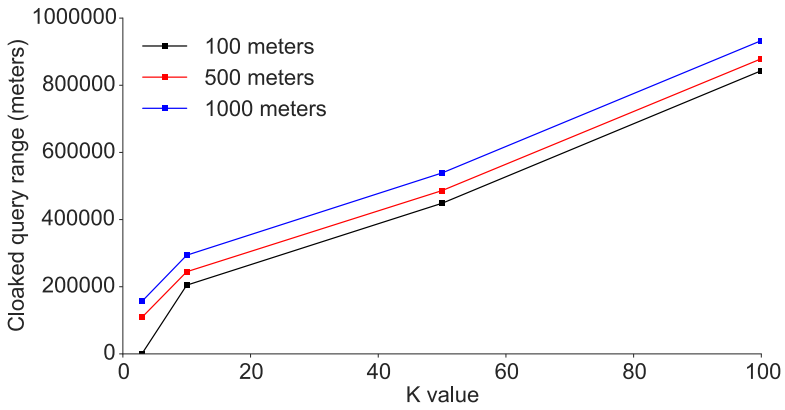


Figure 6.8 – Relationship between query to clock range with k

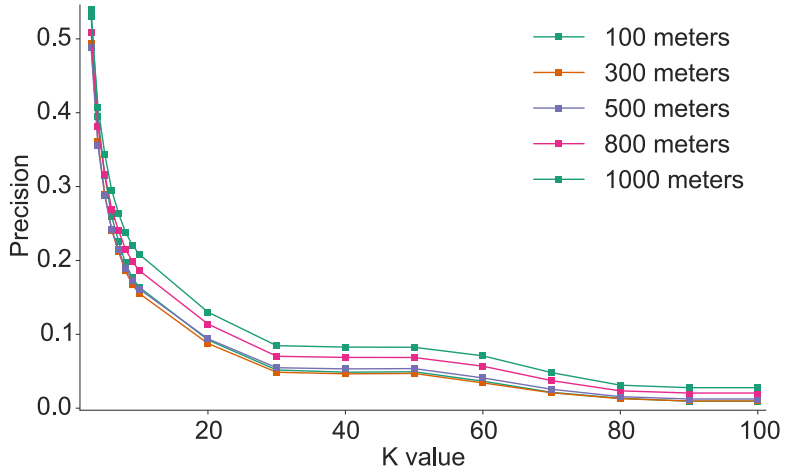


Figure 6.9 – Relationship between the result precision and k

as seen, the difference between the two significantly increases with k and leads to imprecise results. Furthermore, we observe in Figure 6.9 that as the privacy requirement (k) increases the precision lowers significantly with different query ranges. In this case, we define precision as the ratio between the number of POI's lying within the original range specified by the user from her true location ($true\ positives$) to the number of POI's retrieved by the SP ($true\ positives + false\ positives$). Note that measuring the effect of cloaking on recall is not relevant because the original queries and the cloaked queries return all the relevant results.

In conclusion, an approach based on SGX presents a clear advantage over an approach based on spatial cloaking with k-anonymity. Low precision has a great impact on the number of results returned by the LBS. This translates to higher computational and bandwidth requirements. In contrast, being able to return highly precise results, and guaranteeing privacy can make the LBS much more efficient.

6.5 Conclusion and Future Work

In this paper, we have demonstrated the applicability of a hardware-based trusted execution-environment, i.e. Intel SGX to offer a privacy preserving location-based service. We implement a POI-Locator application using the security guarantees offered by SGX, adopting a privacy-by-design principle. We quantify the overheads involved due to the SGX implementation and compare it with the bare-metal execution. We show that SGX-based approach leads to a marginal overhead and provides near-to-the-perfect results. We experimentally show that SGX is a better alternative compared to popular location-privacy preserving approach: spatial-cloaking with k-anonymity, which has a detrimental impact on the precision as the degree of privacy increases.

Our current work, focuses on safeguarding only the user from the service provider. However, the user can retrieve the complete dataset from the service providers by submitting a large number of queries. This is critical when the services provider hosts a privacy-sensitive database. Our future work will address this issue in order to guarantee the privacy of both the parties involved. Furthermore, we will perform a complete evaluation of the SGX-based service, including the memory efficiency, responsiveness and usability to the clients, the number of simultaneous queries that can be handled and the network performance. We will also compare this approach with other well-known privacy-preserving approaches such as Private Information Retrieval (PIR) [49], in terms of accuracy and overheads.

Capturing Human-Mobility Part III

7 Generating Synthetic Mobility Traffic Using RNNs

Abstract

Mobility trajectory datasets are fundamental for system evaluation and experimental reproducibility. Privacy concerns today, however, have restricted sharing of such datasets. This has led to the development of synthetic traffic generators, which simulate moving entities to create pseudo-realistic trajectory datasets. Existing work on traffic generation, superficially matches *a-priori* modeled mobility characteristics, which lacks realism and does not capture the substantive properties of human mobility. Critical applications however, require data that contains these complex, candid and hidden mobility patterns. To this end, we investigate the effectiveness of Recurrent Neural Networks (RNN) to learn these hidden patterns contained in an original dataset to produce a realistic synthetic dataset. We observe that, the ability of RNNs to learn and model problems over sequential data having long-term temporal dependencies is ideal for capturing the inherent properties of location traces. Additionally, the lack of intuitive high-level spatiotemporal structure and instability, guarantees trajectories that are different from the ones seen in the training dataset. Our preliminary evaluation results show that, our model effectively captures the sleep cycles and stay-points commonly observed in the considered training dataset, along with preserving the statistical characteristics and probability distributions of the movement transitions. Although, many questions remain to be answered, we show that generating synthetic traffic by learning the innate structure of human mobility through RNNs is a promising approach.

Keywords: Synthetic mobility traffic; Mobility behavior; Recurrent neural networks.

7.1 Introduction

The pervasiveness of mobile devices equipped with internet connectivity and global-positioning (GPS) functionality has resulted in the collection of large volumes of mobility trajectory data of individuals. This data is used for a variety of applications such as designing and evaluating systems aimed at mobility prediction, urban planning, consumer profiling and traffic management. However, sharing such datasets with untrusted third parties have several privacy implications. Simple heuristics can be applied on such datasets to derive personally identifiable information (PII) of users for blackmailing or stalking purposes [129]. Furthermore, data breaches, unlawful data exchanges and security vulnerabilities has restricted sharing of such datasets for development and research purposes.

This has led to the usage of synthetic traffic generators that simulate or mimic the behavior of moving entities. However, existing traffic generators rely on deterministic models having predetermined movement distribution, which fails to capture the behavioral realism. The Brinkoff data generator [30] uses the road network and a perturbation model to generate mobility traces. The BerlinMod Traffic Generator [58] relies on the Berlin road network and the Secondo DBMS [1] to generate data. MNTG traffic generator [164] provides a web-based road network trajectory generator, which is based on Brinkhoff and BerlinMod movement models. A recent trajectory generator, called Hermoupolis [187], uses existing trajectory datasets to generate a larger synthetic dataset. The underlying idea of this model is to extract semantic data from the raw data which is then used to construct a realistic user behavioral model. To generate synthetic data, such generators pick new sets of semantics and use the extracted mobility behaviors to generate trajectories. Such approaches result in purely discriminative behavioral models, i.e., the conditional probability distribution of a data point is learnt according to another point. Although, such models are suitable for generating datasets that address use cases such as trajectory indexing, they lack realism specially in capturing human behavior, which expands beyond modeling the semantics, trip-based and trajectory-based movements. For example, a user can showcase complex routes to travel from point A to B, vary the wake up-sleep and weekend cycle or change behavior to visit some places depending on certain external factors. These changes are critical for applications such as consumer profiling and behavioral analysis.

To this end, we present a synthetic traffic generator that uses machine learning, i.e., recurrent neural networks (RNN) for extracting the substantive behavioral patterns of users from actual datasets. We then use this trained model to create new and larger datasets, characterized by features that resemble the true properties of users from an actual dataset. Our approach combines the discriminative model with the generative model to learn the joint probability distribution of a dataset. Training in this manner, restricts the output trajectories to the bounds derived from the dataset, however, the generative model results in producing new trajectory sets. In addition to generating synthetic traffic, the trained model can be used to capture the generalized mobility patterns in a given area, which may include the frequently visited places, commonly followed

[1] Secondo DBMS: http://dna.fernuni-hagen.de/secondo/

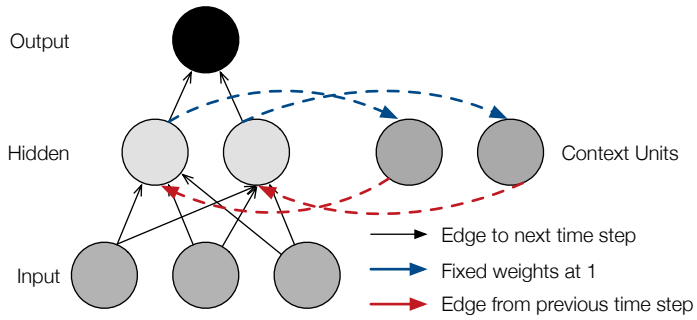


Figure 7.1 – A recurrent neural network architecture. Hidden layer is connected to the context units which feeds back into the hidden layer at the subsequent time step.

trajectories and transportation modes utilized.

The rest of this paper is organized as follows. Section 7.2 presents the necessary background knowledge about RNNs to illustrate our system design. The related literature and the associated shortcomings are discussed in Section 7.2.1. We present our system model in Section 7.3. The evaluation results are presented in Section 7.4. Finally, the conclusion and future work is presented in Section 7.5.

7.2 Recurrent Neural Networks

A typical feed forward neural network has connections from layer n to layer $n+1$. A key determinant, differentiating RNN is the connection from layer n to layer n in addition to the regular connections as shown in Figure 7.1. Such loops enable the network to compute on data from previous cycles, creating a network memory. This influences the network predictions to be influenced by the past values making it ideal to learn a sequence. The length of the network memory is not indefinite and gradually degrades with older information being less relevant. A drawback of RNN is that it suffers from the vanishing gradient problem, which also hinders remembering the past inputs. In order to address this, long short term memory (LSTM) is used which also bridges the long time lags between the inputs. This capability is used to learn sequential data using the recurrent connections between their neural activations at consecutive time steps. For a given input x_t at a time step t the network creates a hidden state h_t, such that it is a non-linear function of the previous hidden state h_{t-1}.

RNNs read through a sequence iteratively, preserving the structure in the model. It goes through each element of the sequence and updates its representation based on that item and the input from the previous state. At each time-step n, the number of hidden unit dimensional vector represents the input sequence. The connections between the hidden units and their respective projections are preserved making learning tractable. Gating units are often utilized in a RNN model, to transform the information flow in a more structured manner. They also control the proportion of the past information which should go forward and models the network to adaptively forget information.

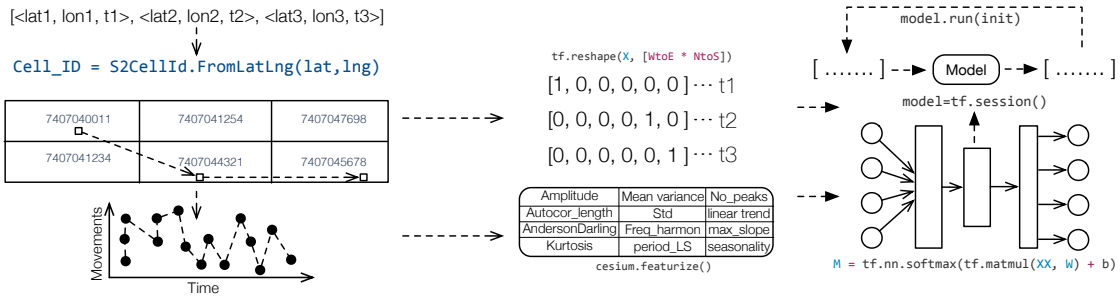


Figure 7.2 – Model training and trajectory generation. The coordinates are mapped on to a grid which are than one hot encoded. Feature exploration is performed on the discretized movements. The extracted features and the vectors are then used for training. The formulated model is instantiated with a sequence of grids to initiate the trajectory generation phase.

The model's remembrance is controlled by averaging the previous hidden state output with the current output. A crucial factor determining RNN suitability is the dataset size, as RNNs have poor generalization properties on small dataset volumes [151].

The effectiveness of RNN to learn the patterns on sequential data has been successfully applied for text generation [177], image captioning [158] and action recognition [214]. Sutskever et al. [232] showed that, when provided with a limited set of vocabulary, RNN outperforms a machine translation system with an unlimited vocabulary on a large scale machine translation task. Our motivation to employ LSTM-based RNN to generate mobility trajectories without making any assumptions regarding the problem structure is based on the successful application of RNN for sequence learning problems.

7.2.1 Related Work

A majority of the existing techniques to generate synthetic traces are based on trajectory modeling. Jian et al. [114] characterizes mobility as goal oriented in terms of Levy flight behavior attributed to the underlying street network. Here, trajectory is modeled in terms of several flights between the underlying street network, with purposive destination locations. The effect of space on the movement patterns is taken into account by modeling the length and the frequency of a flight according to the power-law distribution. Ghosh et al. [83] model trajectories by finding correlation between user-place, place-place and user-user from the GPS traces. These correlations are then used to form a temporal node-based graph structure for user's trajectory. Kim et al. [123] model mobility, characterized by the speed and pause time of the movements, which follow a log-normal distribution. Using the above techniques, the problem of modeling human mobility becomes quite challenging for unpredictable behaviors.

Another category of techniques rely on well established mobility models of moving objects. Random walk is one of such approaches in which the path of a mathematical object is modeled as a succession of random steps on an arbitrary mathematical space. In the case of random waypoint

and random direction model, the movement of mobile users is characterized according to the changes in their location/direction depending on random changes in velocity and acceleration over time. In the truncated Levy walk model, mobility is modeled to follow truncated power-law and further constrained to geographical features such as walk boundary, obstructions and traffic. In the above techniques, the mobile entities can stop suddenly or turn very sharply, failing to capture the true movement patterns of mobile objects. To eliminate such behaviors, Jean-Daniel et al. propose Gauss-Markov (GM) mobility model [28] , to limit the sudden stops and turns within specific regions. In the reference point group mobility model (RPMG) [109], the relationships between different mobile objects is considered to generate synthetic traces as a group of entities. In the above models, the speed and the direction of movement at a new timestamp has no relation to the past locations, furthermore the mobility models are based on stochastic processes and do not truly reflect the realistic mobility characteristics. Furthermore, these approaches result in creation of mobile objects at the same locations in a periodic manner due to the use of bounding parameters.

7.3 System Model

Problem Statement: Given a dataset of trajectories belonging to actual moving subjects, $t_r = \langle u_1, u_2 ... u_n \rangle$, such that $u_i = \langle ..., s_i ... \rangle$ is a trajectory of a user u_i and each point s_i is a three item tuple, (lat_i, lon_i, t_i), where lat and lon are the latitude and longitude coordinates and t is the timestamp, our goal is to extract and learn the user mobility behaviors, in a way that facilitates generation of synthetic but realistic mobility traffic data of a fictional subject $f_i = \langle ..., fs_i ... \rangle$ such that fs_i is also a three item tuple containing the latitude, longitude and timestamp generated by the trained model.

System Design: We base our system design on Long Short-term Memory(LSTM) recurrent neural networks(RNN), by configuring the network model to learn convoluted sequences and extend it to formulate predictions in the spatiotemporal domain [92]. The network with an attention mechanism is trained using actual user trajectories to make it deterministic and follow road networks or other valid paths taken by the legitimate users. Training the model by shuffling the user trajectories, incorporates stochasticity and fuzziness in the model [99], impacting its probabilistic output distribution, which leads to generation of novel trajectory sets. Finally, the complete trajectory sequences can be generated by iteratively feeding the current output trajectory sequence as input to the next step to the trained model, starting at some arbitrary location.

Implementation: Our implementation is based on the TensorFlow library[2] and is depicted in Figure 7.2. In order to construct the model, we first discretize the space by mapping the coordinates to grids by using the Google S2 library[3] for dimensionality reduction. We configure the S2 library to map each coordinate pair to a cell of dimension $38m^2$. This choice is motivated by the localization accuracy of a typical GPS sensor and the performance complexity involved

[2]TensorFlow: www.tensorflow.org

[3]Google S2: pypi.python.org/pypi/s2sphere/

when subdividing the cells to the leaf level. The cell ID's and the timestamp's are one hot encoded and the resulting vectors are fed as inputs to the network. The network is updated at every instant which enables the next movements to be dependent on the recently seen inputs. In order to bound the outputs, we extract features from the input trajectories which ensures that the movement properties are preserved in the synthetically generated traces. Some of the features include the amplitude of movement which captures the difference between the minimum and maximum magnitude of the movements, the autocorrelation length which captures the periodicity, mean variance etc. In order to compute the features, we use the Cesium library[4], which is used to featurize time-series data.

Challenges: We observe that, batch training such a network, with a dataset of around 191 users for roughly 10,000 epochs is sufficient to capture the basic pattern of mobility behavior including the sleep and wake up cycles and the visitation patterns amongst commonly visited places in the area under consideration. However, we observe that, the model does not preserve the ordering of the commonly visited places and the associated transitions between them. We argue that, it can be addressed by selecting the appropriate features which preserve the structure of the trajectories. Training the model with fine-grained features can eliminate such problems. Along with the above issue, the training process presents some interesting challenges, mainly in selecting the size of the network, amount of memory, dealing with the instability while generating trajectories, amount of noise to be injected to increase the models robustness and bounding the outputs by the properties of the real users. We will address such aspects in our future work.

7.4 Evaluation

The model training and evaluation results are based on the Nokia Mobile Dataset [124]. It consists of mobility traces of 191 users collected in Switzerland over a period of two years. We first examine the matching of the generated traces to the road network with respect to the number of trajectories and training epochs. As seen in Figure 7.3, after 60,000 epochs the model learns the paths typically adopted by the moving objects and starts replicating it. We also observe that, the model learns the common points where the objects stop and the stoppage durations while moving.

Next, we evaluate the prediction accuracy of the network. In order to compute it, we first extract the most frequent transitions between the hotspots in the dataset. We observe that, as the number of users considered in the training phase increases, the models next place prediction accuracy increases. However, on the contrary, the prediction accuracy of the next trajectory decreases. This is crucial to validate the generalization property of the model over the training dataset. Our future work will adopt metrics such as negative log-likelihood to quantify such accuracies.

Further, to validate the similarity of the trajectories to the actual user behaviors, we compute the same features used to train the model on the generated traces. We observe that, the model is able to learn the sleep and wake up cycles, movement periodicity and the variance in the movement

[4]Cesium: cesium-ml.org

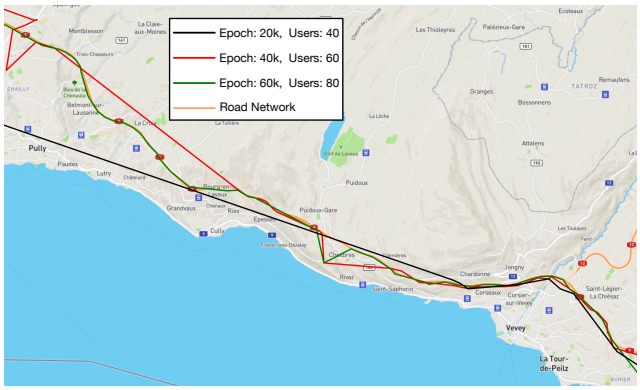
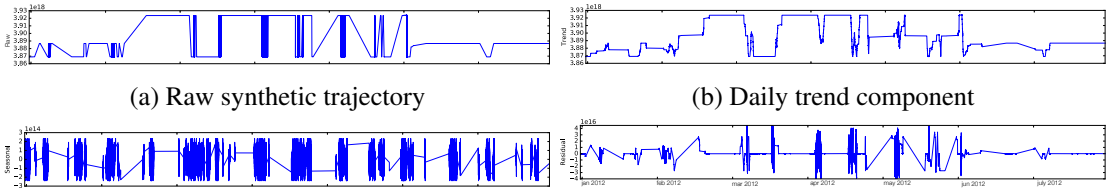


Figure 7.3 – Road network matching accuracy with respect to the the number of users and the number of training epochs.



(a) Raw synthetic trajectory

(b) Daily trend component

(c) Seasonal trend component

(d) Periodic trend component

Figure 7.4 – Seasonal trend decomposition of the synthetic trajectory.

distance magnitudes. We also calculate the season trend decomposition of a generated trace as shown in Figure 7.4. Although, the model preserves the weekday and weekend patterns, we observe some gaps in trajectories and some incoherent movement transitions. We argue that such abnormalities can be addressed by using a Bi-directional RNN and echo state network (ESN). The applicability of such mechanisms and validating the generated trajectories with additional mobility quantifiers such as movement entropy, number of visited places, visitation frequency, evolution in the visited places with time, etc. will be addressed in our future work. We will also validate the generated data points by performing nonlinear logistic regression to discriminate between the actual and synthetic data [99].

7.5 Conclusion

In this paper, we have highlighted the necessity of devising generative models for traffic data generation that outreach the limitations of the existing discriminative models. Our proposed approach uses a long short-term memory based recurrent neural network for generating new data on the basis of an existing dataset. The generated data is suitable for uses cases such as mobility prediction and behavioral analysis. Our future work will focus on evaluating the statistical validity of the generated data and on the challenge of transposing the learnt behavioral model to new areas by applying transfer learning. Although the preliminary results are promising, many questions remain to be answered such as; (1) relationship between the network model and

dataset realism, dataset magnitude and realism (2) mapping the knowledge from a known region to another unknown (target) region and use this knowledge to categorize the users in the target region (3) privacy measures to prevent backtracking of original mobility traces or reproducing the mobility pattern of any individual (4) quantifying the realism in generated traces. In order to add stochasticity in the training data, a part of our future work is to collect data from sources which do not have predictable/repetitive behavior. Our target subjects include students on an exchange program and short-term tourists. We believe that such measures can address the problem of generating datasets having homogenous distribution.

8 Generative Models for Simulating Mobility Trajectories

Abstract

Mobility datasets are fundamental for evaluating algorithms pertaining to geographic information systems and facilitating experimental reproducibility. But privacy implications restrict sharing such datasets, as even aggregated location-data is vulnerable to membership inference attacks. Current synthetic mobility dataset generators attempt to superficially match *a priori* modeled mobility characteristics which do not accurately reflect the real-world characteristics. Modeling human mobility to generate synthetic yet *semantically and statistically realistic* trajectories is therefore crucial for publishing trajectory datasets having satisfactory utility level *while preserving user privacy*. Specifically, long-range dependencies inherent to human mobility are challenging to capture with both discriminative and generative models. In this paper, we benchmark the performance of recurrent neural architectures (RNNs), generative adversarial networks (GANs) and nonparametric copulas to generate synthetic mobility traces. We evaluate the generated trajectories with respect to their geographic and semantic similarity, circadian rhythms, long-range dependencies, training and generation time. We also include two sample tests to assess statistical similarity between the observed and simulated distributions, and we analyze the privacy tradeoffs with respect to membership inference and location-sequence attacks.

8.1 Introduction

The pervasiveness of mobile devices equipped with internet connectivity and global-positioning functionality has resulted in an increasingly large amount of location-data on individuals. This data is beneficial to address and validate spatiotemporal data-based problems; predictive and kNN queries, object tracking, mobility modeling and location privacy among others. Due to the sensitive nature of datasets containing mobility traces, sharing them with untrusted entities present privacy implications. Trivial heuristics can be applied on such datasets to derive personally

identifiable information of individuals, even at aggregate levels [251].

Publicly accessible mobility datasets [124, 167, 266] are usually not adequate for large scale experimental evaluations, compromising scalability tests. This issue incentivizes synthetic mobility trajectory generators that simulate the behavior of moving objects required to attain comprehensive performance valuations. In this context, one typically considers rigid and unnatural mobility models, not guaranteeing the existence or even cardinality of patterns within the synthetic population. Alternative approaches rely on parametric sequential models [132] and Markov processes [27] to learn and generate trajectories. Such techniques also ignore the presence of long-range dependencies[148] inherent to human mobility which features non-Markovian character [133, 265].

It is therefore imperative to generate context-dependent synthetic traces resembling the human-mobility behavior at satisfactory utility levels while preserving user privacy. However, one of the major challenges is the absence of quantitative methods for evaluating the realistic nature of synthetic traces and the associated utility-privacy tradeoff.

To this end, we present several nonparametric approaches to generate large-scale synthetic trajectories by training the models on a real-world dataset followed by hallucinating trajectories using the trained model. We perform an extensive evaluation of the generated trajectories by assessing their geographic and semantic similarity compared to the actual dataset. We use two sample metrics to obtain the statistical similarity between datasets. We then quantify the presence of long-range dependencies by computing the mutual-information decay and conduct privacy-leakage tests on the generated trajectories. We conclude with a discussion on appropriate strategies and applicable evaluation metrics based on our experimental results and tackle open questions and challenges.

8.2 Related Work

Table 8.1 provides a summary of existing trajectory generators, where they formulate the synthetic trajectory simulation as an optimization problem, solved by genetic algorithms under the constraint of *a priori* determined parameters. A fundamental issue is the selection and definition of the parameter space that controls the evolution of the moving objects. The stringent and classified network connections thus influence the realistic nature of the generated trajectories. In several cases, there is no correlation between the future direction of movement and the past locations. Repeated visits to a given location within a short span of time are also observed due to the bounding parameters. Therefore, the symbolic nature of these frameworks result in an implicit location-dependent context, which compromises the realistic nature of the generated activity patterns. To address these drawbacks associated with parametric modeling, [183] propose a GAN-based approach to generate trajectories, where the discriminator is based on a convolutional neural network (CNN) [142]. Similarly, we explore other deep learning architectures based on RNNs known to model sequential data better than CNNs [209]. We also investigate generative

Table 8.1 – Categorization of current approaches to generate synthetic trajectories and parameters.

Technique	Model name	Parameters considered
Free movement	GSTD [235]	statistical distributions (mean, skew, standard deviation)
	G-TERD [178]	speed, rotation-angle, direction
	Oporto [85]	start time, end time, velocity, orientation
Road networks	Brinkoff [31]	speed, street capacity, nearby object location, shortest path
	SUMO [19]	road length, headway time, lane change times
	BerlinMOD [58]	road network, trip start and end, Brinkoff model
	ST-ACTS [88]	Geo-dependency model
	Hermoupolis [187]	mobility pattern, road network, points of interest
Multi environments	MWGen [252]	trip plan, road network, floor plan, routing graph
	MNTG [165]	movement model, moving objects, simulation time
Sequential models	Markov models [27]	semantic locations, geographic
	Semi-Markov models [17]	stay points, transition paths

models based on the nonparametric copulas of [79].

8.3 Synthesizing Trajectories using Generative Modeling

First we explore the benefits of applying deep learning architectures to synthesize mobility trajectories. RNNs use their hidden memory representation to process input sequences and we select four architectures: (1) Char-RNN (SRNN) [94], (2) RNN-LSTM [108], (3) recurrent highway networks (RHN) [273], and (4) pointer sentinel mixture model (PSMM) [161]. For GANs, where two neural networks compete in a zero-sum game framework, we select two architectures: (1) SGAN [257], and (2) RGAN [63]. These architectures differ in their capacity to manipulate their internal memory representation and propagate gradients along the network. In addition to neural-network based solutions, we also evaluate *copulas*; a seldom explored generative model in the machine learning community. Given a bivariate random vector (X_1, X_2), Sklar's theorem [221] states that the joint density[1] is $f(x_1, x_2) = f_1(x_1) f_2(x_2) c(F_1(x_1), F_2(x_2))$, where f and f_i are the marginal densities, F_i the marginal distributions, and c the copula density. In other words, the bivariate density can be uniquely described by the product between its marginal densities and a copula density representing its dependence structure. A useful consequence of this representation is that, by taking the logarithm on both sides, estimation of the joint density can be performed in two steps: the marginal distributions first, and the copula afterwards. In a nutshell, copulas allow to flexibly specify the marginal and joint behavior of random variables.

An important aspect in the generative context is that, because $U = F(X) \sim U(0,1)$ for any continuous random variable with distribution F, the copula is a distribution with uniform margins. Hence, from a copula sample (U_1, U_2), one obtains a sample on the original scale using the inverse cumulative distributions via $(X_1, X_2) = (F_1^{-1}(U_1), F_2^{-1}(U_2))$. For further details on copulas, we refer the reader to [116]. In this paper, we combine the kernel-based nonparametric copulas of [79] with the empirical distribution function of the margins obtain highly flexible models.

[1] It is usually stated for the distribution rather than the density and for random vectors of arbitrary dimension.

Data representation Given a dataset of n mobility trajectories, where a trajectory T_u of an individual u is a temporally ordered sequence of tuples, such that, $T_u = \langle (l_1,t_1),(l_2,t_2)...(l_n,t_n)\rangle$, where $l_i = (lat_i, lon_i), 0 \leqslant i \leqslant n$, the latitude-longitude coordinate pair and t, the timestamp such that $t_{i+1} > t_i$. We first transform the location data onto a uniform grid for dimensionality reduction using a technique that preserves spatial locality[2], thus translating T_u into a 2-D trace $S(t) = \langle (c_1,t_1),(c_2,t_2)...(c_n,t_n)\rangle$, where c_i is the *geo-hash* of the projected cell ID and the timestamp t_i.

8.4 Experiments, Results and Discussion

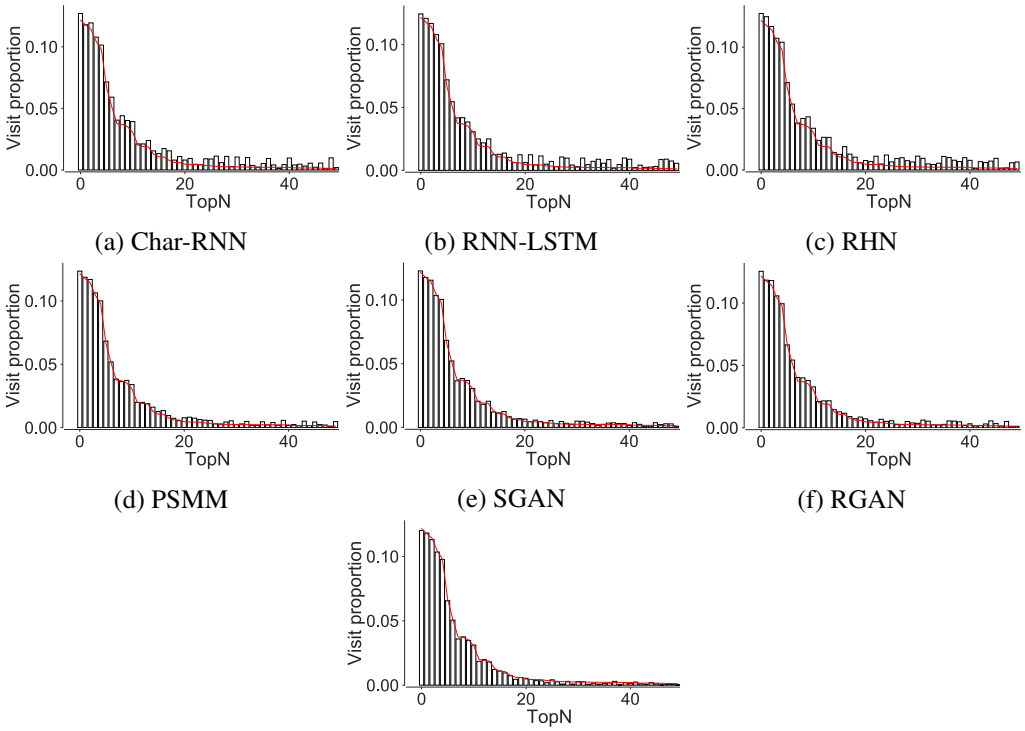
Experimental setup A complete trajectory sequence can be generated by iteratively feeding the current output trajectory sequence as input for the next step to the trained model. RNNs are trained on the geo-hashes and timestamps of all the individuals present in the dataset in a deterministic framework. GANs are first trained to model and then successively reproduce the traces in the same representation, which is mapped back to the (lat_i, lon_i) coordinates. We use the standard implementations of the predictive algorithms and hyper-parameters as described in their respective papers. To use copulas as generative models, we rely on the *rvinecopulib* package [176], whose `vine` routine implements the automatic kernel-based fitting of the dependence structure.

Dataset Experiments are performed using the Nokia mobile dataset [137] that consists of mobility trajectories of individuals collected in Switzerland. We use a total of 70M data points to train the considered models.

Evaluation We perform the evaluation of the generated trajectories using this dataset from four distinct dimensions: (1) geographic and semantic similarity, (2) statistical similarity (3) long-range dependencies and (4) privacy tests. In order to assess the geographic and semantic similarity, we compare the probability distribution of visiting $topN$ locations (visit-time and dwell-time) in the generated trajectories for each technique compared to the true dataset (see Figure 8.1). Char-RNN, RGAN and copulas have the closest fit to the true distribution indicating that the $topN$ locations are very well preserved in the respective synthetic datasets.

To evaluate the statistical similarity, we use Mean Maximum Discrepancy (MMD) [93] to test whether one can reject the null hypothesis that a synthetic sample has the same distribution as the data. MMD works by replacing the probability densities with embeddings that facilitate the computation of distances between distributions. Note that defining distance metrics in the context of time series data such as mobility trajectories is challenging due to the alignment concerns [63]. We thus consider the time axis for alignment as done by [63]. The results along with the training and generation time for each approach is shown in Table 8.2, where we observe that all approaches achieve similar results in terms of MMD, with copulas standing out with a lower value. We can thus infer that copulas can synthesize distributions with statistical characteristics closer to the observed ones. Regarding the computational efficiency, copulas require a fraction of the time

[2]Google S2: https://s2geometry.io/



(a) Char-RNN (b) RNN-LSTM (c) RHN

(d) PSMM (e) SGAN (f) RGAN

(g) Copula

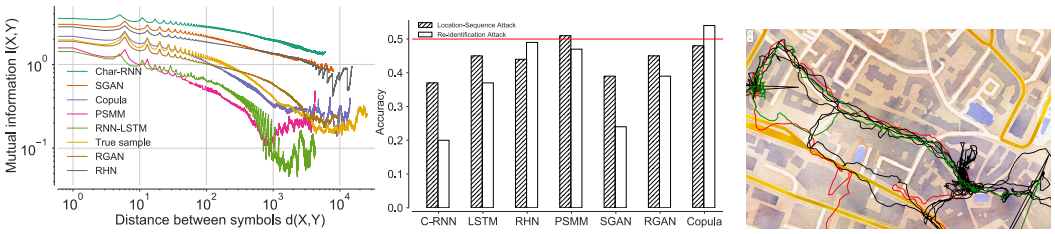
Figure 8.1 – TopN visited locations for real and synthetic trajectories generated by each method. We select $N = 50$ out of a total of 286 locations. The red curve shows the distribution for the true dataset.

Table 8.2 – Mean and standard deviation of real vs. synthetic data (lower is better) from 30 repetitions. Second row is CPU time indicating the training/fit+generation time.

Metric/Method	**Char-RNN**	**RNN-LSTM**	**RHN**	**PSMM**	**SGAN**	**RGAN**	**Copula**
MMD	0.32(1e-3)	0.27(9e-4))	0.30(1e-3)	0.21(6e-4)	0.19(7e-4)	0.21(6e-4)	0.01(6e-4)
CPU time (sec)	9k+∼10	10.3k+∼14	12.7k+∼15	10.5k+∼15	11.2k+∼15	11.5k+∼14	6.5 + 0.76

needed by NN-based approaches.

Figure 8.2(a) shows the result of long-range dependency test, in terms of mutual information decay [148, 157]. We observe a power-law decay in case of GANs, copulas and RNN-LSTM indicating that they account for the long-range dependencies in mobility trajectories. Figure 8.2(b) shows the results of two privacy tests: (1) location-sequence attack, and (2) membership interference attack. Given a synthetic dataset, (1) answers to what level of accuracy can trajectories in the dataset be reconstructed [217], and (2) an adversary's accuracy of inferring if a target individual contributed to the specific trajectory [201]. For these tests, we use the the location-privacy and mobility meter [217], where obfuscation is performed using the location hiding mechanism. Given a completely random distribution the accuracy of a recovered user-information is 0, we therefore suspect that the privacy-based score is biased towards representations which do not



(a) MI-decay (b) Privacy leakage (c) Sample trajectories

Figure 8.2 – (a) long-range dependency test (symbols denote individual location coordinates), (b) privacy test with location hiding as privacy preserving mechanism (red line indicates a random guess) (c) sample trajectories generated by two best approaches (copulas (black) and SGANs (red)) follow the road network for the most part and also synthesize stays at some locations indicating a point of interest. Trajectory from the actual dataset in the same area is depicted in green.

accurately capture the statistical properties of the true dataset.

8.5 Conclusion and Future Work

In this work, we propose and evaluate a variety of generative models to synthesize mobility trajectories. To the best of our knowledge, this is the first study to do so using seven different approaches while evaluating their realism across four dimensions. From the results and discussion, we observe that regarding statistical and semantic properties, copulas have an advantage over all other methods. Additionally, all NN-based methods are time consuming, which makes copulas favorable when computational efficiency is important to the end-users. As future work, we will consider datasets collected in bigger cities and generate larger synthetic datasets to evaluate the performance of these models under high movement stochasticity. From curbing the privacy leakage of the true dataset while maintaining utility, trajectory generation can be designed as an optimization problem with an objective to jointly maximize statistical similarity and privacy. But it is still not clear how to assess such property and adaptive/configurable metric, and is part of ongoing work [179]. While this paper represents an initial comparative study of various generative models, a deeper understanding of their performances will be needed to compute utility-privacy scores as applied to online services by [127] before publicly releasing the synthetic datasets. Another interesting avenue for research is to apply transfer learning in order to map a mobility behavioral model captured in one city on to another region.

9 Breadcrumbs: A Rich Mobility Dataset with Point of Interest Annotations

Abstract

In this paper, we present *Breadcrumbs*, a mobility dataset collected in the city of Lausanne (Switzerland) from multiple mobile phone sensors (GPS, WiFi, Bluetooth) from 81 users for a duration of 12 weeks. Currently available mobility datasets are restricted to geospatial information obtained through a single sensor at low spatiotemporal granularities. Furthermore, this passively collected data lacks ground-truth information regarding points of interest and their semantic labels. These features are critical in order to push the possibilities of geospatial data analysis towards analyzing mobility behaviors and movement patterns at a fine-grained scale. To this end, *Breadcrumbs* provides ground-truth and semantic labels for the points of interest of all the participants. The dataset also contains fine-grained demographic attributes, contact records, calendar events and social relationship tags between the participants. In order to demonstrate the significance of the ground-truth annotations, we discuss several use cases of this dataset. Furthermore, we compare four contrasting and widely used unsupervised clustering approaches for point of interest extraction from geolocation trajectories. Using the ground-truth information, we perform a detailed performance validation of these techniques and highlight their strengths and weaknesses. Given that mobility data is derived from an individuals inherent need of participating in activities, narrowing the gap between raw trajectory data points and complete trip annotation in essential. We thus make *Breadcrumbs* accessible to the research community in order to facilitate research in the direction of supervised human mobility learning schemes.

Keywords: mobility modeling; mobility prediction; mutual information.

Dataset	#Participants	Duration	#Events	sampling rate	Location	GPS	Check-ins	GSM	WiFi	Bluetooth	Annotation
GeoLife [267]	178	5.5 years	25M	5 sec	Beijing	✓	✗	✗	✗	✗	✗
MDC [125]	185	3 years	11M	-	Lausanne	✓	✗	✓	✓	✓	relationships
Privamov [166]	100	15 months	15M	-	Lyon	✓	✗	✓	✓	✗	✗
Reality Mining [190]	100	9 months	5M	-	Boston	✗	✗	✗	✗	✓	relationships
FourSquare [254]	3112	10 months	9M	-	New York	✗	✓	✗	✗	✗	relationships
blebeacon [220]	46	1 month	5M	-	California	✗	✗	✗	✗	✓	✗
hyccups [51]	72	63 days	-	-	Bucharest	✗	✗	✗	✓	✗	relationships
sigcomm2009 [195]	76	2 days	-	120 sec	Barcelona	✗	✗	✗	✓	✓	✗
telefonica [29]	342	4 weeks	-	-	Spain	✗	✗	✓	✗	✗	✗
ParticipAct [44]	300	1 year	-	-	Bologna	✓	✗	✗	✓	✓	✗
Nodobo [20]	27	4 months	5M	-	Glasgow	✗	✗	✓	✓	✗	✗
d4d challange [71]	9M	1 year	-	-	Senegal	✗	✗	✓	✗	✗	✗
Gowalla [45]	196,591	1.5 years	6M	-	Worldwide	✗	✓	✗	✗	✗	relationships
Brightkite [44]	58,228	1.5 years	4M	-	Worldwide	✗	✓	✗	✗	✗	relationships
Breadcrumbs	81	12 weeks	14M	50 sec	Lausanne	✓	✗	✗	✓	✓	ground truth semantic labels relationships

Table 9.1 – A descriptive summary of currently available and widely used geospatial mobility datasets and their features.

9.1 Introduction

The proliferation of GPS equipped smartphones and Internet connectivity has simplified the process of collecting positional data generated by moving entities. Spatiotemporal data streams (trajectories) generated by using people centric sensing technologies [33] can be stored and made available as mobility datasets. Modeling human mobility using such datasets is increasingly gaining importance as cities are experiencing rapid transformation and growth, which demands a good understanding of individual mobility behavior. Mobility datasets are therefore fundamental for designing and evaluating algorithms pertaining to Geographic Information Systems (GIS) and to facilitate experimental reproducibility. More specifically, the advancement of techniques addressing spatiotemporal data-based problems such as predictive queries [53], object tracking [246], trajectory indexing [36], mobility modeling [16] and location privacy [216] have transpired due to availability of several geospatial mobility datasets [125, 166, 190, 253, 254, 267].

While the publicly available mobility datasets have been widely used in GIS research to answer classical GIS research questions (movement analysis, trajectory indexing and queries), we highlight four critical limitations that stifle pushing the possibilities of geospatial data analysis further. These include: (1) lack of positioning information from multiple sensors, (2) unavailability of geolocation points at a high spatiotemporal granularity throughout the span of data collection duration, (3) lack of ground truth information regarding participant points of interest (POI), and (4) unavailability of semantic information regarding the POI. Datasets such as [190, 254, 255] are restricted to traces derived from a single sensor; either GPS, GSM, WiFi or Bluetooth. Having access to high granularity multi-sensor positioning data can lead complex and richer comparative and compositional studies [166]. Furthermore, since mobility data is derived from an individual's inherent need of participating in activities, lack of ground truth and semantic information is a critical limitation of available passively collected datasets. Currently, large amount of trajectory data is captured, but the associated activity information is poor, which is crucial for research domains such as social network pattern mining [46, 59], behavioral regularities [32], and behavioral entropy [134].

In this paper, we present a methodological description of geospatial data collection process to avail a dataset capturing multiple aspects of human mobility behavior. We present *Breadcrumbs*, a mobility dataset containing high granularity geolocation data points from GPS, WiFi, Bluetooth and accelerometer sensors from 81 individuals in Switzerland for a period of 12 weeks. We further enrich this dataset with point of interest (POI) ground-truth annotation and semantic labels along with demographic attributes, social relationships, calendar events and contact records. This information is especially important given that the last decade has seen an increasing demand of understanding the semantic behavior of moving objects in multiple sectors [146]. This refers to semantic abstractions of the raw mobility trajectories annotated with the knowledge extracted from the participants. Having access to the ground-truth, regarding the places where an individual actually spent time and has a meaningful accord with, is also crucial for corroborating the performance of data mining algorithms.

Along with the dataset description, we frame a research question that highlights the importance of a unique feature of *Breadcrumbs*, i.e., ground-truth information. Our research question is thus: Given the ground-truth information, which clustering approach and parameter settings provides the best sensitivity and specificity results? We perform a systematic comparison of four widely used spatiotemporal clustering approaches: (1) *k*-means, (1) DBSCAN, (2) DJ Cluster, and (4) DT Cluster. We also present our approach to perform accurate validation over the POI ground-truth labels and clusters extracted by different algorithms. We experimentally demonstrate how the ground-truth information can be used as labels to aid computing the ROC characteristics.

The remainder of the paper is organized as follows. We present a brief review of the existing mobility datasets and a summary of popular POI extraction techniques in Section 9.2. This Section also presents some use cases facilitated by the *Breadcrumbs* dataset. The data collection process is presented in Section 9.3 followed by the quantitative analysis in Section 9.4. Then, we present a comparison of four clustering algorithms and precisely describe the way we evaluate them in an evaluation framework in Section 9.5. We finally conclude the paper in Section 9.6.

9.2 Related Work & Use Cases

In this section, we review the existing mobility datasets and list their features. We also present and summarize the spatiotemporal clustering techniques used to extract POIs from the geospatial mobility datasets. Finally, we highlight some of use cases and fields of research that could benefit from the *Breadcrumbs* dataset.

9.2.1 Mobility Datasets and Applications

Geolocation mobility datasets mainly contain passively collected positioning data points that form a trajectory [20, 71, 166, 195, 267]. Such datasets focussing explicitly on geo-positioning information have been extensively used in several research domains such as discovery of points

of interests [244], computing trajectory similarity [141] and designing frameworks for processing trajectory queries [261]. Another domain of geospatial data research studies the interaction between human mobility behaviors and social relationships. This was driven by several datasets that provide annotated social relationships with spatiotemporal data points [45, 125, 254]. Such annotated datasets have been used to estimate similarity between users based on their location histories and infer potential social ties. The performance of the framework was validated against the ground-truth about social relationship collected from the participants.

The above trends have led to a new domain in location privacy, that focusses on inferring social relationships from mobility datasets. This task also involves formulating novel inference attacks and improved threat models [13]. Mobility datasets containing demographic information [44, 45, 254] have also contributed to location privacy research, wherein these datasets have been used to construct attacks to infer sensitive demographic attributes [269]. Furthermore, call detail record datasets with home and work place labels have been used to construct attacks against aggregated mobility datasets [251].

In Table 9.1, we present a summary of currently available mobility datasets along with the associated features and the data types. We observe that majority of the datasets are restricted to positioning data from either one or two sensors and does not provide demographic attributes or ground-truth information. MDC dataset offers basic demographic information including the participants sex, age group and gender. However this information is not available for all the users of the dataset. Contrary to the existing datasets, we ensure collection of equivalent data points for each participant, ensuring a satisfactory users to data points ratio. *Breadcrumbs* also provides ground-truth annotations and exhaustive demographic attributes as compared to existing mobility datasets as highlighted in Table 9.1, necessary to push the boundaries of current research. This information is critical in order to identify attack vectors and proactively fix the vulnerabilities before releasing user information. These efforts by the research community have resulted in devising improved strategies for anonymizing and aggregation of user location data [13]. Such efforts are also necessary to offer a satisfactory trade-off between utility and privacy of trajectory data [263]. Along these lines, we argue that the annotations provided by the *Breadcrumbs* dataset with regards to POI ground truth and semantic labelling will foster improvement in constructing such utility/privacy measures. For instance, formulating a probabilistic model for obfuscation mechanisms, where only the actual points of interests are obfuscated to to maintain a satisfactory trade-off [62].

9.2.2 Point of Interest Extraction

The seminal work in adopting data mining and clustering techniques for spatiotemporal POI extracting was proposed by Ashbrook et al. [12], where they presented an iterative approach to extract clusters while imposing spatiotemporal bounds. Then, a two-level clustering approach was proposed by Montoliu et al. [171] wherein the spatiotemporal trajectories are first clustered in the time domain and then in the spatial domain to detect stay-points and successively extract

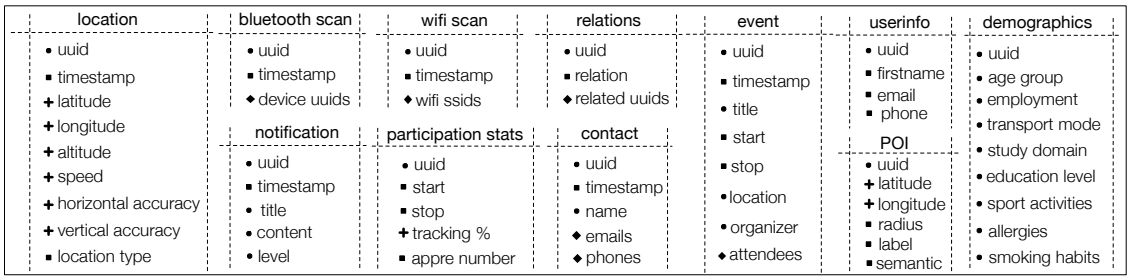


Figure 9.1 – Breadcrumbs database schema.

stay-regions. Several density based clustering approaches were later proposed such as Density-Joinable clustering [271], Density-Time clustering [102], Time-Density Clustering [75] and ZOI detect [136]. These approaches use several parameters to perform clustering in order to extract the POIs. These parameters include maximum/minimum distance/time between the trajectory data points, cluster shape, maximum number of data points per cluster among others. Kulkarni et al. [135] proposed a parameter-less technique for extracting POIs from spatiotemporal trajectories without any *a-priori* assumptions. In this paper, we use a clustering algorithm based on Density-Time clustering (see Section 9.3.2) to identify hotspots in the dataset participants trajectories. We then validated these hotspots through the participants to construct the ground-truth. Using this ground-truth information, we compare the performance of clustering technique described in this section.

9.2.3 Research Areas

In this section, we provide a non-exhaustive list of the research areas and application domains that might benefit from the features provided by *Breadcrumbs*.

Next-place Prediction. Given the current location of a user, next-place prediction aims at forecasting the place where the user will head next. POIs used in conjunction with statistical models, such as Markov chains, are known to address this problem reasonably well [75, 76]. Other approaches focused on Bayesian networks, neural networks and decision trees as detailed in [66, 240]. However a lot of research recently focused on leveraging recent findings in machine learning to improve next-place predictions. In this regard, the need to qualitatively compare prediction methods with each other, requires the access to the clusters extracted from the data and the ground truth associated with these clusters.

Trajectory Prediction. Given the current location of a user, trajectory prediction aims at forecasting the trajectory or path that the user will follow while heading to his next POI. More complex statistical models are required to perform such prediction and the historical GPS traces collected by the user are needed to create more accurate predictions [40]. In a way which is

similar to next-place prediction, the ground-truth is needed to qualitatively assess the forecasts.

Trajectory Indexing. The indexing and retrieval of trajectories and sub-trajectories is a central problem for a wide range of applications, such as car sharing, ride hailing, traffic forecasting, etc. Recent retrieval techniques focus on the ability to query by trajectories, i.e., the query itself is in the form of a trajectory and the result contain the most relevant matching trajectories of the dataset [38, 233]. As trajectory indexing can be affected by factors such as the quality of the GPS signal and its sampling rate, a realistic dataset is often necessary to test indexing mechanisms.

Synthetic Trajectory Generation. Location data is often considered as being sensitive. As a result, sharing them publicly comes with privacy implication and it is difficult to release large and dense datasets made of real trajectories. Therefore, a common practice consists in creating synthetic but realistic trajectories. Some models, such as BerlinMod [58], aim at generating trajectories for benchmarking spatiotemporal databases. More recently, machine learning as been used extensively to generate synthetic trajectories. In this regard, having access to sensor data, such as the accelerometer, can help at generating more realistic trajectories.

Privacy Preserving LBS. As mentioned, sharing location data is often associated with privacy concerns. For instance, can the demographic group to which a user belong be guessed on the basis of the data recoded by the sensors of his mobile phones? Having access to a dataset that gathers both location data and demographic data is key to devise and mitigate such inference attacks. Some research have been designed in this domain in order to infer demographics data by using the entropy level of an individual [175]. Other research works also tried to infer demographics information based on other types of data such as access points and location check-ins [248, 270]. Our dataset is clearly an added value because of the richness of the demographics information.

Health and Mobility Behavior. Several research studies have been done in the domain of health related to mobility behavior. For example, some researchers studies the mobility of senior individuals [70] and derived indicators based on GPS sensor data. Other researchers studied the influence of having a dog on the mobility of senior individuals [238]. Finally, [218] presented an analysis related to mobility behaviors in the context of cognitive diseases such as Alzheimer. With the locations of individuals and the additional data captured with the survey, it is possible to conduct health analysis combined with mobility patterns. Figures 9.12, 9.13, 9.14 and 9.15 illustrate four examples of research topics related to sport exercise frequency, seasonal allergies, smokers and diet respectively.

Supervised POI detection. User points of interest are currently extracted from mobility trajectories using clustering approaches that are unsupervised in nature. We propose a supervised POI

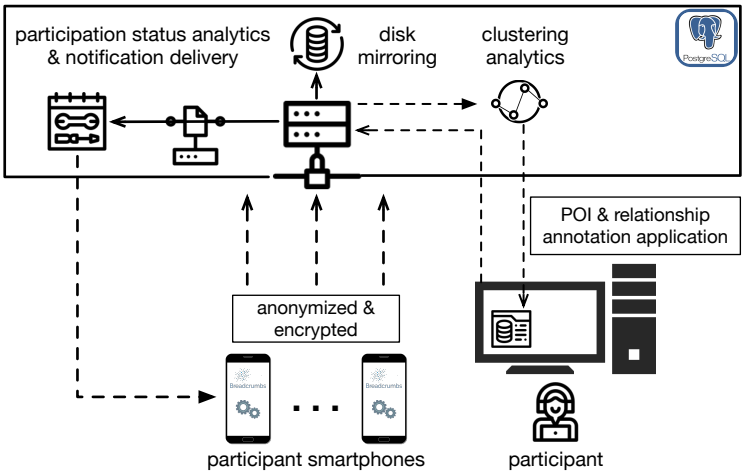


Figure 9.2 – Breadcrumbs system architecture.

detection approach, where the ground truth annotations serve as labels. In order to formalize the POI classification problem, the raw latitude, longitude and timestamps and passed through a time-series featurizer. The featurizer calculates a large number of trajectory characteristics/features such as autocorrelation, entropy, wavelet transform coefficients, number of peaks and crossings among others. This set of featurized trajectories can be trained on a binary classifier using the ground truth labels or the semantic labels to predict the next semantic place. Such a classifier can be used to asses the feature relevance contributing to a mobility trajectory terminating with a valid POI with a given semantic label.

9.3 Data Collection Method

The data collection campaign was designed to address the limitations of the currently available spatiotemporal mobility datasets. In this Section, we describe the design framework adopted to meet each of the limitation and the resulting tradeoffs. We group the limitations in two categories: (1) collecting high-granularity data points from multiple sensors, and (2) collecting ground-truth information about participant mobility. In order to address these limitations, *Breadcrumbs* data collection campaign aimed at collecting positioning data from GPS, WiFi, Bluetooth and accelerometer sensors, demographic information, calendar events and contact records, ground truth information labelled by the participants at the end of the campaign. The dataset contains the above information for 81 individuals, collected mainly in the city of Lausanne (Switzerland) for a period of 12 weeks. The participants include students from 5 different faculties from universities located in Switzerland and some of their relatives. Geolocation information is classified under Personally Identifiable Information (PII) by the EU privacy regulations (GDPR) [1]. The system architecture adopted for the *Breadcrumbs* data collection campaign is presented in Figure 9.2.

[1] European Union General Data Protection Regulation: https://eugdpr.org/

9.3.1 High Granularity of Multi-Sensor Data

The data was collected through a mobile application installed on participants' smartphones. The application was designed with an objective to be non-intrusive to the participant activity, while optimizing the data collection to power consumption tradeoff. The sampling rate was calibrated to meet the operational requirements for one day's smartphone usage and obtain a satisfactory granularity in the geolocation data points. To this end, we limited the sampling of GPS location and accelerometer reading only when the distance between the two location instances is five meters or greater. The Service Set Identifier (SSID) of the WiFi access point/s and device UUIDs of Bluetooth device/s scan was recorded instantaneously or through the periodic scans by the smartphones The data was stored locally, anonymized, encrypted and uploaded to the server when the device is connected to a known WLAN access point.

The server consisted of a PostgreSQL database, a participant notification delivery engine, maintenance and data mirroring frameworks. The complete database schema is shown in Figure 9.1. The periodic analysis based on this data was used to send the participation status notifications to the respective individual. This system was also used to recommend data collection and power usage improvement strategies to ensure an uninterrupted data stream.

9.3.2 Ground-Truth Information

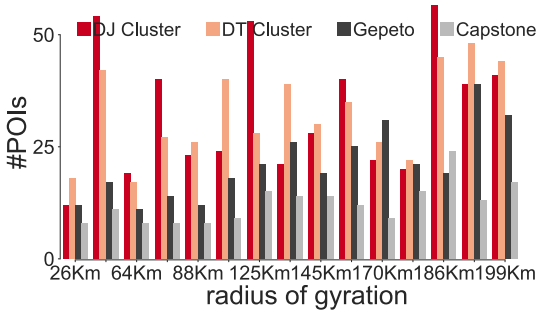


Figure 9.3 – MDC Dataset

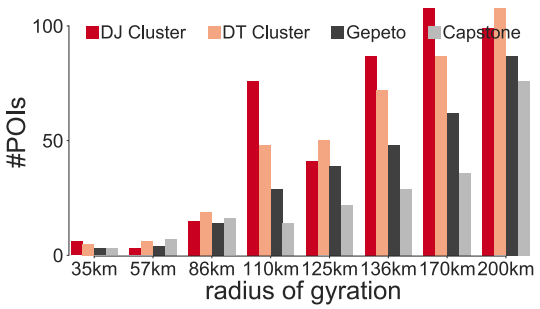


Figure 9.4 – Geolife Dataset

The socio-demographic information collected from the participants include age group, employment/marital status, mode of transportation during the week/week-end, field of study, level of education, frequency of participation in sport activities, allergies, smoking habits among several others. In order to obtain the ground-truth regarding the points of interest, we first computed all the spatiotemporal clusters present in an individual's trajectories throughout the data collection period. In order to select the appropriate technique, we first compared several spatiotemporal clustering approaches. Our objective was to select the approach that computes all the relevant clusters relying on minimal a-priori parameters while minimizing duplicate/redundant zones. We focus on verifying that the approach does not eliminate any true positives while allowing for the presence of false positives under a satisfactory threshold. It is crucial to have approximately equal distribution between true positives and false positives in order to train supervised machine learning approaches.

In order to analyze the sensitivity thresholds of the clustering algorithms in extracting POISs , we benchmark the performance of commonly used spatiotemporal clustering approaches on the MDC dataset [125] (same region as *Breadcrumbs*) and the Geolife dataset [267] to assess the number of clusters extracted. In order to inform the sensitivity analysis, we leverage a large number of existing works that have benchmarked the mobility behavior of individuals in these datasets [237]. We compared the approaches specified by Gambs et al. [75], (1) DJ Cluster [271], (2) DT Cluster [43], (3) TD Cluster [75] and a parameter-less approach (4) Capstone [135]. In order to perform this analysis, we select users with distinct activity areas captured with respect to the radius of gyration of movement to include distinct mobility behaviors. As shown in Figures 9.3 and 9.4 DJ Cluster [271] detects a significantly high number of POIs, not typical for an average user based on the mobility behavioral studies by Thamason et al. [237]. The parameter-less approach [135] and TD Cluster (a variant of DT-clustering) on the other hand detects fewer POIs and could potentially lead to elimination of true positives.

We therefore used a clustering approach based on DT Cluster [43] to compute all the clusters in an individual's trajectories which offers an optimal tradeoff between potentially displaying two or more clusters at a single location and omitting true POIs. We considered that a cluster is defined by a centroid and a radius, the latter is computed using all the points contained in a cluster. Here, we modify the original DT-clustering approach, where we merge two clusters if they overlap with one another, by accounting for the centroids of the clusters as well as their radius. Furthermore, we add a filter on the number of visits to a cluster and select the ones where a participant visited least 3 times during the 12 weeks of the data collection timeframe. These clusters were then displayed to each respective participants along with the validation option to annotate each of the clusters and further provide semantic labels using our application at the end of the collection campaign. The labels span 9 categories (transport, study, residency, work, sustenance, shopping, sports, leisure and other (free-text)). Additionally, each participant tagged their relationship with other individuals participating in the data collection campaign. This information along with the demographic information, geolocation points forming trajectories was aggregated upon removing all the participant identifiers to assemble the dataset. The preprocessing involved removal of the duplicate points and merging all the records sequentially.

9.4 Quantitative Analysis

In this Section we perform quantitative analysis of the *Breadcrumbs* data and present the different feature sets along with the descriptive statistics. *Breadcrumbs* dataset contains 46,380,042 records gathering GPS, WiFi, Bluetooth and accelerometer data points. The dataset was collected for a period of 12 weeks from March-May 2018, retrieving geolocation data of 81 individuals at a sampling rate of 60 sec. The aggregate distance traversed by the participants amount to 548,210 km and the average distance covered per participant is approximately 6,768 km. Figure 9.5 shows the spatial extent traversed by the participants. The largest age groups present in the campaign were 18-21 and 22-27, with 53% and 44% of the sample falling to these age ranges respectively. 57% of the participants identified as females and 73% participants were in a bachelors degree

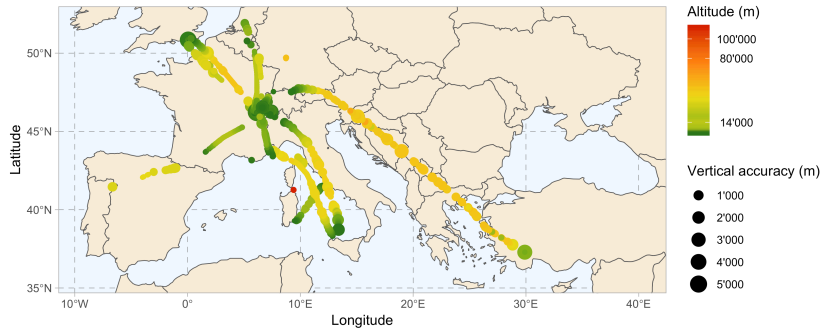


Figure 9.5 – Spatial extent of the geolocation data accompanied with the respective vertical accuracy of data-points.

program and 25% in a masters program. The participants span the faculty of law, medicine, business, economics, literature, physics, biology, chemistry and computer science.

Variable	Min	Q25	Median	Q75	Max	Mean	SD
Longitude	-43.285	6.516	6.588	6.825	100.761	6.582	4.461
Latitude	-22.971	46.317	46.520	46.537	55.663	46.217	2.090
Altitude	-450.0	385.659	415.286	486.020	111779.612	463.582	547.385
Speed	0.0	1.40	9.640	21.460	295.489	13.428	16.860
Horizontal accuracy	0.202	8.0	12.0	32.0	149000.0	70.760	1214.937
Vertical accuracy	1.875	3.0	6.0	10.0	58410.027	14.837	110.618

Table 9.2 – Descriptive statistics of the GPS data points

Type	Total #records	Min per user	Avg #records	Max per user	Size
GPS	14656971	23481	180950	476445	1006.4 MB
WiFi	19363007	17256	239049	441629	443.3 MB
Bluetooth	60986	0	753	5919	6.3 MB
Accelerometer	12299078	18534	151840	421813	844.5 MB

Table 9.3 – Number of data points and ratio per participant

9.4.1 Geolocation Data

The summary of the GPS data points are presented in Table 9.2. Here, the horizontal accuracy indicates a radius about a 2-dimensional point, implying that the true unknown location is within the circle. Vertical accuracy gives the altitude correctness of a 1-dimensional location within the region defined by the radius. The median horizontal and vertical accuracy of the GPS data points is less than 10 meters or less. Table 9.3 shows the number of records collected by the different sensors. The distribution of the POIs in the city and the horizontal accuracy of the GPS coordinates is shown in Figure 9.6.

The WiFi SSIDs are not mapped to the GPS locations, instead a unique identifier corresponding to the MAC address of the WiFi access point is stored. These identifiers act as a spatial indicator of the participant location. The recurrent WiFi connections along with the respective participant is shown in Figure 9.7.

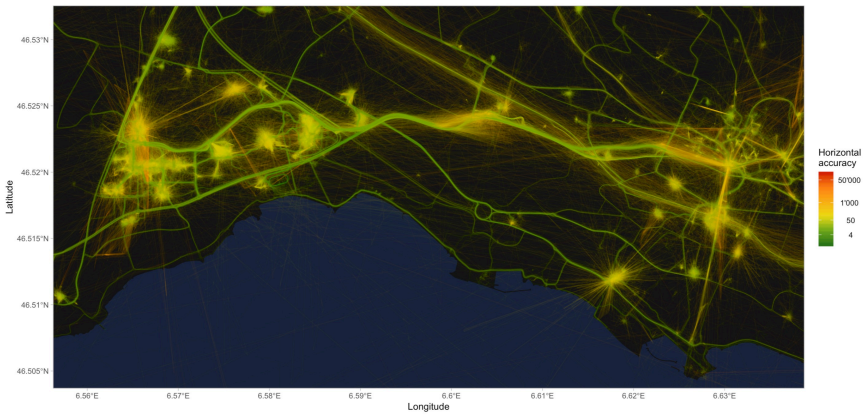


Figure 9.6 – POI clusters and horizontal accuracy of the GPS locations.

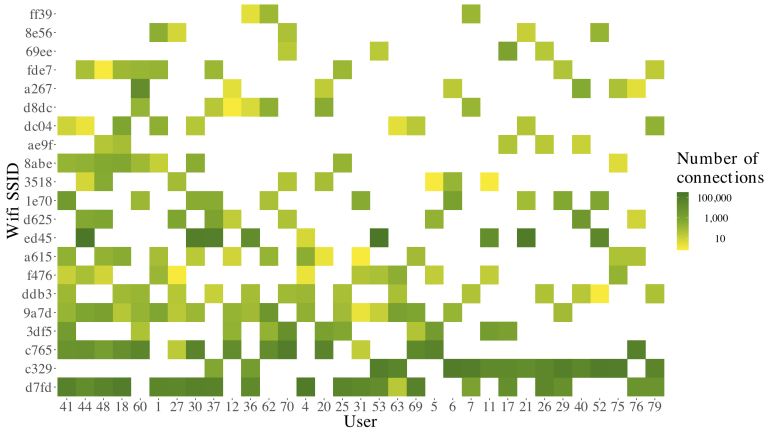


Figure 9.7 – Recurrent WiFi connections and the respective participant.

9.4.2 Points of Interest

In this Section, we summarize the point of interest information extracted form the raw trajectories and present the descriptive statistics regarding the following aspects: (1) distribution of semantic labels, (2) connectivity graph of POI sharing, and (3) geographical overview of the POIs. We find that a majority of the POIs are located at transport hubs, university area and leisure places as shown in Figure 9.8. We also observe the distinct POI clusters shared by a different user groups along with several isolated POIs in Figure 9.9. A nodes in the figure denotes a unique POI and an edge signifies a user connectivity to that POI.

9.4.3 Demographic attributes

In this section, we present the descriptive statistics pertaining to the demographics information collected using survey questionnaires. This information includes transportation mode preference of the participants, health related information and information such as parents' home region and

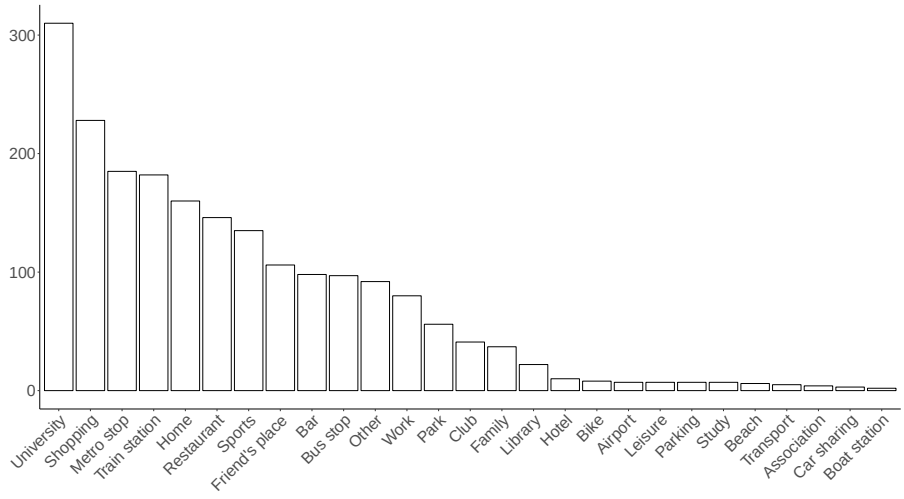


Figure 9.8 – Distribution of POIs according to their semantic labels.

high school location.

From this survey, we have highlighted mobility trends related to transportation modes preferences and patterns as preliminary results. Figure 9.10 shows the transportation mode preferences during weekday and weekend. We observe an increased usage of private transport (cars) during the weekend compared to the weekdays where the participants rely on public transportation, including trams, bus and trains. The usage of bikes and walking preference is similar during the weekdays and the weekend. Figure 9.11 presents the weekly mobility pattern choices according to the parents' home region. We observe that the most represented patterns are *Public Transportation + Bike/Walking* and *Car + Public Transportation + Bike/Walking*. Secondly, the figure also highlights that the least represented is the individuals who only use cars on a weekly basis. The most represented parents' home regions of the individuals who follow the first pattern are located in *France* and *Another Swiss state*. The most represented parents' home region of the individuals who follow the second pattern is located in *Canton de Vaud* (this Swiss district includes the Lausanne region). This finding indicates that most of the individuals of the second pattern study very close to their parent's home region.

In Figures 9.12, 9.13, 9.14 and 9.15, we superimposed health characteristics on weekly transportation mode preferences. Health characteristics are the following: frequency of sport exercise of the Breadcrumbs' participants, if they have seasonal allergies or not, if they smoke or not and the type of diet of the participants. The figures highlight that some specific health characteristics may be related to some particular transportation modes preferences. For example, the majority of the Breadcrumbs' participants who eat a diversified food and most of the time organic seem not use cars during the week in Figure 9.15.

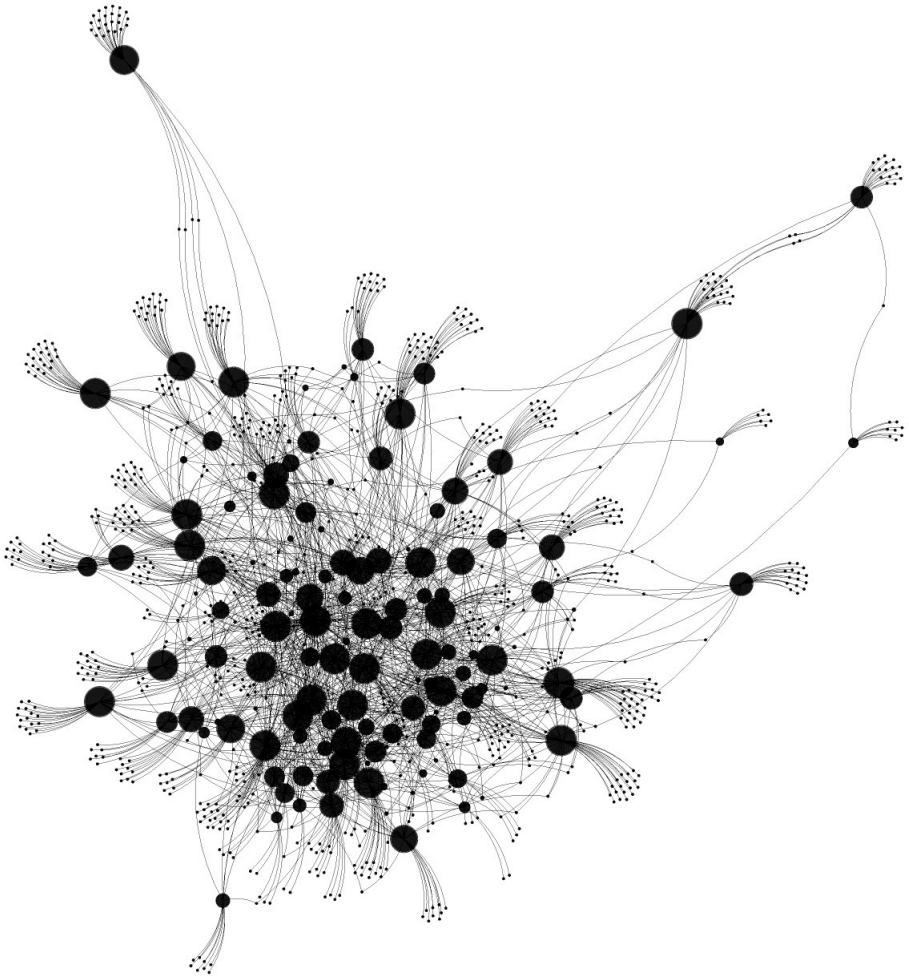


Figure 9.9 – POI to user connectivity graph.

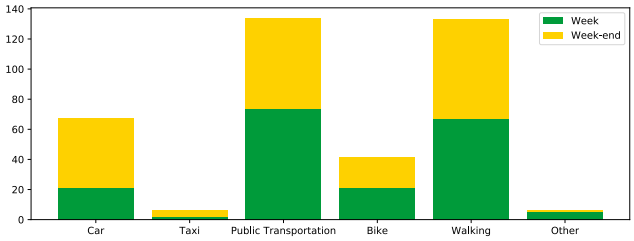


Figure 9.10 – Transportation Modes Preferences for weekday and weekend.

9.5 Clustering Comparison & Validation

In this section, we perform a comparative analysis of four clustering approaches that aim at extracting POIs belonging to all the participants of the *Breadcrumbs* project. We also focus on describing the usage of the ground-truth information to validate and compare the performance of different clustering algorithms.

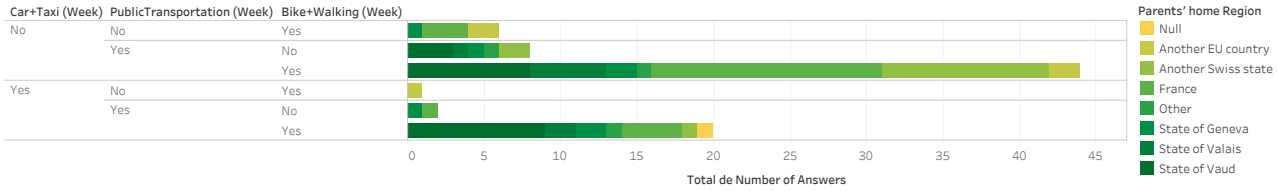


Figure 9.11 – Transportation Modes Weekly Usage And Parents' Home Region.

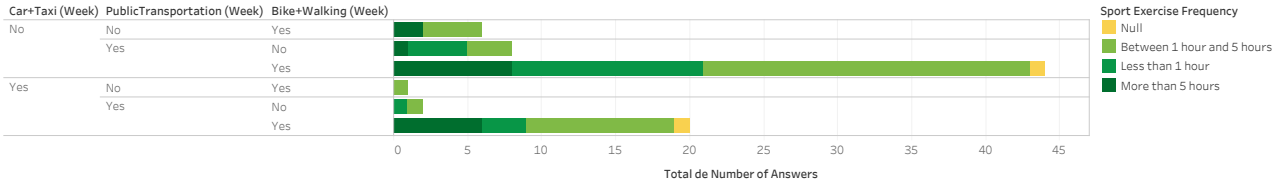


Figure 9.12 – Transportation Modes Weekly Usage And Sport Exercise Frequency.

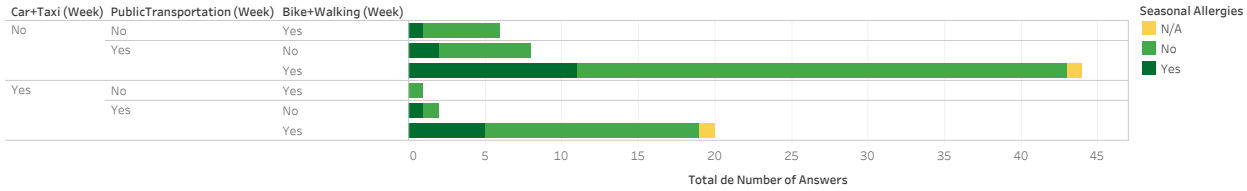


Figure 9.13 – Transportation Modes Weekly Usage And Seasonal Allergies.

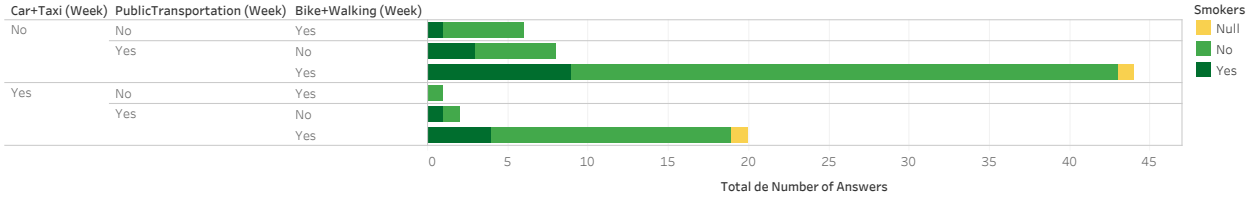


Figure 9.14 – Transportation Modes Weekly Usage And Smokers.

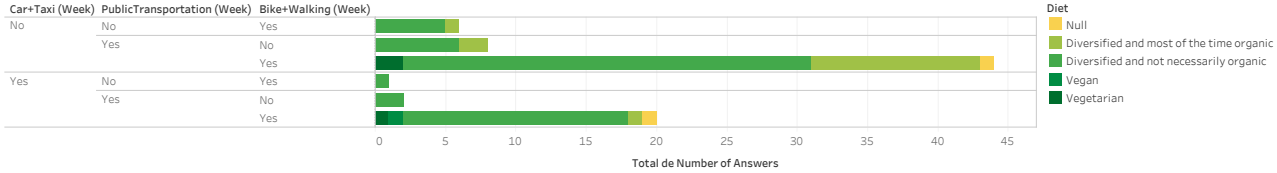


Figure 9.15 – Transportation Modes Weekly Usage And Diet.

9.5.1 Clustering Algorithm Descriptions

K-means and DBSCAN only account for the spatial dimension of the locations, while DJ Cluster and DT Cluster utilize both, the spatial and temporal dimensions of the locations.

k-means. *k*-means is a widely used spatiotemporal clustering algorithm. Amongst the different

versions of *k*-means, we rely on the one proposed by Hartigan and Wong [103]. The algorithm takes two input values: *k*, indicating the number of clusters to be obtained, and the set of data-points to be be grouped in to *k* clusters. The output is *k* clusters representing the user points of interest. The algorithm operates as follows:

- *k* points are randomly chosen in the initial set of points and considered as the initial centroids of the *k* clusters;
- All the points of the initial set of points are then assigned with their closest centroid based on the Euclidean distance between points and the centroids;
- All the centroids of the *k* clusters are then updated by calculating the mean of all points being associated to a cluster;
- Finally the algorithm iterates until converging into a stable state in which there is no more additional way to minimize the total sum of squared Euclidean distances between points and their related centroid.

DBSCAN. DBSCAN is a clustering algorithm based on the density of the points linked to clusters. Unlike *k*-means, we do not need to specify the number clusters *a priori* and is not known beforehand. The algorithm takes three input values: *eps*, which is the maximum radius of the neighborhood of a point, *minPts*, which indicates the minimum number of points that must be in the neighborhood of a point, and the initial set of points (i.e., locations) that must be grouped in several clusters. These clusters represent the user POIs. DBSCAN operates as follows:

- The algorithm starts evaluating each data-point of the initial by computing the density with respect to all the other points using the *eps* value and the *minPts*. This step associates a category to each point between the core, border and noise points. Noise points are later deleted;
- Neighborhood core points, which are density-reachable, are then associated with the same cluster;
- Finally, border points are associated to the nearest cluster.

DJ Cluster. DJ Cluster is a clustering algorithm that takes into account spatiotemporal dimensions [271]. The algorithm takes four input parameters: *minPts*, which is minimum of points located in a candidate cluster, *r*, which is the radius of a candidate cluster, *speed-threshold*, which is maximum speed we will use to extract the meaningful points from which the candidate clusters are built, and the initial set of points (i.e., locations).

- The algorithm initiates by deleting all moving points (a moving point is deleted if the time difference with the previous point is greater than *speed-threshold*);

- Then, we compute the neighborhood density of each remaining point and only keep the points that have at least *minPts* points in a specific *r*. The remaining points of this step are the candidate clusters;
- Finally, a merging step is realized. The algorithm merges the candidates clusters that share at least one common point.

DT Cluster. DT Cluster is a spatiotemporal clustering algorithm [43]. The algorithm takes three input parameters: *d*, which is the radius of the candidate clusters created at the beginning of the process, *t*, which is minimum time period spent in the candidate clusters also created at the beginning of the algorithm, and the initial set of points (i.e., locations).

- The algorithm starts by evaluating the full set of points (i.e., locations) like a time series. It extracts all candidate clusters by gathering the successive points that are located in the same *d* for a duration greater than *t*;
- To finish the process, a merging step is performed. This step merges all the candidate clusters that are lie at a distance *d/3* from each other.

9.5.2 Evaluation Framework and Parameters

In this section, we present the performance evaluation framework. The evaluation is based on two steps that will be described hereafter. For sake of simplicity, we will take the example of one single *Breadcrumbs* participant. However, the evaluation in the results section considers all the 81 participants. The evaluation takes as input the ground-truth information labelled by the user and a clustering algorithm Figure 9.16 presents the ground-truth of the participant as well as the annotations (i.e., yes or no) linked to every points. The valid POIs are therefore the points labelled as *yes* by the participant. The points labeled *no* are as well crucial to the evaluation framework.

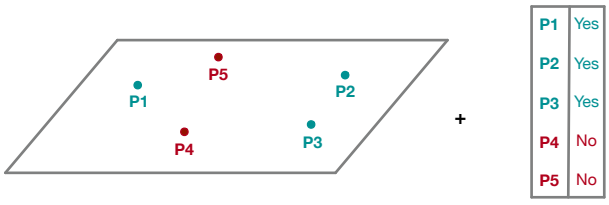


Figure 9.16 – Ground-truth and annotation (validation by the participant with a yes or a no for each point detected)

The first step consists in linking each point of the ground-truth with a corresponding cluster, which was extracted by the clustering algorithm. To do so, we will find the closest cluster to each point of the ground-truth by computing the euclidian distance between them. The output of step one is illustrated in Figure 9.17.

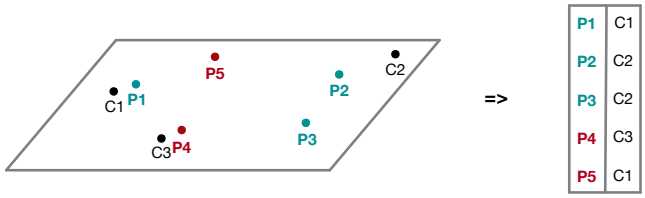


Figure 9.17 – Links between Ground-truth (P) and Clusters (C) according to a minimum distance

Clustering Algorithm	Parameter Values
***k*-means**	k = 10/30/100/200/300/1000
DBSCAN	minPts = 30 / eps = 0.003/0.001/0.0007/0.0002/0.0001
DJ Cluster	radius = 60.0 m / speed-threshold = 1.5 km/h / minPts = 10/20/50/100/200/500
DT Cluster	t = 15 mins (900 sec) / d = 40/60/100/150/300 m

Table 9.4 – Parameters of the Clustering Algorithms

The second step aims at extracting the number of true positives, false positives, true negatives and false negatives that will enable us to compute the true positive rate (sensitivity) and false positive rate (1 - specificity) in order to formulate the ROC (Receiver Operating Characteristic) curve. In order to perform this step, we will use a parameter *d* that will help to determine if a cluster is located in a validation zone around a point of the ground-truth. The validation results are described in the column *Distance Validation* in Figure 9.18. The true positives (TP) have a ground-truth validation and a distance validation that correspond to *Yes*, while the true negatives (TN) have a ground-truth validation and a distance validation that are equal to *No*. The false positives have a ground-truth validation value that is equal to *No* and a distance validation value to *Yes*. Similarly the false negatives posses contrary labels to the above.

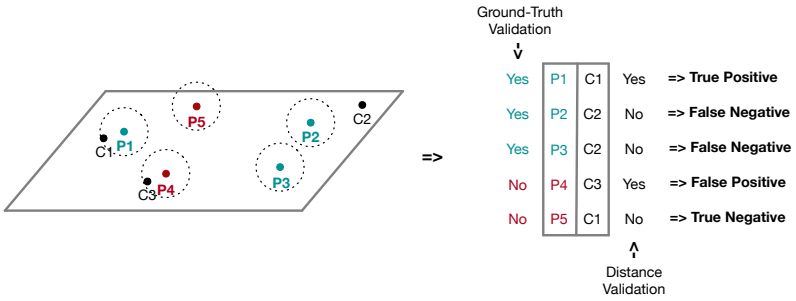


Figure 9.18 – Final Annotation (TP, TN, FP and FN)

The ROC curve is computed with the true positive rate, indicated on the y-axis, and true negative rate, indicated on the x-axis.

- True positive rate (sensitivity): True Positives / (True Positives + False Negatives)
- False positive rate (1 - specificity): False Positives / (False Positives + True Negatives)

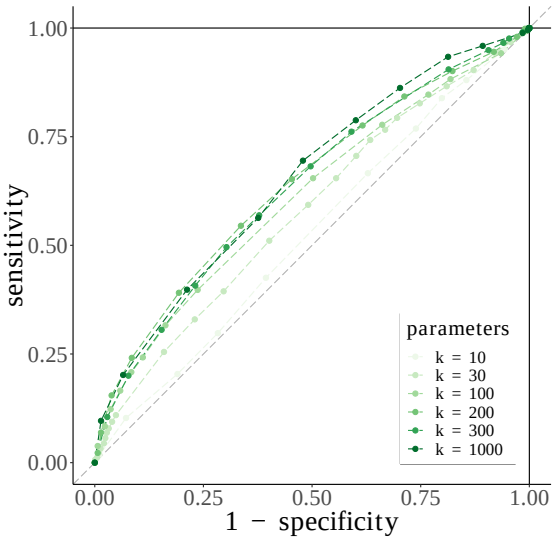


Figure 9.19 – ROC Curve - *k*-means Clustering Algorithm

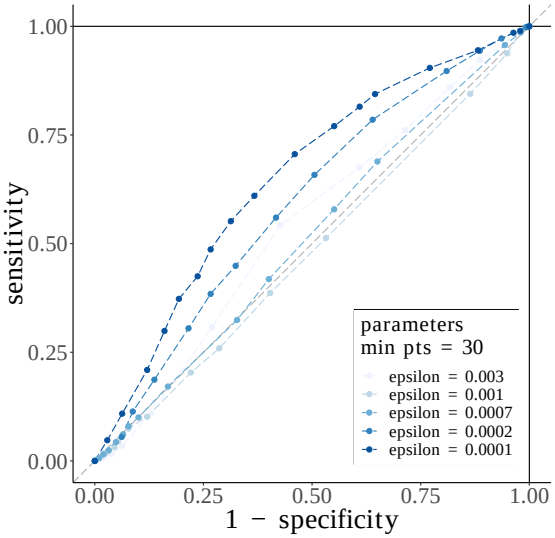


Figure 9.20 – ROC Curve - DBSCAN Clustering Algorithm

Table 9.4 describes all the selected parameters for each of clustering algorithms. We selected the parameters according to previous research works in the same area [136, 174] and the more plausible and values according to the spatial or spatiotemporal clustering algorithm context.

9.5.3 Results

The results of our comparative analysis are shown in Figures 9.19, 9.20, 9.21 and 9.22 for *k*-means, DBSCAN, DJ Cluster and DT Cluster respectively. It is crucial to notice that the diagonal (the light gray dotted line from the bottom left corner to the top right corner) on each graph represents

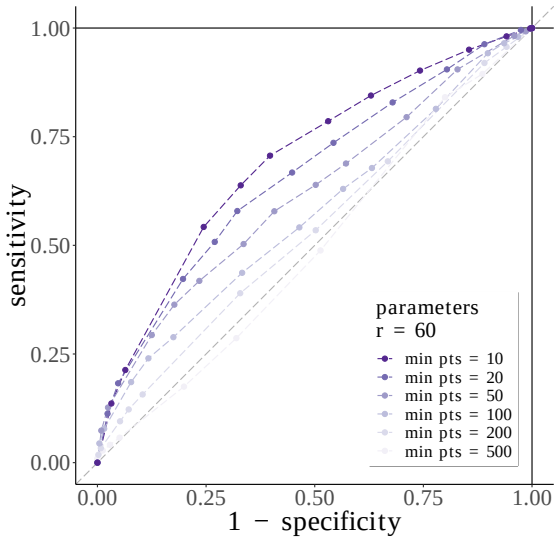


Figure 9.21 – ROC Curve - DJ Cluster Algorithm

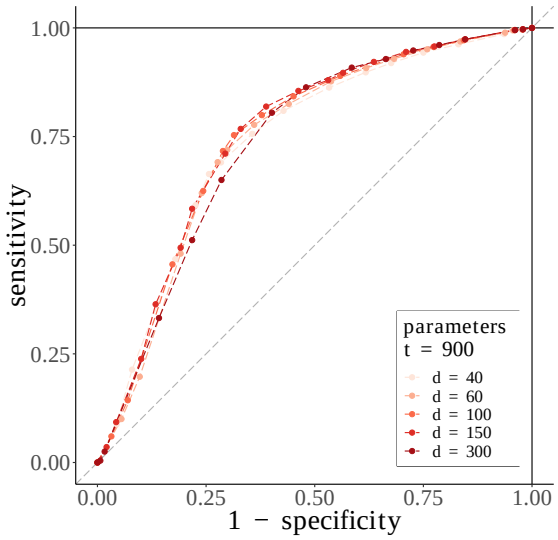


Figure 9.22 – ROC Curve - DT Cluster Algorithm

the worst case situation in which the algorithm has no discrimination capability to identify a cluster as a POI. Regarding *k*-means in Figure 9.19, increasing the *k* marginally increases the performance until reaching a limit whereafter the performance gain halts. As observed in Figure 9.20, DBSCAN depicts a better performance as compared to *k*-means. DBSCAN introduces the notion of density of the neighborhood of each point evaluated, which increases the accuracy compared to the *k*-means The lower the epsilon (that indicates the area evaluated around a point), the higher is the algorithmic performance. We also observe that DJ cluster performs better as compared to both DBSCAN and *k*-means (see Figure 9.21). Furthermore, the parameter *minPts* when set to the least number provides the best results This can be justified based on the lower values of the radius *r* (i.e., 60 meters) in addition to deletion of all the moving points. DT

Cluster provides the best overall performance as observed in Figure 9.22 We also notice that the highest value of parameter d (i.e., 300 meters) is not necessarily correlated with the best performance Based on these results we can conclude that the clustering approaches account for the spatiotemporal parameters are comparatively better adapt to extract POIs.

9.6 Conclusion

In this paper, we introduce a feature rich geolocation mobility dataset *Breadcrumbs*. In addition to fine-grained demographic attributes, contact and calendar records, social relationships we also provide ground truth and semantic labels for the points of interest. We describe the complete data collection process and our methodology to collect ground-truth information in GIS domain. Our qualitative analysis has shed light on several aspects of this dataset including the POI connectivity, WiFI connection recurrence, POI distribution. We specify the use cases, applicable research domains and validation methodologies using the unique features of *Breadcrumbs*. In addition, we have also highlighted the utility of health related information and transportation modes preferences. To showcase a use case of our dataset, we have performed a comparative study of four clustering approaches to extract POI cluster from GPS trajectories. We have proposed a validation methodology while using the ground-truth labels illustrated the obtained results. We learn that DT Clustering outperforms DJ cluster, DBSCAN and k-means and we discuss its implication. We make *Breadcrumbs* accessible to the research community in order to facilitate and advance the GIS research.

Conclusion

In this thesis, we studied human mobility and explored research questions while proposing solutions and amendments to current shortcomings associated with three different perspectives of human mobility.

1. **Modeling human mobility.** Can mobility trajectories serving as a proxy for movement behaviors of individuals be quantified in order to facilitate data-driven modeling of human mobility?
2. **Predicting human mobility.** Can the implicit properties of mobility trajectories be leveraged to formulate privacy-preserving frameworks for predicting human mobility?
3. **Capturing human mobility.** How feasible is the applicability of recent deep-learning architectures to capture and replicate human mobility dynamics?

We divided the thesis in three parts following these three perspectives; each containing the associated objectives, research questions and our contributions. This section of the thesis recalls our key contributions related to each perspective, highlights the limitations of our research and providers some insights into the opportunities for future-work.

Summary of our Contributions

In the first part of the thesis, we explored human mobility from the modeling perspective, wherein we highlighted the shortcomings of adopting a data-agnostic approach for mobility modeling. In the three chapters belonging to this part, we experimentally demonstrated the consequences of relying on rigid *a priori* assumptions governing mobility dynamics such as misleading performance evaluations, biased conclusions and unfair cross-model comparisons. To this end, in Chapter 1 we performed a large-scale semi-automated literature review on human mobility modeling and presented our insights that serve as a roadmap for research and practice in this area. We highlighted research trends and shortcomings in current research practices predominantly concerning failure in identifying source of empirical gains and adoption of inaccurate validation methodologies. We proposed a data-driven mobility modeling framework

that relies on mobility meta-attributes and a sound performance validation strategy. In Chapter 2 we revised the widely accepted assumptions governing human mobility and demonstrated the ill-founded nature of the mobility-entropy estimation scheme. We empirically proved the invalidity of the currently established upper limit of the mobility prediction by demonstrating that human mobility is governed by non-Markovian character. In Chapter 3, we explored the consequences of applying data-agnostic bounds to extract points of interests from location trajectories. We proposed a technique to identify points of interest that is independent of spatiotemporal or behavioral bounds and hence applicable to a wide variety of datasets containing users having disparate mobility behaviors.

In the second part of the thesis, we explored on means to realize an end-to-end privacy-preserving framework for operationalizing mobility prediction driven location-based services. In Chapter 4 we focussed on designing a mobility prediction engine leveraging realtime location data-streams that minimizes sharing private location traces with third party service providers. We demonstrated that, this approach constantly adapts to dynamic user mobility behaviors, utilizes considerably less data as compared to the conventional approaches while achieving comparable levels of prediction accuracy. In Chapter 5 we demonstrated the practicality of this approach by implementing the complete prediction pipeline consisting of point of interest detection and next-place prediction on a commodity smartphone chipset. This technique offloads computationally expensive server side tasks on to the digital signal processor chips that executes signal processing pipelines by treating 3-dimensional location trajectories as 2-dimensional signals. In Chapter 6 we focussed on achieving a satisfactory level of trade-off between service utility and user privacy by encapsulating a location-based service onto Intel Software Guard eXtensions (SGX), leveraging its privacy guarantees. We demonstrated that hardware based trusted execution environments offer a promising alternative for delivering privacy-aware location-based services while preserving the service utility.

In the third part of this thesis, we explored on means to capture human mobility trajectories to address the issue to limited mobility dataset availability. We adopted two different approaches to this end: (1) replicating mobility trajectories by applying deep-learning to existing datasets, and (2) collecting real-world location trajectories from moving entities in Chapter. In Chapter 7 and Chapter 8, we calibrated and assessed generative and discriminative models in their ability learn, memorize and reproduce synthetic mobility trajectories given a real-world dataset. We performed an extensive evaluation of the generated trajectories by assessing their geographic and semantic similarity and discussed the trade-offs and insights. Chapter 9 addressed the second aspect wherein we launched a data collection campaign to aggregate dataset containing geospatial data from multiple mobile phone sensors (GPS, WiFi, Bluetooth) from 81 participants for a duration of 12 weeks. We also collected ground truth information regarding the user points of interest, their semantic labels along with the social relationships amongst the participants. We illustrated the data collection methodology and presented several use cases that leverage the unique characteristics of this dataset.

Limitations and Future Work

In Chapter 1, we performed a large-scale literature review in the domain of human mobility modeling consisting of 1680 articles published in the span of last two decades. Mobility modeling is a large domain, of which we explored one aspect explicitly defining geospatial movements as *spatiotemporal positions of moving objects (instant, point) devoid of annotations such as semantic, visual or transportation modes*. We focus on modeling mobility based on raw data aggregated from individuals in the form of latitude, longitude and the timestamps, without considering any other supplementary information. As data collection and processing techniques continue to evolve, collection of semantic information such as visited places, observations, news, events and personal sentiments is getting simplified. This information opens up new possibilities by serving as additional attributes to improve the mobility modeling process [184]. Applying visual analytics to mobility modeling can create added value by blending the computing capability of mobile platforms with perceived visual information [128]. In addition, we explore Markov processes, deep learning and statistical learning and assess their capacity to model next-stop movements. However, several data mining approaches, such as template matching and decision tress, which have significantly lower computational complexity, could be added to our data-driven model selection framework [86]. This will not only provide users with a wider array of options to select the mobility modeling technique from considering the availability of computing resources, but it will also improve the calibration of meta-attribute thresholds.

In Chapter 2, we demonstrate the inadequacy of the current approach to estimate mobility entropy that is based on Fano's inequality [274] and Lempel-Ziv encoding [227]. We experimentally show that this estimation could be improved by accounting for the long distance dependencies present in mobility trajectories, however we did not formalize this concept. In order to enhance the understanding of the correlation between these dependencies and predictability, it is critical to mathematically formulate the correlation between long-distance dependencies in location trajectories and asymptotic converge of the entropy. Recent developments and understanding of deep learning architectures can be used to interpret the relation between limits of gradient based learning algorithms and the duration of the dependencies to be captured [21]. These results could be used to formalize the trade-offs between memorization capabilities of recurrent neural network models and latching on information for a certain period.

In Chapter 3, 4 and 5 we present approaches to extract points of interests from mobility trajectories independent of any parameters and forecast the next place. We implement these approaches on a mobile phone chipset demonstrating the feasibility of our approach to operate on commodity smartphones. However, our approach only considers single user's mobility trajectories, while training the prediction models and utilizing them in location-based services. However, the accuracy of such models could be greatly improved by the collective knowledge gained from the aggregate knowledge derived from mobility trajectories of multiple individuals. Approaches such as federated learning that enable distributed machine learning by formulating model training on a large corpus of decentralized data could be applied to achieve high accuracy, while preserving user privacy [1]. Several federated learning libraries are now available operational on mobile devices,

which could be leveraged to extend our privacy preserving mobility prediction pipeline [101]. It had been shown that for tasks such as keyboard stroke prediction, training on an aggregated dataset on client devices achieves better prediction recall [101]. Federated learning systems provide users a higher level of control over their data thus incorporating privacy by design with distributed training and aggregation over a large subset of users.

In Chapter 6, we leverage the Intel Software Guard eXtensions (SGX) trusted execution environment to provide high service utility levels without compromising on user privacy. However, it has recently been shown that SGX is susceptible to several attack vectors including microarchitectural and side-channel attacks, compromising the privacy gurantees [242]. As a result, providing a satisfactory trade-off between utility and privacy in location-based services is still an open research challenge. To this end, the inherent properties of mobility trajectories could be leveraged to facilitate cryptographic primitives such as partial homomorphic encryption [26].

Bibliography

[1] Google federated learning. https://research.googleblog.com/2017/04/federated-learning-collaborative.html.

[2] Mobile apps privacy compliance. https://venturebeat.com/2018/01/19/safedk-55-of-mobile-apps-dont-comply-with-european-privacy-regulations/.

[3] Uber privacy. https://www.usatoday.com/story/tech/2014/11/19/uber-privacy-tracking/19285481/.

[4] Uber starwood. https://www.forbes.com/sites/ronhirson/2015/03/23/uber-the-big-data-company/#4af73b4118c7.

[5] A. Al-Molegi, M. Jabreel, and B. Ghaleb. Stf-rnn: Space time features-based recurrent neural network for predicting people next location. *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–7, 2016.

[6] N. A. Amirrudin, S. H. Ariffin, N. A. Malik, and N. E. Ghazali. User's mobility history-based mobility prediction in lte femtocells network. In *2013 IEEE International RF and Microwave Conference (RFM)*, pages 105–110. IEEE, 2013.

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

[8] S. Arimoto. Information-theoretical considerations on estimation problems. *Information and control*, 19(3):181–194, 1971.

[9] A. Asahara, K. Maruyama, A. Sato, and K. Seto. Pedestrian-movement prediction based on mixed markov-chain model. In *GIS*, 2011.

[10] D. Ashbrook and T. Starner. Learning significant locations and predicting user movement with gps. In *SEMWEB*, 2001.

[11] D. Ashbrook and T. Starner. Learning significant locations and predicting user movement with gps. In *Wearable Computers, 2002.(ISWC 2002). Proceedings. Sixth International Symposium on*, pages 101–108. IEEE, 2002.

[12] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7:275–286, 2003.

[13] M. Backes, M. Humbert, J. Pang, and Y. Zhang. walk2friends: Inferring social links from mobility profiles. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1943–1957. ACM, 2017.

[14] F. M. Bandi, B. Perron, A. Tamoni, and C. Tebaldi. The scale of predictability. *Journal of Econometrics*, 208(1):120–140, 2019.

[15] H. Bapierre, G. Groh, and S. Theiner. A variable order markov model approach for mobility prediction. *Pervasive Computing*, pages 8–16, 2011.

[16] A.-L. Barabasi. The origin of bursts and heavy tails in human dynamics. *Nature*, 435(7039):207, 2005.

[17] M. Baratchi, N. Meratnia, P. J. Havinga, A. K. Skidmore, and B. A. Toxopeus. A hierarchical hidden semi-markov model for modeling mobility data. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 401–412. ACM, 2014.

[18] M. Baratchi, N. Meratnia, P. J. M. Havinga, A. K. Skidmore, and B. A. G. Toxopeus. A hierarchical hidden semi-markov model for modeling mobility data. In *UbiComp*, 2014.

[19] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz. Sumo–simulation of urban mobility. In *The Third International Conference on Advances in System Simulation (SIMUL 2011), Barcelona, Spain*, volume 42, 2011.

[20] S. Bell, A. McDiarmid, and J. Irvine. Nodobo: Mobile phone as a software sensor for social network research. In *2011 IEEE 73rd vehicular technology conference (VTC Spring)*, pages 1–5. IEEE, 2011.

[21] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[22] Y. Bengio, P. Y. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5 2:157–66, 1994.

[23] C. Bergmeir and J. M. Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012.

[24] S. K. Bharti and K. S. Babu. Automatic keyword extraction for text summarization: A survey. *arXiv preprint arXiv:1704.03242*, 2017.

[25] W. Bialek and N. Tishby. Predictive information. *arXiv preprint cond-mat/9902341*, 1999.

[26] V. Biksham and D. Vasumathi. Homomorphic encryption techniques for securing data in cloud computing: A survey. *International Journal of Computer Applications*, 975:8887, 2017.

[27] V. Bindschaedler and R. Shokri. Synthesizing plausible privacy-preserving location traces. In *Security and Privacy (SP), 2016 IEEE Symposium on*, pages 546–563. IEEE, 2016.

[28] J. D. M. M. Biomo, T. Kunz, and M. St-Hilaire. An enhanced gauss-markov mobility model for simulations of unmanned aerial ad hoc networks. In *2014 7th IFIP Wireless and Mobile Networking Conference (WMNC)*, pages 1–8, May 2014.

[29] A. Bogomolov, B. Lepri, J. Staiano, N. Oliver, F. Pianesi, and A. Pentland. Once upon a crime: towards crime prediction from demographics and mobile data. In *Proceedings of the 16th international conference on multimodal interaction*, pages 427–434. ACM, 2014.

[30] T. Brinkhoff. Generating network-based moving objects. In *Scientific and Statistical Database Management, 2000. Proceedings. 12th International Conference on*, pages 253–255. IEEE, 2000.

[31] T. Brinkhoff. A framework for generating network-based moving objects. *GeoInformatica*, 6(2):153–180, 2002.

[32] F. Calabrese, G. D. Lorenzo, L. Liu, and C. Ratti. Estimating origin-destination flows using mobile phone location data. *IEEE Pervasive Computing*, 10:36–44, 2011.

[33] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. A. Peterson, H. Lu, X. Zheng, M. Musolesi, G.-S. Ahn, et al. The rise of people-centric sensing. *IEEE Internet Computing*, (4):12–21, 2008.

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

[35] J. Capka and R. Boutaba. Mobility prediction in wireless networks using neural networks. In *Management of Multimedia Networks and Services*, pages 320–333. Springer Berlin Heidelberg, 2004.

[36] V. P. Chakka, A. Everspaugh, and J. M. Patel. Indexing large trajectory data sets with seti. In *CIDR*, 2003.

[37] S. Chang, Y. Zhang, W. Han, M. Yu, X. Guo, W. Tan, X. Cui, M. J. Witbrock, M. A. Hasegawa-Johnson, and T. S. Huang. Dilated recurrent neural networks. In *NIPS*, 2017.

[38] B. Chapuis and B. Garbinato. Geodabs: Trajectory indexing meets fingerprinting at scale. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 1086–1095. IEEE, 2018.

[39] B. Chapuis, A. Moro, V. Kulkarni, and B. Garbinato. Capturing complex behaviour for predicting distant future trajectories. In *MobiGIS*, 2016.

[40] B. Chapuis, A. Moro, V. Kulkarni, and B. 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.

[41] X. Chen, J. Pang, and R. Xue. Constructing and comparing user mobility profiles for location-based services. SAC, pages 261–266, 2013.

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

[43] Y. Chen and L. Tu. Density-based clustering for real-time stream data. In *KDD*, 2007.

[44] S. Chessa, M. Girolami, L. Foschini, R. Ianniello, A. Corradi, and P. Bellavista. Mobile crowd sensing management with the participact living lab. *Pervasive and Mobile Computing*, 38:200–214, 2017.

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

[46] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *KDD*, 2011.

[47] N. Chomsky. On certain formal properties of grammars. *Information and control*, 2(2):137–167, 1959.

[48] Y. Chon, H. Shin, E. Talipov, and H. Cha. Evaluating mobility models for temporal prediction with high-granularity mobility data. In *2012 IEEE International Conference on Pervasive Computing and Communications*, pages 206–212. IEEE, 2012.

[49] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan. Private information retrieval. In *FOCS*, 1995.

[50] J. Chung, S. Ahn, and Y. Bengio. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*, 2016.

[51] R. I. Ciobanu and C. Dobre. CRAWDAD dataset upb/hyccups (v. 2016-10-17). Downloaded from https://crawdad.org/upb/hyccups/20161017, Oct. 2016.

[52] A. Clauset, C. R. Shalizi, and M. E. Newman. Power-law distributions in empirical data. *SIAM review*, 51(4):661–703, 2009.

[53] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR*, 2002.

[54] A. Cuttone, S. Lehmann, and M. C. González. Understanding predictability and exploration in human mobility. *EPJ Data Science*, 7(1):2, 2018.

[55] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel. Unique in the Crowd: The privacy bounds of human mobility. *Scientific Reports*, 3, Mar. 2013.

[56] T. 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.

[57] T. M. T. Do and D. Gatica-Perez. Contextual conditional models for smartphone-based human mobility prediction. UbiComp '12, pages 163–172, 2012.

[58] C. Düntgen, T. Behr, and R. H. Güting. Berlinmod: a benchmark for moving object databases. *The VLDB Journal—The International Journal on Very Large Data Bases*, 18(6):1335–1368, 2009.

[59] N. Eagle and A. Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10:255–268, 2005.

[60] E. Eftelioglu, X. Tang, and S. Shekhar. Geographically robust hotspot detection: A summary of results. In *ICDMW*, pages 1447–1456, 2015.

[61] J. L. Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990.

[62] E. ElSalamouny and S. Gambs. Differential privacy models for location-based services. *Transactions on Data Privacy*, 9(1):15–48, 2016.

[63] C. Esteban, S. L. Hyland, and G. Rätsch. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*, 2017.

[64] M. Ester, H.-p. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231, 1996.

[65] V. Etter, M. Kafsi, and E. 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.

[66] V. Etter, M. Kafsi, and E. Kazemi. Been there, done that: What your mobility traces reveal about your behavior. Technical report, 2012.

[67] H. Fanaee-T and J. Gama. An eigenvector-based hotspot detection. *arXiv preprint arXiv:1406.3191*, 2014.

[68] K. Farrahi and D. Gatica-Perez. Discovering routines from large-scale human locations using probabilistic topic models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):3, 2011.

[69] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng, A. Guo, and D. Jin. Deepmove: Predicting human mobility with attentional recurrent networks. In *WWW*, 2018.

[70] M. P. Fillekes, C. Röcke, M. Katana, and R. Weibel. Self-reported versus gps-derived indicators of daily mobility in a sample of healthy older adults. *Social Science and Medicine*, 220:193 – 202, 2019.

[71] B. Furletti, R. Trasarti, P. Cintia, and L. Gabrielli. Discovering and understanding city events with big data: the case of rome. *Information*, 8(3):74, 2017.

[72] Y. Gahi, M. Guennoun, Z. Guennoun, and K. El-Khatib. Privacy preserving scheme for location-based services. *J. Information Security*, 3:105–112, 2012.

[73] K. Gai, M. Qiu, and H. Zhao. Privacy-preserving data encryption strategy for big data in mobile cloud computing. *IEEE Transactions on Big Data*, pages 1–1, 2017.

[74] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Show me how you move and i will tell you who you are. In *SPRINGL*, 2010.

[75] S. Gambs, M.-O. Killijian, and M. N. 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*, pages 34–41. ACM, 2010.

[76] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez. Next place prediction using mobility markov chains. In *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, page 3. ACM, 2012.

[77] S. Gambs, M.-O. Killijian, and M. Núñez del Prado Cortez. Show me how you move and i will tell you who you are. *Trans. Data Privacy*, pages 103–126, 2011.

[78] B. Gedik and L. Liu. Protecting location privacy with personalized k-anonymity: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 7:1–18, 2008.

[79] G. Geenens, A. Charpentier, and D. Paindaveine. Probit transformation for nonparametric kernel estimation of the copula density. *Bernoulli*, 23(3):1848–1873, 2017.

[80] S. Gerchinovitz, P. Ménard, and G. Stoltz. Fano's inequality for random variables. *arXiv preprint arXiv:1702.05985*, 2017.

[81] M. Gerla. Ipv6 flow handoff in ad hoc wireless networks using mobility prediction. 1999.

[82] S. C. Geyik, E. Bulut, and B. K. Szymanski. Pcfg based synthetic mobility trace generation. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[83] S. Ghosh and S. K. Ghosh. Modeling of human movement behavioral knowledge from gps traces for categorizing mobile users. In *WWW*, 2017.

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

[85] F. Giannotti, A. Mazzoni, S. Puntoni, and C. Renso. Synthetic generation of cellular network positioning data. In *Proceedings of the 13th annual ACM international workshop on Geographic information systems*, pages 12–20. ACM, 2005.

[86] F. Giannotti and D. Pedreschi. Mobility, data mining and privacy - geographic knowledge discovery. In *Mobility, Data Mining and Privacy*, 2008.

[87] G. Gidófalvi and F. Dong. When and where next: Individual mobility prediction. MobiGIS '12, pages 57–64, 2012.

[88] G. Gidofalvi and T. B. Pedersen. St–acts: a spatio-temporal activity simulator. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 155–162. ACM, 2006.

[89] Google. S2 Geometry. https://s2geometry.io/, 2017. [Online; accessed 25-July-2018].

[90] P. Grassberger. Estimating the information content of symbol sequences and efficient codes. *IEEE Transactions on Information Theory*, 35(3):669–675, 1989.

[91] P. Grassberger. Entropy estimates from insufficient samplings. *arXiv preprint physics/0307138*, 2003.

[92] A. Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

[93] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[94] S. Grossberg. Recurrent neural networks. *Scholarpedia*, 8(2):1888, 2013.

[95] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *MobiSys*, 2003.

[96] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services*, pages 31–42. ACM, 2003.

[97] L. Gu. Moving kriging interpolation and element-free galerkin method. *International journal for numerical methods in engineering*, 56(1):1–11, 2003.

[98] D. Gupta, B. Mood, J. Feigenbaum, K. Butler, and P. Traynor. Using intel software guard extensions for efficient two-party secure function evaluation. In *International Conference on Financial Cryptography and Data Security*, pages 302–318. Springer, 2016.

[99] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*, volume 1, page 6, 2010.

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

[101] A. Hard, K. Rao, R. Mathews, F. Beaufays, S. Augenstein, H. Eichner, C. Kiddon, and D. Ramage. Federated learning for mobile keyboard prediction. *ArXiv*, abs/1811.03604, 2018.

[102] R. Hariharan and K. Toyama. Project lachesis: parsing and modeling location histories. In *International Conference on Geographic Information Science*, pages 106–124. Springer, 2004.

[103] J. Hartigan and M. Wong. Algorithm AS 136: A K-means clustering algorithm. *Applied Statistics*, pages 100–108, 1979.

[104] M. D. Hauser, N. Chomsky, and W. T. Fitch. The faculty of language: What is it, who has it, and how did it evolve? *science*, 298(5598):1569–1579, 2002.

[105] A. R. Hevner. A three cycle view of design science research. *Scandinavian journal of information systems*, 19(2):4, 2007.

[106] J. Hightower, S. Consolvo, A. LaMarca, I. Smith, and J. Hughes. Learning and recognizing the places we go. In *International Conference on Ubiquitous Computing*, pages 159–176. Springer, 2005.

[107] W. Hilberg. Der bekannte grenzwert der redundanzfreien information in texten-eine fehlinterpretation der shannonschen experimente? *Frequenz*, 44(9-10):243–248, 1990.

[108] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9 8:1735–80, 1997.

[109] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. A group mobility model for ad hoc wireless networks. In *MSWiM '99*, 1999.

[110] Idiap. NMDC Dataset. https://www.idiap.ch/dataset/mdc/download, 2012. [Online; accessed 26-July-2018].

[111] E. L. Ikanovic and A. Mollgaard. An alternative approach to the limits of predictability in human mobility. *EPJ Data Science*, 6(1):12, 2017.

[112] Inria. PrivaMOv Dataset. https://projet.liris.cnrs.fr/privamov/project/, 2012. [Online; accessed 26-July-2018].

[113] J. Jeong, M. Leconte, and A. Proutiere. Mobility prediction using non-parametric bayesian model. *arXiv preprint arXiv:1507.03292*, 2015.

[114] B. Jiang, J. Yin, and S. Zhao. Characterizing the human mobility pattern in a large street network. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 80 2 Pt 1:021136, 2009.

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

[116] H. Joe. *Dependence Modeling with Copulas*. Chapman & Hall/CRC, 2014.

[117] I. Jung, M. Kulldorff, and O. J. Richard. A spatial scan statistic for multinomial data. *Statistics in medicine*, 29 18:1910–8, 2010.

[118] J. H. Kang, W. Welbourne, B. Stewart, and G. Borriello. Extracting places from traces of locations. *Mobile Computing and Communications Review*, 9:58–68, 2004.

[119] V. Karande, E. Bauman, Z. Lin, and L. Khan. Sgx-log: Securing system logs with sgx. In *AsiaCCS*, 2017.

[120] A. Karatzoglou, A. Jablonski, and M. Beigl. A seq2seq learning approach for modeling semantic trajectories and predicting the next location. In *SIGSPATIAL/GIS*, 2018.

[121] S. Keele et al. Guidelines for performing systematic literature reviews in software engineering. Technical report, Technical report, Ver. 2.3 EBSE Technical Report. EBSE, 2007.

[122] U. Khandelwal, H. He, P. Qi, and D. Jurafsky. Sharp nearby, fuzzy far away: How neural language models use context. *arXiv preprint arXiv:1805.04623*, 2018.

[123] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. *Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications*, pages 1–13, 2006.

[124] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. Laurila. Towards rich mobile phone datasets: Lausanne data collection campaign. *Proc. ICPS, Berlin*, 2010.

[125] N. Kiukkonen, J. Blom, O. Dousse, D. Gatica-Perez, and J. K. Laurila. Towards rich mobile phone datasets: Lausanne data collection campaign. 2010.

[126] I. Kontoyiannis, P. H. Algoet, Y. M. Suhov, and A. J. Wyner. Nonparametric entropy estimation for stationary processes and random fields, with applications to english text. *IEEE Transactions on Information Theory*, 44(3):1319–1327, 1998.

[127] A. Krause and E. Horvitz. A utility-theoretic approach to privacy and personalization. In *AAAI*, volume 8, pages 1181–1188, 2008.

[128] R. Krüger. Visual analytics of human mobility behavior. 2017.

[129] J. Krumm. Inference attacks on location tracks. In *Pervasive*, 2007.

[130] K. A. Küçük, A. Paverd, A. Martin, N. Asokan, A. Simpson, and R. Ankele. Exploring the use of intel sgx for secure many-party applications. In *Proceedings of the 1st Workshop on System Software for Trusted Execution*, SysTEX '16, pages 5:1–5:6, New York, NY, USA, 2016. ACM.

[131] V. Kulkarni, B. Chapuis, and B. Garbinato. Privacy-preserving location-based services by using intel sgx. In *HumanSys'17*, 2017.

[132] V. Kulkarni and B. Garbinato. Generating synthetic mobility traffic using rnns. In *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, GeoAI '17, pages 1–4, New York, NY, USA, 2017. ACM.

[133] V. Kulkarni, A. Mahalunkar, B. Garbinato, and J. D. Kelleher. On the inability of markov models to capture criticality in human mobility. *arXiv preprint arXiv:1807.11386*, 2018.

[134] V. Kulkarni, A. Mahalunkar, B. Garbinato, and J. D. Kelleher. Examining the limits of predictability of human mobility. *Entropy*, 21(4):432, 2019.

[135] V. Kulkarni, A. Moro, B. Chapuis, and B. 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, New York, NY, USA, 2017. ACM.

[136] V. Kulkarni, A. Moro, and B. 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.

[137] J. K. Laurila, D. Gatica-Perez, I. Aad, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, M. Miettinen, et al. The mobile data challenge: Big data for mobile computing research. In *Pervasive Computing*, number EPFL-CONF-192489, 2012.

[138] J. K. Laurila, D. Gatica-Perez, I. Aad, B. J., O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen. The mobile data challenge: Big data for mobile computing research. In *Pervasive Computing*, 2012.

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

[140] T. Le Hung, M. Michele, Catasta an Lucas Kelsey, and A. Karl. Next place prediction using mobile data. Mobile Data Challenge by Nokia, 2012.

[141] E. Leal, L. Gruenwald, J. Zhang, and S. You. Towards an efficient top-k trajectory similarity query processing algorithm for big trajectory data on gpgpus. *2016 IEEE International Congress on Big Data (BigData Congress)*, pages 206–213, 2016.

[142] Y. LeCun, Y. Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.

[143] S. Lee and K. C. 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.

[144] A. Lesne, J.-L. Blanc, and L. Pezard. Entropy estimation of very short symbolic sequences. *Physical Review E*, 79(4):046208, 2009.

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

[146] Z. Li, B. Ding, J. Han, R. Kays, and P. Nye. Mining periodic behaviors for moving objects. In *KDD*, 2010.

[147] Z. Li and J. Han. *Mining Periodicity from Dynamic and Incomplete Spatiotemporal Data*, pages 41–81. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[148] H. W. Lin and M. Tegmark. Critical behavior from deep dynamics: a hidden dimension in natural language. *arXiv preprint arXiv:1606.06737*, 2016.

[149] H. W. Lin and M. Tegmark. Critical behavior in physics and probabilistic formal languages. *Entropy*, 19(7), 2017.

[150] M. Lin, W. J. Hsu, and Z. Q. 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.

[151] Z. C. Lipton. A critical review of recurrent neural networks for sequence learning. *CoRR*, abs/1506.00019, 2015.

[152] Q. Liu, S. Wu, L. Wang, and T. Tan. Predicting the next location: A recurrent model with spatial and temporal contexts. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[153] T. Louail, M. Lenormand, O. G. C. Ros, M. Picornell, R. Herranz, E. Frias-Martinez, J. J. Ramasco, and M. Barthelemy. From mobile phone data to the spatial structure of cities. In *Scientific reports*, 2014.

[154] X. Lu, E. Wetter, N. Bharti, A. J. Tatem, and L. Bengtsson. Approaching the limit of predictability in human mobility. *Scientific reports*, 3:2923, 2013.

[155] C. Ma, S. Wan, B. Han, and H. Gui. A framework for hybrid location prediction via decision tree classification. In *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2202–2207. IEEE, 2018.

[156] H. R. Madala and A. G. Ivakhnenko. *Inductive learning algorithms for complex systems modeling*, volume 368. CRC press Boca Raton, 1994.

[157] A. Mahalunkar and J. D. Kelleher. Using regular languages to explore the representational capacity of recurrent neural architectures. *arXiv preprint arXiv:1808.05128*, 2018.

[158] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *CoRR*, abs/1412.6632, 2014.

[159] W. Mathew, R. Raposo, and B. Martins. Predicting future locations with hidden markov models. In *UbiComp*, 2012.

[160] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

[161] S. Merity, C. Xiong, J. Bradbury, and R. Socher. Pointer sentinel mixture models. *CoRR*, abs/1609.07843, 2016.

[162] Microsoft. GeoLife Dataset. https://www.microsoft.com/en-us/download/, 2012. [Online; accessed 26-July-2018].

[163] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

[164] M. F. Mokbel, L. Alarabi, J. Bao, A. Eldawy, A. Magdy, M. Sarwat, E. Waytas, and S. Yackel. Mntg: an extensible web-based traffic generator. In *International Symposium on Spatial and Temporal Databases*, pages 38–55. Springer, 2013.

[165] M. F. Mokbel and W. G. Aref. Sole: scalable on-line execution of continuous queries on spatio-temporal data streams. *The VLDB Journal—The International Journal on Very Large Data Bases*, 17(5):971–995, 2008.

[166] S. B. Mokhtar, A. Boutet, L. Bouzouina, P. Bonnel, O. Brette, L. Brunie, M. Cunche, S. D. 'Alu, V. Primault, P. Raveneau, H. Rivano, and R. Stanica. Priva'mov: Analysing human mobility through multi-sensor datasets. 2017.

[167] S. B. Mokhtar, A. Boutet, L. Bouzouina, P. Bonnel, O. Brette, L. Brunie, M. Cunche, S. D'Alu, V. Primault, P. Raveneau, et al. Priva'mov: Analysing human mobility through multi-sensor datasets. In *NetMob 2017*, 2017.

[168] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: a location predictor on trajectory pattern mining. In *KDD*, 2009.

[169] A. Monreale, F. Pinelli, R. Trasarti, and F. Giannotti. Wherenext: A location predictor on trajectory pattern mining. KDD, 2009, pages 637–646, 2009.

[170] R. Montoliu and D. Gatica-Perez. Discovering human places of interest from multimodal mobile phone data. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, page 12. ACM, 2010.

[171] R. Montoliu and D. Gatica-Perez. Discovering human places of interest from multimodal mobile phone data. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, page 12. ACM, 2010.

[172] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz. Analysis of the clustering properties of the hilbert space-filling curve. *IEEE Trans. Knowl. Data Eng.*, 13:124–141, 2001.

[173] A. Moro and B. 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.

[174] A. Moro and B. Garbinato. A location privacy estimator based on spatio-temporal location uncertainties. pages 322–337, 05 2017.

[175] A. Moro, B. Garbinato, and V. Chavez-Demoulin. Discovering demographic data of users from the evolution of their spatio-temporal entropy, 2018.

[176] T. Nagler and T. Vatter. rvinecopulib: high performance algorithms for vine copula modeling, 2018.

[177] R. Nallapati, B. Zhou, C. N. dos Santos, and y. Çaglar Gülçehre and Bing Xiang, booktitle=CoNLL. Abstractive text summarization using sequence-to-sequence rnns and beyond.

[178] M. A. Nascimento, D. Pfoser, and Y. Theodoridis. Synthetic and real spatiotemporal datasets. *IEEE Data Eng. Bull.*, 26(2):26–32, 2003.

[179] M. Nasr, R. Shokri, and A. Houmansadr. Machine learning with membership privacy using adversarial regularization. *arXiv preprint arXiv:1807.05852*, 2018.

[180] M. E. Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351, 2005.

[181] A. D. Nguyen, P. Sénac, V. Ramiro, and M. Diaz. Steps - an approach for human mobility modeling. In *Networking*, 2011.

[182] O. Ossama and H. M. Mokhtar. Similarity search in moving object trajectories. In *Proceedings of the 15th International Conference on Management of Data*, pages 1–6. Citeseer, 2009.

[183] K. Ouyang, R. Shokri, D. S. Rosenblum, and W. Yang. A non-parametric generative model for human trajectories. In *IJCAI*, pages 3812–3817, 2018.

[184] C. Parent, S. Spaccapietra, C. Renso, G. L. Andrienko, N. V. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. A. F. de Macêdo, N. Pelekis, Y. Theodoridis, and Z. Yan. Semantic trajectories modeling and analysis. *ACM Comput. Surv.*, 45:42:1–42:32, 2013.

[185] P. N. Pathirana, A. V. Savkin, and S. Jha. Mobility modelling and trajectory prediction for cellular networks with mobile base stations. In *MobiHoc*, 2003.

[186] D. J. Patterson, L. Liao, D. Fox, and H. A. Kautz. Inferring high-level behavior from low-level sensors. In *UbiComp*, 2003.

[187] N. Pelekis, S. Sideridis, P. Tampakis, and Y. Theodoridis. Hermoupolis: a semantic trajectory generator in the data science era. *SIGSPATIAL Special*, 7(1):19–26, 2015.

[188] R. Pellungrini, L. Pappalardo, F. Pratesi, and A. Monreale. A data mining approach to assess privacy risk in human mobility data. In *TIST*, 2017.

[189] Y. Peng, P. A. Flach, C. Soares, and P. Brazdil. Improved dataset characterisation for meta-learning. In *Discovery Science*, 2002.

[190] A. Pentland. Reality mining of mobile communications: Toward a new deal on data. *The Global Information Technology Report 2008–2009*, 1981, 2009.

[191] F. Pérez-Cruz. Kullback-leibler divergence estimation of continuous distributions. *2008 IEEE International Symposium on Information Theory*, pages 1666–1670, 2008.

[192] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer. Global and local state context prediction. In *Artificial Intelligence in Mobile Systems*, 2003.

[193] J. Petzold, F. Bagci, W. Trumler, and T. Ungerer. Comparison of different methods for next location prediction, 2006. pages 909–918. Springer, 2006.

[194] B. Pfahringer, H. Bensusan, and C. G. Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *ICML*, pages 743–750, 2000.

[195] A.-K. Pietilainen and C. Diot. CRAWDAD dataset thlab/sigcomm2009 (v. 2012-07-15). Downloaded from https://crawdad.org/thlab/sigcomm2009/20120715, July 2012.

[196] R. Pires, D. Gavril, P. Felber, E. Onica, and M. Pasin. A lightweight mapreduce framework for secure processing with sgx. In *Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 1100–1107. IEEE Press, 2017.

[197] R. Pires, M. Pasin, P. Felber, and C. Fetzer. Secure content-based routing using intel software guard extensions. In *Middleware*, 2016.

[198] U. S. Potluri, A. Madanayake, R. J. Cintra, F. M. Bayer, S. Kulasekera, and A. Edirisuriya. Improved 8-point approximate dct for image and video compression requiring only 14 additions. *IEEE Trans. on Circuits and Systems*, 61-I:1727–1740, 2014.

[199] B. Prabhala, J. Wang, B. Deb, T. L. Porta, and J. Han. Leveraging periodicity in human mobility for next place prediction. In *WCNC*, pages 2665–2670, 2014.

[200] V. V. Prelov and E. C. van der Meulen. Mutual information, variation, and fano's inequality. *Problems of Information Transmission*, 44(3):185–197, 2008.

[201] A. Pyrgelis, C. Troncoso, and E. De Cristofaro. Knock knock, who's there? membership inference on aggregate location data. *arXiv preprint arXiv:1708.06145*, 2017.

[202] S.-M. Qin, H. Verkasalo, M. Mohtaschemi, T. Hartonen, and M. Alava. Patterns, entropy, and predictability of human mobility and life. *PloS one*, 7(12):e51353, 2012.

[203] V. Rastogi and S. Nath. Differentially private aggregation of distributed time-series with transformation and encryption. In *SIGMOD 2010*, pages 735–746.

[204] I. Rhee, M. Shin, S. Hong, K. Lee, S. J. Kim, and S. Chong. On the levy-walk nature of human mobility. *IEEE/ACM Transactions on Networking*, 19:630–643, 2011.

[205] D. E. Riedel, S. Venkatesh, and W. Liu. Recognising online spatial activities using a bioinformatics inspired sequence alignment approach. *Pattern Recognition*, 41:3481–3492, 2008.

[206] N. K. Saini and A. Trivedi. Refined cluster based mobility prediction with weighted algorithm. In *CICN*, pages 350–354, Nov 2010.

[207] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee. Recent advances in recurrent neural networks. *arXiv preprint arXiv:1801.01078*, 2017.

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

[209] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[210] C. Schreckenberger, S. Beckmann, and C. Bartelt. Next place prediction: A systematic literature review. In *Proceedings of the 2nd ACM SIGSPATIAL Workshop on Prediction of Human Mobility*, pages 37–45. ACM, 2018.

[211] T. Schürmann. Scaling behaviour of entropy estimates. *Journal of Physics A: Mathematical and General*, 35(7):1589, 2002.

[212] C. E. Shannon. Prediction and entropy of printed english. *Bell Labs Technical Journal*, 30(1):50–64, 1951.

[213] C. E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55, 2001.

[214] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. *CoRR*, abs/1511.04119, 2015.

[215] P. C. Shields. Universal redundancy rates do not exist. *IEEE transactions on information theory*, 39(2):520–524, 1993.

[216] R. Shokri, G. Theodorakopoulos, J.-Y. L. Boudec, and J.-P. Hubaux. Quantifying location privacy. *2011 IEEE Symposium on Security and Privacy*, pages 247–262, 2011.

[217] R. Shokri, G. Theodorakopoulos, J.-Y. Le Boudec, and J.-P. Hubaux. Quantifying location privacy. In *Security and privacy (sp), 2011 ieee symposium on*, pages 247–262. IEEE, 2011.

[218] N. Shoval, G. K. Auslander, T. Freytag, R. Landau, F. Oswald, U. Seidl, H.-W. Wahl, S. Werner, and J. Heinik. The use of advanced tracking technologies for the analysis of mobility in alzheimer's disease and related cognitive diseases. *BMC Geriatrics*, 8(1):7, Mar 2008.

[219] H. Si, Y. Wang, J. Yuan, and X. Shan. Mobility prediction in cellular network using hidden markov model. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–5. IEEE, 2010.

[220] D. Sikeridis, I. Papapanagiotou, and M. Devetsikiotis. CRAWDAD dataset unm/blebeacon (v. 2019-03-12). Downloaded from https://crawdad.org/unm/blebeacon/20190312, Mar. 2019.

[221] A. Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publications de L'Institut de Statistique de L'Université de Paris*, 8:229–231, 1959.

[222] G. Smith, R. Wieser, J. Goulding, and D. Barrack. A refined limit on the predictability of human mobility. In *Pervasive computing and communications (PerCom), 2014 IEEE international conference on*, pages 88–94. IEEE, 2014.

[223] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pages 4424–4434, 2017.

[224] W.-S. Soh and H. S. Kim. Qos provisioning in cellular networks based on mobility prediction techniques. *IEEE Communications Magazine*, 41:86–92, 2003.

[225] F. Somaa, C. Adjih, I. E. Korbi, and L. A. Saidane. A bayesian model for mobility prediction in wireless sensor networks. In *2016 International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN)*, pages 1–7, Nov 2016.

[226] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327 5968:1018–21, 2010.

[227] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[228] A. Stopczynski, V. Sekara, P. Sapiezynski, A. Cuttone, M. M. Madsen, J. E. Larsen, and S. Lehmann. Measuring large-scale social networks with high resolution. *PloS one*, 9(4):e95978, 2014.

[229] J. A. Storer. *Data compression: methods and theory*. Computer Science Press, Inc., 1987.

[230] W. Su, S.-J. Lee, and M. Gerla. Mobility prediction and routing in ad hoc wireless networks. *Int. Journal of Network Management*, 11:3–30, 2001.

[231] I. Sutskever, J. Martens, and G. E. Hinton. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1017–1024, 2011.

[232] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[233] B. Tang, M. L. Yiu, K. Mouratidis, and K. Wang. Efficient motif discovery in spatial trajectories using discrete fréchet distance. EDBT, 2017.

[234] T. Tango and K. Takahashi. A flexibly shaped spatial scan statistic for detecting clusters. *International Journal of Health Geographics*, 4:11 – 11, 2005.

[235] Y. Theodoridis, J. R. Silva, and M. A. Nascimento. On the generation of spatiotemporal datasets. In *International Symposium on Spatial Databases*, pages 147–164. Springer, 1999.

[236] A. Thomason, N. Griffiths, and V. Sanchez. Identifying locations from geospatial trajectories. *Journal of Computer and System Sciences*, 82(4):566–581, 2016.

[237] A. Thomason, N. Griffiths, and V. Sanchez. Identifying locations from geospatial trajectories. *Journal of Computer and System Sciences*, 82(4):566–581, 2016.

[238] R. J. Thorpe, E. M. Simonsick, J. S. Brach, H. Ayonayon, S. Satterfield, T. B. Harris, M. Garcia, S. B. Kritchevsky, A. Health, and B. C. Study. Dog ownership, walking behavior, and maintained mobility in late life. *Journal of the American Geriatrics Society*, 54(9):1419–1424, 2006.

[239] I. Trajkovic, C. Reller, M. Wolf, and H.-A. Loeliger. Modelling and filtering almost periodic signals by time-varying fourier series with application to near-infrared spectroscopy. *2009 17th European Signal Processing Conference*, pages 632–636, 2009.

[240] L. H. Tran, M. Catasta, L. K. McDowell, and K. Aberer. Next place prediction using mobile data. In *Proceedings of the Mobile Data Challenge Workshop (MDC 2012)*, number CONF, 2012.

[241] K. S. Trivedi. *Probability & statistics with reliability, queuing and computer science applications*. John Wiley & Sons, 2008.

[242] J. Van Bulck, M. Minkin, O. Weisse, D. Genkin, B. Kasikci, F. Piessens, M. Silberstein, T. F. Wenisch, Y. Yarom, and R. Strackx. Foreshadow: Extracting the keys to the intel {SGX} kingdom with transient out-of-order execution. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 991–1008, 2018.

[243] B. Vegetabile, J. Molet, T. Z. Baram, and H. Stern. Estimating the entropy rate of finite markov chains with application to behavior studies. *arXiv preprint arXiv:1711.03962*, 2017.

[244] S. Vhaduri and C. Poellabauer. Cooperative discovery of personal places from location traces. *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–9, 2016.

[245] Y. Virkar and A. Clauset. Power-law distributions in binned empirical data. *The Annals of Applied Statistics*, pages 89–119, 2014.

[246] C.-C. Wang, C. E. Thorpe, S. Thrun, M. Hebert, and H. F. Durrant-Whyte. Simultaneous localization, mapping and moving object tracking. *I. J. Robotics Res.*, 26:889–916, 2007.

[247] H. Wang, Z. Yang, and Y. Shi. Next location prediction based on an adaboost-markov model of mobile users. *Sensors*, 19(6):1475, 2019.

[248] P. Wang, F. Sun, D. Wang, J. Tao, X. Guan, and A. Bifet. Inferring demographics and social networks of mobile device users on campus from ap-trajectories. In *WWW*, 2017.

[249] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural computation*, 8(7):1341–1390, 1996.

[250] A. D. Wyner and J. Ziv. Some asymptotic properties of the entropy of a stationary ergodic data source with applications to data compression. *IEEE Transactions on Information Theory*, 35(6):1250–1258, 1989.

[251] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin. Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1241–1250. International World Wide Web Conferences Steering Committee, 2017.

[252] J. Xu and R. H. Güting. Mwgen: a mini world generator. In *Mobile Data Management (MDM), 2012 IEEE 13th International Conference on*, pages 258–267. IEEE, 2012.

[253] X.-Y. Yan, X.-P. Han, B.-H. Wang, and T. Zhou. Diversity of individual mobility patterns and emergence of aggregated scaling laws. *Scientific reports*, 3:2678, 2013.

[254] D. Yang, D. Zhang, Z. Yu, and Z. Yu. Fine-grained preference-aware location search leveraging crowdsourced digital footprints from lbsns. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 479–488. ACM, 2013.

[255] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu. Modeling user activity preference by leveraging user spatial temporal characteristics in lbsns. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1):129–142, 2015.

[256] J. Yang, J. Xu, M. Xu, N. Zheng, and Y. Chen. Predicting next location using a variable order markov model. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoStreaming*, pages 37–42. ACM, 2014.

[257] L. Yu, W. Zhang, J. Wang, and Y. Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.

[258] S. Yu and H. Kobayashi. A hidden semi-markov model with missing data and multiple observation sequences for mobility tracking. *Signal Processing*, 83:235–250, 2003.

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

[260] H. Zang and J. 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.

[261] B. Zhang, Y. Shen, Y. Zhu, and J. Yu. A gpu-accelerated framework for processing trajectory queries. *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1037–1048, 2018.

[262] J. Zhao, J. Xu, R. Zhou, P. Zhao, C. Liu, and F. Zhu. On prediction of user destination by sub-trajectory understanding: A deep learning based approach. In *CIKM*, 2018.

[263] K. Zhao, Z. Tu, F. Xu, Y. Li, P. Zhang, D. Pei, L. Su, and D. Jin. Walking without friends: Publishing anonymized trajectory dataset without leaking social relationships. *IEEE Transactions on Network and Service Management*, 2019.

[264] Q. Zhao, Y. Shi, Q. Liu, and P. Fränti. A grid-growing clustering algorithm for geo-spatial data. *Pattern Recognition Letters*, 53:77–84, 2015.

[265] Z.-D. Zhao, S.-M. Cai, and Y. Lu. Non-markovian character in human mobility: Online and offline. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 25(6):063106, 2015.

[266] Y. Zheng, X. Xie, and W.-Y. Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33(2):32–39, 2010.

[267] Y. Zheng, X. Xie, and W.-Y. Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Eng. Bull.*, 33:32–39, 2010.

[268] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM, 2009.

[269] Y. Zhong, N. J. Yuan, W. Zhong, F. Zhang, and X. Xie. You are where you go: Inferring demographic attributes from location check-ins. In *Proceedings of the eighth ACM international conference on web search and data mining*, pages 295–304. ACM, 2015.

[270] Y. Zhong, N. J. Yuan, W. Zhong, F. Zhang, and X. Xie. You are where you go: Inferring demographic attributes from location check-ins. In *WSDM*, 2015.

[271] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen. Discovering personal gazetteers: an interactive clustering approach. In *Proceedings of the 12th annual ACM international workshop on Geographic information systems*, pages 266–273. ACM, 2004.

[272] W. Zhu, C. Zhang, S. Yao, X. Gao, and J. Han. A spherical hidden markov model for semantics-rich human mobility modeling. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[273] J. G. Zilly, R. K. Srivastava, J. Koutník, and J. Schmidhuber. Recurrent highway networks. In *ICML*, 2017.

[274] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on information theory*, 23(3):337–343, 1977.

[275] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *IEEE transactions on Information Theory*, 24(5):530–536, 1978.

List of Figures

List of Tables