University College London

Doctoral Thesis

Evaluating Privacy-Friendly Mobility Analytics on Aggregate Location Data

Author:
Apostolos Pyrgelis

Supervisor:
Dr. Emiliano De Cristofaro

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy in the

Information Security Research Group
Department of Computer Science

February 18, 2019

Declaration of Authorship

I, Apostolos PYRGELIS, declare that this thesis titled, “Evaluating Privacy-Friendly Mobility Analytics on Aggregate Location Data” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Abstract

Evaluating Privacy-Friendly Mobility Analytics on Aggregate Location Data

by Apostolos PYRGELIS

Doctor of Philosophy

Information Security Research Group

Department of Computer Science

University College London

Information about people's movements and the locations they visit enables a wide number of mobility analytics applications, e.g., real-time traffic maps or urban planning, aiming to improve quality of life in modern smart-cities. Alas, the availability of users' fine-grained location data reveals sensitive information about them such as home and work places, lifestyles, political or religious inclinations. In an attempt to mitigate this, aggregation is often employed as a strategy that allows analytics and machine learning tasks while protecting the privacy of individual users' location traces.

In this thesis, we perform an end-to-end evaluation of crowdsourced privacy-friendly location aggregation aiming to understand its usefulness for analytics as well as its privacy implications towards users who contribute their data. First, we present a time-series methodology which, along with privacy-friendly crowdsourcing of aggregate locations, supports mobility analytics such as traffic forecasting and mobility anomaly detection. Next, we design quantification frameworks and methodologies that let us reason about the privacy loss stemming from the collection or release of aggregate location information against knowledgeable adversaries that aim to infer users' profiles, locations, or membership. We then utilize these frameworks to evaluate defenses ranging from generalization and hiding, to differential privacy, which can be employed to prevent inferences on aggregate location statistics, in terms of privacy protection as well as utility loss towards analytics tasks. Our results highlight that, while location aggregation is useful for mobility analytics, it is a weak privacy protection mechanism in this setting and that additional defenses can only protect privacy if some statistical utility is sacrificed. Overall, the tools presented in this thesis can be used by providers who desire to assess the quality of privacy protection before data release and its results have several implications about current location data practices and applications.

Impact Statement

The research presented in this thesis revolves around the usefulness of location aggregation as a means to enable mobility analytics in modern cities, and investigates its privacy implications for users who contribute their data to the process. As a result, the results of this work impact various entities of our data-driven society such as researchers, technology vendors, legal groups, as well as individuals.

First, the privacy frameworks presented in this work demonstrate that aggregation is a *weak* protection mechanism in the setting of location crowdsourcing and that additional defenses are required to guarantee users' end-to-end privacy. Our evaluation of such defenses shows that it is challenging to achieve strong privacy without sacrificing the statistical utility of the data. Nonetheless, we are hopeful that the methods contributed in this thesis will help the research community to develop novel defenses that achieve better balance in this inherent tradeoff between privacy and utility. Accordingly, we believe that our work will inspire researchers to evaluate the privacy protection offered by aggregation in other data domains, such as the medical, web search, network traffic, and smart metering, ones.

Furthermore, this thesis's results have real-world implications on current location data practices and applications. More precisely, we demonstrate that mobility analytics can be enabled by privacy-friendly crowdsourced aggregate location statistics with minimal overhead on users' devices. As a result, privacy-by-design is a viable option for providers who are interested in deploying data collection services that benefit our society while minimizing the exposure of users' sensitive data. While such a practice will only be a *first step* towards privacy-conscious data collection approaches, we are confident that advances in research will spawn new techniques providing stronger privacy protection. Moreover, our results show that third-parties whose services rely on aggregate location statistics should be careful about the privacy expectations they create to their customers and could employ the tools presented in this thesis to assess the privacy protection before publishing such data for the purpose of analytics.

Finally, we are hopeful that this dissertation will raise wider awareness about the privacy implications of aggregate location statistics and location data *in general*. More precisely, our experimental outcomes should draw the attention of users who agree to share their location data with service providers under the promise that it will be utilized only in anonymized and aggregated form. Similarly, legal entities should update their policies regarding the proper ways to collect, use, and share, location data as well as inform regulations about general privacy-friendly data practices. We are optimistic that such legal frameworks in combination with user education around privacy issues will allow our society to embrace and safely adopt modern data-driven technologies.

Acknowledgements

I wish to express my sincere appreciation to all those who have contributed to this thesis and supported me throughout the realization of this research.

First and foremost, I would like to express my gratitude to my supervisor Emiliano De Cristofaro for being a constant source of inspiration for the last couple of years. His guidance, advice, and support, have been essential for this work, while his insights and challenges have made this doctoral journey more than enjoyable.

I would also like to thank Gordon Ross and Mirco Musolesi for providing useful feedback and helpful suggestions at various stages of this work, as well as George Danezis and Alastair Beresford for serving as my viva examiners. Furthermore, I have been extremely lucky to collaborate with and learn from great researchers among which are: Carmela Troncoso, Luca Melis, Ilias Leontiadis, Joan Serrà, Nicolas Kourtellis, and Claudio Soriente.

I would like to thank everyone in the Information Security Research Group at UCL for creating an ideal and stimulating academic environment, where working and interacting was a real pleasure. I would also desire to acknowledge the Alan Turing Institute and Telefonica Research for hosting me as part of internship programs and giving me the opportunity to work in their interdisciplinary environments.

This research would not have been possible without my family. They have always supported every decision of my life in so many different ways. I am also indebted to my friends and all the great people that I have met in London and around the world during these years. Last but not least, I would like to thank Goml for her unconditional love and support, for making life fun every day, and reminding me of what is really important.

* This research was supported by a Xerox University Affairs Committee grant on “Secure Collaborative Analytics” and University College London (UCL).

Contents

Declaration of Authorship iii

Abstract v

Impact Statement vii

Acknowledgements ix

List of Figures xv

List of Tables xix

List of Abbreviations xxi

1 Introduction 1

1.1 Research Questions and Contributions . . . 3

1.2 Thesis Outline . . . 4

1.3 Collaboration . . . 6

2 Background 7

2.1 Machine Learning . . . 7

2.1.1 Regression . . . 8

2.1.2 Classification . . . 10

2.1.3 Dimensionality Reduction . . . 12

2.2 Differential Privacy . . . 13

2.2.1 Definition . . . 13

2.2.2 Mechanisms . . . 14

2.3 Metrics . . . 16

2.3.1 Error Metrics 16
2.3.2 Correlation Coefficients 16
2.3.3 Statistical Distance Measures 18

3 Literature Review 21
3.1 Mobility Analytics 21
3.1.1 Mobility Forecasting 21
3.1.2 Detecting Mobility Anomalies 23
3.2 Privacy-Preserving Location Statistics 24
3.2.1 Privacy-Preserving Aggregation 24
3.2.2 Private Location Data Publishing 26
3.3 Membership Inference 27
3.3.1 Membership Inference on Aggregate Statistics 27
3.3.2 Membership Inference on Machine Learning Models 28
3.4 Location Privacy 28
3.4.1 Privacy in Location Based Services (LBS) 28
3.4.2 Privacy of Mobility Trajectories 30
3.4.3 Quantifying Location Privacy 31
3.4.4 Aggregate Location Privacy 32

4 Datasets 33
4.1 Notation 33
4.2 Transport for London 34
4.3 San Francisco Cabs 36

5 Privacy-Friendly Mobility Analytics on Aggregate Location Data 39
5.1 Forecasting Traffic in Regions of Interest 40
5.1.1 Removing Seasonality 41
5.1.2 Predicting Traffic Volumes in ROIs 41
5.2 Detecting Traffic Anomalies 42
5.3 Enhancing Predictions in Presence of Anomalies 46
5.4 Crowdsourcing Privacy-Friendly Mobility Analytics 50

5.4.1 Data Collection . . . 51
5.4.2 Experimental Evaluation . . . 52

6 Quantifying Privacy Leakage from Aggregate Location Data 57

6.1 Evaluating Aggregation Based Location Privacy . . . 58
6.1.1 Framework Description . . . 58
6.1.2 Adversarial Prior Knowledge . . . 61
6.1.3 Inference Strategies . . . 64
6.2 Evaluation of Raw Aggregate Location Time-Series . . . 67
6.2.1 Training and Testing Data . . . 67
6.2.2 User Profiling . . . 67
6.2.3 User Localization . . . 71
6.2.4 Privacy Implications of Regular Mobility Patterns . . . 75
6.3 Evaluation of Defenses . . . 76
6.3.1 Privacy and Utility Metrics . . . 77
6.3.2 Output Perturbation . . . 78
6.3.3 Input Perturbation . . . 80
6.3.4 Discussion . . . 82

7 Membership Inference on Aggregate Location Data 85

7.1 Modeling Membership Inference on Aggregate Location Data . . . 86
7.1.1 Notation . . . 86
7.1.2 Distinguishability Game . . . 87
7.1.3 Adversarial Prior Knowledge . . . 88
7.1.4 Distinguishing Function . . . 90
7.1.5 Privacy Metric . . . 90
7.2 Experiment Design . . . 91
7.3 Membership Inference on Raw Aggregate Location Data . . . 93
7.3.1 Subset of Locations . . . 94
7.3.2 Participation in Past Groups . . . 97
7.3.3 Length of Inference Period . . . 102

7.3.4 Discussion . . . 104

7.4 Evaluation of Defenses . . . 105

7.4.1 Experiment Design . . . 105

7.4.2 Results . . . 108

7.4.3 Discussion . . . 110

8 Understanding Membership Inference On Aggregate Location Data 113

8.1 Understanding Membership Inference on Aggregate Locations . . . 114

8.1.1 Experiment Setup . . . 114

8.1.2 Subset of Locations . . . 116

8.1.3 Participation in Past Groups . . . 122

8.1.4 Discussion . . . 125

8.2 Evaluation of Defenses . . . 126

8.2.1 Experiment Design . . . 127

8.2.2 Results . . . 130

8.2.3 Discussion . . . 140

9 Conclusion 143

9.1 Discussion . . . 144

9.2 Limitations . . . 145

9.3 Future Work . . . 146

Bibliography 149

A Privacy-Preserving Data Aggregation 171

B Additional Experimental Results for Chapter 6 175

B.1 User Profiling . . . 175

B.2 User Localization . . . 179

List of Figures

4.1 TFL dataset – Hourly traffic volumes of two stations: (a) Canary Wharf and (b) Clapham Common. 34

4.2 SFC dataset – Hourly traffic volumes of two ROIs with ids: (a) 7160 and (b) 8554. 36

4.3 # of cab events on the 100×100 San Francisco grid (May 19 – June 8, 2008). . . . 37

5.1 TFL: Green Park station's time-series without (a) and with (b) de-seasonalization. 40

5.2 Hourly traffic forecasts for TFL's Green Park station on March 25: (a) predictions with the seasonal ARMA model and (b) forecast error. 43

5.3 Hourly traffic forecasts for SFC's region with id 8755 on June 5: (a) predictions with the seasonal ARMA model and (b) forecast error. 44

5.4 Aggregate time-series, forecasts, and detected anomalies, on TFL's (a) North Greenwich station, March 9–11, and (b) Wembley Park station, March 11–12. . . 45

5.5 Aggregate time-series, forecasts, and detected anomalies, on (a) TFL's Arsenal station, March 19–22, and (b) SFC's region with id 8261, May 31–June 1. 46

5.6 Forecast of passengers exiting TFL's Arsenal station on March 19–21 based on a local ARMA model vs. the VAR model trained with information from correlated stations. 48

5.7 Forecast of passengers exiting TFL's Arnos Grove station on March 20 based on a local ARMA model vs. the VAR model trained with information from Arsenal station. 48

5.8 Taxi volume forecast on SFC region with id 8556 on May 27–28 based on a local ARMA model vs. the VAR model trained with information from correlated ROIs. 49

5.9 Execution times of the encryption phase for increasing number of users per group in the (a) TFL and (b) SFC settings. 53

6.1 FREQ_ROI prior – Adv's profiling error. 68
6.2 ROI_DAY_WEEK prior – Adv's profiling error. 69
6.3 TIME_DAY_WEEK prior – Adv's profiling error. 70
6.4 FREQ_ROI prior – Adv's localization error. 72
6.5 ROI_DAY_WEEK prior – Adv's localization error. 73
6.6 ROI_DAY prior – Adv's hourly profiling error. 75
6.7 Privacy Gain (PG) achieved by output perturbation DP mechanisms on the TFL dataset. 80
6.8 Privacy Gain (PG) achieved by output perturbation DP mechanisms on the SFC dataset. 80
6.9 Privacy Gain (PG) achieved by SpotMe [Que+11] for increasing values of π, on the TFL and SFC datasets. 81

7.1 Distinguishability Game (DG) between the adversary (Adv) and the challenger (Ch), capturing membership inference on aggregate location time-series. The game is parameterized by the set of users (U), the aggregation group size (m), and the inference period (T_I). 88
7.2 TFL, Subset of Locations prior ($\alpha = 0.11$, $|T_I| = 168$) – Adv's performance for different values of m. 95
7.3 Subset of Locations prior – Privacy Loss (PL) for different values of m. 95
7.4 SFC, Subset of Locations prior ($\alpha = 0.2$, $|T_I| = 168$) – Adv's performance for different values of m. 96
7.5 TFL, Same Groups as Released prior ($\beta = 150$, $|T_I| = 168$) – Adv's performance for different values of m. 97
7.6 Same Groups as Released prior – Privacy Loss (PL) for different values of m. . . 98
7.7 SFC, Same Groups as Released prior ($\beta = 150$, $|T_I| = 168$) – Adv's performance for different values of m. 99
7.8 TFL, Different Groups than Released prior (β=300, $|T_I|$=168) – Adv's performance for different values of m. 100
7.9 Different Groups than Released prior – Privacy Loss (PL) for different values of m. 101

7.10 SFC, Different Groups than Released prior (β=300, $|T_I|$=168) – Adv's performance for different values of m. 102

7.11 Same Groups as Released prior ($\beta = 150$) – Adv's performance for variable inference period length ($|T_I|$), on (a) TFL and (b) SFC, and Privacy Loss (PL) on (c) TFL and (d) SFC. 103

7.12 Privacy Gain (PG) achieved by differentially private mechanisms with different values of ϵ, against a MLP classifier trained on *raw* aggregates and tested on *noisy* aggregates. 109

7.13 Privacy Gain (PG) achieved by differentially private mechanisms with different values of ϵ, against a MLP classifier trained and tested on *noisy* aggregates. 110

8.1 Membership inference performance with the Subset of Locations prior: Features vs. PCA. 115

8.2 TFL, Subset of Locations prior ($\alpha = 0.11, m = 1,000, |T_I| = 168$): Aggregate spatio-temporal correlation over the top 2 components per victim: (a) original heatmap, ascending-order sorted heatmap by location and time slot popularity computed on (b) all victims, (c) top 10%, and (d) bottom 10% of distinguishable victims. 117

8.3 TFL, Subset of Locations prior ($\alpha = 0.11, m = 1,000$, $|T_I| = 168$): Aggregate normalized frequency of time slots over the inference week. 118

8.4 SFC, Subset of Locations prior ($\alpha = 0.2, m = 100$, $|T_I| = 168$): Aggregate spatio-temporal correlation over the top 5 components per victim: (a) original heatmap, ascending-order sorted heatmap by location and time slot popularity computed on (b) all victims, (c) top 10%, and (d) bottom 10% of distinguishable victims. 120

8.5 SFC, Subset of Locations prior ($\alpha = 0.2, m = 100, |T_I| = 168$): Aggregate normalized frequency of time slots over the inference week. 121

8.6 Same Groups as Released prior: Aggregate spatio-temporal correlation over the top components per victim for (a) TFL ($\beta = 500, m = 9,500$, $|T_I| = 168$) and (b) SFC ($\beta = 800, m = 500, |T_I| = 168$) datasets. 122

8.7 SFC, Spatial generalization: Privacy Gain (PG). 130

8.8 Temporal generalization: Privacy Gain (PG). 131
8.9 Data generalization: Privacy Gain (PG) with fixed intervals of size x. 133
8.10 Data generalization: Privacy Gain (PG) with adaptive ranges (i.e., using x' buckets). 134
8.11 Suppression: Privacy Gain (PG). 136
8.12 Sampling: Privacy Gain (PG). 137
8.13 FPA perturbation: Privacy Gain (PG). 139
8.14 SFC, Sampling in combination with FPA: Privacy Gain (PG). 140

B.1 ROI_DAY prior – Adv's profiling error. 176
B.2 TIME_DAY prior – Adv's profiling error. 176
B.3 LAST_WEEK (a), LAST_DAY (b), and LAST_HOUR (c), priors – Adv's profiling error on the TFL dataset. 177
B.4 LAST_WEEK (a), LAST_DAY (b), and LAST_HOUR (c), priors – Adv's profiling error on the SFC dataset. 178
B.5 TIME_DAY_WEEK prior – Adv's localization error. 179
B.6 LAST_WEEK (a), LAST_DAY (b), and LAST_HOUR (c), priors – Adv's localization error on the TFL dataset. 180
B.7 LAST_WEEK (a), LAST_DAY (b), and LAST_HOUR (c), priors – Adv's localization error on the SFC dataset. 181

List of Tables

6.1 Notation. 58

6.2 Different ways to build adversarial prior knowledge. 63

6.3 MRE (utility loss) of output perturbation mechanisms on the TFL dataset. . . . 79

6.4 MRE (utility loss) of output perturbation mechanisms on the SFC dataset. . . . 79

6.5 SpotMe [Que+11]: MRE (utility loss) for increasing values of π on the TFL and SFC datasets. 81

7.1 Notation. 87

7.2 MRE (utility loss) of aggregate location time-series with various differentially private mechanisms and ϵ parameter on the TFL dataset. 108

7.3 MRE (utility loss) of aggregate location time-series with various differentially private mechanisms and ϵ parameter on the SFC dataset. 108

8.1 Subset of Locations prior: Feature importance of a Random Forest classifier for the top and bottom 10% distinguishable victims: TFL ($\alpha = 0.11, m = 1,000$, $|T_I| = 168$) and SFC ($\alpha = 0.2, m = 100$, $|T_I| = 168$). 119

8.2 Different Groups than Released prior: Feature importance of a Logistic Regression classifier for the top and bottom 10% distinguishable victims: TFL ($\beta = 500, m = 9,500$, $|T_I| = 168$) and SFC ($\beta = 800, m = 500$, $|T_I| = 168$). 123

8.3 Utility metrics corresponding to a random guess. 129

8.4 SFC, Utility metrics for spatial generalization. 130

8.5 TFL, Utility metrics for temporal generalization. 131

8.6 SFC, Utility metrics for temporal generalization. 132

8.7 TFL, Utility metrics for data generalization with fixed intervals of size x. 133

8.8 SFC, Utility metrics for data generalization with fixed intervals of size x. 134

8.9 TFL, Utility metrics for data generalization with adaptive ranges (i.e., using x' buckets). 135

8.10 SFC, Utility metrics for data generalization with adaptive ranges (i.e., using x' buckets). 135

8.11 TFL, Utility metrics for suppression. 136

8.12 SFC, Utility metrics for suppression. 137

8.13 TFL, Utility metrics for sampling. 138

8.14 SFC, Utility metrics for sampling. 138

8.15 TFL, Utility metrics for FPA perturbation. 139

8.16 SFC, Utility metrics for FPA perturbation. 140

8.17 SFC, Utility metrics for sampling in combination with FPA ($\epsilon = 1.0$). 141

8.18 SFC, Utility metrics for sampling in combination with FPA ($\epsilon = 10.0$). 141

List of Abbreviations

ARMA	**A**uto **R**egressive **M**oving **A**verage
AUC	**A**rea **U**nder the **C**urve
CDR	**C**all **D**etail **R**ecord
DP	**D**ifferential **P**rivacy
EFPAG	**E**nhanced **F**ourier **P**erturbation **A**lgorithm with **G**aussian noise
FN	**F**alse **N**egative
FP	**F**alse **P**ositive
FPA	**F**ourier **P**erturbation **A**lgorithm
FPR	**F**alse **P**ositive **R**ate
GPS	**G**lobal **P**ositioning **S**ystem
GSM	**G**au**s**sian **M**echanism
IBBE	**I**dentify **B**ased **B**roadcast **E**ncryption
IOT	**I**nternet **O**f **T**hings
k-NN	**k** **N**earest **N**eighbors
LBS	**L**ocation **B**ased **S**ervice
LPM	**L**a**p**lacian **M**echanism
LR	**L**ogistic **R**egression
MDD	**M**obility **D**ata **D**onors
MLP	**M**ulti **L**ayer **P**erceptron
NDA	**N**on **D**isclosure **A**greement
PCA	**P**rincipal **C**omponents **A**nalysis
PDF	**P**robability **D**ensity **F**unction
PETs	**P**rivacy **E**nhancing **T**echnologie**s**
PG	**P**rivacy **G**ain

PIR	**P**rivate **I**nformation **R**etrieval
PL	**P**rivacy **L**oss
POI	**P**oint **O**f **I**nterest
PPV	**P**ositive **P**redictive **V**alue
RF	**R**andom **F**orest
RFE	**R**ecursive **F**eature **E**limination
ROI	**R**egion **O**f **I**nterest
SCM	**Simple** **C**ounting **M**echanism
TN	**T**rue **N**egative
TP	**T**rue **P**ositive
TPR	**T**rue **P**ositive **R**ate
VAR	**V**ector **A**uto **R**egression

To GriPy

Chapter 1

Introduction

We live in the era of big data analytics and machine learning, which enable a wide range of valuable applications including recommendations [WWY15], spam filtering [GC09], mobility analytics [Sen+18], personalized health monitoring [Wei+12], autonomous vehicles [Boj+16], etc. The fuel behind such applications is the availability of large-scale data that people constantly generate through their interactions with online services like search engines, social networks, and retail stores, as well as ubiquitous Internet of Things (IOT) devices and installations such as mobile phones, smart watches, intelligent cars, and sensor networks. As a result, the modern economy has become *data-driven*, and as such, data collection and sharing practices are employed by industry vendors as a way to empower their analytics, improve as well as customize their services, and increase their revenue.

Alas, while the availability of big data facilitates useful analytics tasks and applications, it also raises severe privacy issues. That is because people's digital footprints are a source of rich information about various aspects of their lives such as their demographics, habits and interests, home, work, and travel places, relationships, health status, psychological and emotional states [Mar16]. Such privacy concerns highlight the increasing need for technologies and frameworks that aim at protecting users' right to privacy and have drawn the attention of various communities. For instance, the research community has developed *privacy-enhancing technologies* (PETs) [Gol07], which minimize the exposure of personal data during computations while legal groups have established regulations enforcing entities to comply with when it comes to the collection, use, and sharing of personal information [Eur18].

Given the privacy issues raised by big data, a common strategy employed by entities aiming to enable analytics tasks on sensitive inputs is *aggregation*. The idea behind it is that aggregate statistics provide useful insights about a population, while limiting the exposure of information about its individuals. Such an approach is advocated by the research community as advances in secure computation techniques allow efficient aggregation on private data while eliminating the need for trusted third-parties [KDK11; Shi+11; MDD16]. Furthermore, differential privacy [DR+14] can be used to bound the privacy leakage from aggregate statistics and its techniques are also employed for their collection or release, in modern applications. Example cases are those of Google which collects aggregate statistics from users' browsers to detect malicious activity [Bil+14], or Apple which enhances fast typing and searching applications for Mac and iOS devices through crowdsourcing [Gre18; Tan+17].

In this dissertation, we study a specific type of big data that drives a wide range of analytics in the context of modern smart-cities, namely, location data. Information about users' whereabouts has become increasingly available with the development of mobile networks as well as the widespread usage of GPS-enabled, always-on, always-connected devices. Typically, entities with access to users' location data – e.g., telecommunication service providers, location based services, transportation authorities, navigation application providers – utilize or share it for various mobility analytics. The aim of such tasks is to improve the quality of life in modern cities by, e.g., providing real-time traffic statistics [Gar+13], forecasting events [HZS16], and detecting mobility anomalies [Pan+13].

While such intelligent applications are beneficial to our society, the exposure of location data also raises serious privacy concerns. People's locations along with their semantics reveal sensitive information about them such as their home and work places [Kru07], lifestyles [Pan17; Kul14], political or religious inclinations [Ber15]. A possible approach towards mitigating such concerns is *anonymizing* location traces before sharing it for the purpose of analytics, however, a number of research efforts have shown that this is an extremely challenging task as mobility patterns are inherently unique and tied to the subjects that generate them [GP09; DM+13; ZB11].

Hence, similar to other data domains, a common strategy is for entities to collect or release of *aggregate* location statistics for the purpose of mobility analytics. Location aggregation is not only advocated by research efforts as a privacy-friendly solution [Shi+10; Pop+11; Que+11], but it also finds wide applicability in industrial settings. Representative examples are the Uber Movement project [Ube18] which provides aggregate data from its platform for urban planning purposes, or Telefonica's Smart Steps [Tel18] which monetizes footfall statistics through advertising and business analytics. Similarly, applications like Waze [Waz18] crowdsource average driving speeds to build traffic models that improve navigation in modern cities, while CityMapper's Smart Ride collects traces from users' smartphones to rank public transport journeys and identify gaps in commuting networks [Cit18].

1.1 Research Questions and Contributions

The widespread use of aggregation as a practical and privacy-friendly strategy employed to collect or share location data for the purpose of analytics motivates a number of interesting questions that have received little attention from the research community. In particular, one might wonder whether mobility analytics, e.g., in the context of a city, are feasible on aggregate location data crowdsourced from users' devices in a privacy-friendly manner. Moreover, another important question is whether aggregate location statistics *leak* information about individual users contributing their data for the computation of the aggregates. To this end, how is the adversarial model formulated, which is the adversarial goal, and what metrics can capture potential privacy loss, are issues that should be formally addressed. Finally, an open problem is which and what type of defense mechanisms can be used to limit possible privacy loss stemming from aggregate locations, and how much does their deployment reduce the usefulness of the statistics towards the analytics tasks.

In this dissertation, we set to shed light on these research questions. We focus on a specific type of statistic, namely, *aggregate location time-series* indicating the number of people transiting in certain areas, over time. In this setting, we employ two real-world mobility datasets capturing transport patterns in modern cities and we perform an end-to-end evaluation aiming to understand the power and performance of privacy-friendly aggregation as an enabler for mobility analytics as well as its efficiency as a privacy protection mechanism. To this end, the general contributions of this work can be summarized as follows:

1. We demonstrate how privacy-friendly crowdsourced aggregate location time-series can be useful for predictive mobility analytics tasks like traffic forecasting and anomaly detection, in the context of modern smart-cities.

2. We design and implement quantification frameworks and methodologies that allow us to reason about privacy leakage stemming from the collection or release of aggregate location time-series. We consider knowledgeable adversaries who attempt to exploit the aggregate statistics towards different goals that threaten the privacy of individual users such as profiling, localization, and membership inference.

3. We utilize our proposed frameworks to measure the privacy protection offered by defense mechanisms such as generalization, hiding, and differential privacy, that can be applied on aggregate location time-series to limit the success of the considered adversaries towards their goals, and we study the privacy/utility tradeoffs that arise.

1.2 Thesis Outline

The organization of this dissertation and its detailed contributions are:

Chapter 2 introduces preliminary notions, concepts, algorithms, and metrics, that are widely used throughout this thesis. Then, Chapter 3 reviews work related to the research performed in this dissertation, while Chapter 4 describes the mobility datasets that are employed for our experimental evaluations.

In Chapter 5, we investigate the feasibility of mobility analytics on crowdsourced aggregate location data. In particular, we present a methodology based on time-series modeling that enables analytics tasks like traffic forecasting and anomaly detection, in the context of modern smart-cities. Furthermore, we design and experimentally evaluate the computation, communication, and energy overhead, of a mobile application prototype that allows the privacy-friendly collection of aggregate location data from users' devices, using private computation techniques.

Next, Chapter 6 describes a framework geared to reason about privacy loss for individual users stemming from the collection or release of aggregate location statistics. In more detail, we introduce an adversary that aims at *profiling* or *localizing* users, given some prior knowledge about them and the aggregate locations of a certain time period. To this end, we propose a few approaches to build the adversarial prior knowledge and present inference strategies towards her goals. Using the proposed framework, we quantify the privacy leakage originating from raw aggregate location time-series and we evaluate the privacy protection provided by defenses based on input and output perturbation vis-à-vis the error they introduce to the aggregate statistics.

In Chapter 7, we also study privacy loss stemming from aggregate location data, but from a different adversarial perspective. More precisely, we focus on membership inference attacks whereby the goal of the adversary is to infer if a user's location data was used to compute the aggregate statistics. We formalize the problem as a *distinguishability game* and we instantiate the adversarial task with a machine learning classifier trained on the adversarial prior knowledge. We measure the privacy leakage from raw aggregate locations and examine characteristics that affect its performance. Moreover, we study the privacy protection obtained from defense mechanisms that guarantee differential privacy with respect to the utility loss of the aggregate statistics.

Then, Chapter 8 provides a deeper understanding of membership inference attacks in the setting of aggregate location time-series. We employ a dimensionality reduction approach to obtain insights about locations and times that are important for the inference task; then, we investigate the mobility characteristics of those users which are more prone to it than others. We use these insights to limit the success of the inference by informing a variety of defense strategies ranging from generalization and hiding, to differential privacy. For these defense approaches, we evaluate the privacy/utility tradeoffs that they achieve for a wide range of analytics tasks like traffic forecasting, anomaly detection, hotspot discovery, and location labeling.

Finally, in Chapter 9, we conclude the thesis with a discussion about the implications of its results, its limitations, as well as potential avenues for future research.

1.3 Collaboration

The content presented in this thesis has been co-authored with other researchers and most of it has been published and presented at Computer Science conferences. More specifically, the work of Chapter 5 has been done in collaboration with my supervisor Dr. Emiliano De Cristofaro, my second supervisor Dr. Gordon Ross, and was published at ACM SIGSPATIAL 2016 [PDR16]. Then, the research presented in Chapters 6, 7, and 8, has been conducted along with Prof. Carmela Troncoso and Dr. Emiliano De Cristofaro. In particular, work of Chapter 6 was published at PETS 2017 [PTD17], while that of Chapter 7 at NDSS 2018 [PTD18], where it also received the *Distinguished Paper Award*. Finally, the work discussed in Chapter 8 is currently under submission.

Chapter 2

Background

In this chapter, we introduce preliminary notions, algorithms, concepts, and metrics, which are widely used throughout this thesis. We first review machine learning algorithms and models that we employ for the purpose of realizing mobility analytics on aggregate location time-series or evaluating their privacy implications for individual users contributing their data to the aggregation process. Subsequently, we define the notion of differential privacy and describe widely used mechanisms that achieve its guarantees. Finally, we discuss about metrics that we use in this thesis to capture the quality of analytic tasks (e.g., forecast error, classification accuracy, etc.).

2.1 Machine Learning

Machine learning is a computer science field that studies statistical techniques and algorithms that can *learn* from and make predictions on data by building models from sample inputs. Typical applications of machine learning include classification, regression, clustering, and dimensionality reduction. In this section, we review algorithms that we utilize for the purpose of realizing mobility analytics on aggregate locations (Chapter 5) as well as analyzing their privacy implications for individual users (Chapters 7 and 8).

2.1.1 Regression

Regression analysis consists of a set of statistical processes that estimate relationships among variables, and is widely used for forecasting. We review regression techniques that we deploy in Chapter 5 aiming to perform predictive mobility analytics on aggregate location time-series.

Auto Regressive Moving Average

To predict traffic volumes as well as detect mobility anomalies in Regions Of Interest (ROIs), we model their time-series with an Auto-Regressive Moving Average (ARMA) model. In particular, we build on the work by Box et al. [Box+15] who present an iterative method for choosing and estimating ARMA models.

Given a time-series Y_t, an ARMA model is a tool for understanding it, and predicting its future values. The model is usually denoted as ARMA(p, q), where AR(p) denotes the autoregressive model of order p and MA(q) refers to the moving average model of order q. More specifically, an ARMA(p, q) model is defined as:

$$Y_t = c + \sum_{i=1}^{p} \phi_i \cdot Y_{t-i} + e_t + \sum_{j=1}^{q} \theta_j \cdot e_{t-j} \tag{2.1}$$

where c is a constant, $\phi_1, \dots, \phi_p$, and $\theta_1, \dots, \theta_q$, are model parameters, and $e_t, e_{t-1}, \dots,$ are error terms which are assumed to be independent and identically distributed random variables sampled from a normal distribution with zero mean.

In general, an ARMA model expresses the estimated value of Y at time t, as the sum of p terms that compute the current value of Y as the weighted sum of most recent values of Y (the AR component) plus the weighted sum of q terms representing the average variation over q previous periods (the MA component).

Vector Auto Regression

To make enhanced predictions for the traffic volumes of ROIs in the presence of anomalies, we utilize a Vector Auto Regression (VAR) model trained on the aggregate time-series of multiple correlated ROIs. VARs are statistical models used in econometrics [AP03] to capture linear interdependencies among multiple time-series, and consist a generalization of uni-variate autoregressive models (AR models) that allow more than one evolving variable. All variables in a VAR model are treated symmetrically and each of them has an equation explaining its evolution based on its own lags as well as those of the other model variables. VAR modeling requires the prior knowledge of a list of variables which can be hypothesized to affect each other inter-temporally.

A VAR model describes the evolution of a set of h (endogenous) variables over a sample period $T = \{t_1, \dots, t_{|T|}\}$ as a linear function of their past values. The variables are collected in a vector y_t of size $(h, 1)$, whose $i -$ th element y_{it} is the observation of the variable i, at time t. A p-th order VAR model, denoted as VAR(p), is given by the equation:

$$y_t = c + A_1 \cdot y_{t-1} + A_2 \cdot y_{t-2} + \dots A_p \cdot y_{t-p} + e_t \tag{2.2}$$

where c is a vector of constants with size $(h, 1)$, A_i is a time-invariant matrix of size (h, h), and e_t is a vector of error terms with size $(h, 1)$ where: (a) $E(e_t) = 0$, every error term has mean zero, (b) $E(e_t e_t^{'}) = \Omega$, the co-variance matrix of error terms is Ω, and (c) $E(e_t e_{t-j}^{'}) = 0$, for any non-zero j there is no serial correlation in the individual error terms.

Forecast Error

To evaluate the quality of the predictions using a regression technique we employ the forecast error which captures the difference between the real and the predicted value of a time-series. More precisely, we utilize the absolute forecast error defined as:

$$e_t =\mid Y_t - \hat{Y}_t \mid \tag{2.3}$$

where Y_t is the actual time-series value at time slot t (ground truth) and $\hat{Y}_t$ is the predicted value for that time slot using the regression technique. For ease of presentation, we also convert the forecast error into a percentage error as:

$$c_t = \frac{e_t}{Y_t} \times 100 \tag{2.4}$$

2.1.2 Classification

Classification techniques aim at identifying to which set of categories (i.e., classes) a new observation belongs to, given as input a training set of observations whose category is known. It is considered as an instance of supervised machine learning, whereby a training set of correctly identified observations is available. We review classification algorithms that we employ in Chapters 7 and 8, to realize membership inference attacks on aggregate location data.

Logistic Regression

Logistic Regression (LR) [Cox58] is a linear model whose probabilities describing the possible outcomes of a single trial are modeled via a logistic (logit) function computed on a combination of the predictors. The parameters of the model are estimated with maximum likelihood estimation, using an iterative algorithm.

Nearest Neighbors

Nearest Neighbors (k-NN) [CH67] is an example of an *instance-based learning* algorithm as it does not build an internal model, rather it simply stores instances of the training data. Classification is performed with a simple majority vote of the nearest neighbors of each data point: a query point is assigned the data class which has the most representatives within the nearest neighbors of the point. The optimal configuration of the parameter k is data-dependent; generally larger values of k reduce the effect of noise on classification, but make the boundaries between classes less distinct.

Random Forest

Random Forest (RF) [Bre01] is an ensemble learning method which constructs a number of decision trees during training and outputs the majority class voted by the individual trees during testing. With RF, each tree in the ensemble is built from a sample drawn with replacement from the training set. When splitting a node during the construction of the tree, the split that is picked is the best possible among a random subset of the features. As a result of this randomness, the bias of the forest usually slightly increases but, due to averaging, its variance also decreases.

Multi Layer Perceptron

Multi Layer Perceptron (MLP) is a type of artificial neural network, consisting of at least three layers of nodes: the input, the hidden, and the output layers. Except for the input nodes, each other node is a neuron that uses a non-linear sigmoid activation function. MLP employs stochastic gradient descent along with a learning technique called back propagation [RHW85] to update its parameters. Finally, its multiple layers along with the non-linear activation function allow it to distinguish data that is not linearly separable.

Classification Metrics

The performance of a classifier is typically evaluated by counting the true/false positives/negatives (denoted as TP, FP, TN, and FN, respectively) of its predictions on a *test* set containing samples that were not used during its training. With these statistics a number of metrics that capture different aspects of its success towards the classification task can be calculated. Among these are Precision (or Positive Predicitve Value, PPV), True Positive Rate (TPR), and False Positive Rate (FPR), which are respectively defined as:

$$\mathrm{PPV} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{2.5}$$

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.6}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{2.7}$$

Moreover, a metric that combines the classifier's precision and recall is the F1 score which is calculated as:

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{PPV} \cdot \mathrm{TPR}}{\mathrm{PPV} + \mathrm{TPR}} \tag{2.8}$$

i.e, as the harmonic mean of precision (PPV) and recall (TPR).

Finally, one can combine the classifier's true positive rate (TPR) and false positive rate (FPR) values at various classification thresholds to derive the Receiver Operating Characteristic (ROC) curve, and compute the Area Under the Curve (AUC) score capturing the classifier's overall performance for the task under examination.

2.1.3 Dimensionality Reduction

Another machine learning application is dimensionality reduction which aims at reducing the number of considered variables for a task by retaining a set of principal variables. Typical dimensionality reduction approaches include feature *projection* or feature *selection*, and their techniques are widely used to make machine learning models simpler and more interpretable, reduce their training times, and enhance their generalization. Next, we review dimensionality reduction techniques that we employ in Chapters 7 and 8.

Principal Component Analysis

Principal Component Analysis (PCA) [Jol02] is a representative example of feature *projection* techniques that transform the data of a high-dimensional space to a space of fewer dimensions. More precisely, PCA converts observations of correlated variables to linearly uncorrelated ones (principal components) via an orthogonal transformation. The first principal component accounts for as much variability in the data as possible, and each succeeding component has the largest variance possible while being orthogonal to the previous components.

Recursive Feature Elimination

Recursive Feature Elimination (RFE) is an example of feature *selection* techniques which reduce the input feature space by selecting only a subset of features for the training of a model. In particular, with RFE a machine learning model is repeatedly trained and features with low weights are removed until the feature space reaches a specified size.

2.2 Differential Privacy

In Chapters 6, 7, and 8, we evaluate the protection that various defenses provide against inferences on aggregate location statistics. One of them is Differential Privacy (DP) [Dwo08], which is the established framework to define private functions on a dataset, that are free from inferences.

2.2.1 Definition

In a nutshell, differential privacy ensures that the addition or removal of a single dataset row does not (significantly) affect the outcome of an analysis on the dataset. More formally:

Definition 2.1. A randomized algorithm $\mathcal{M}$ is (ϵ, δ)-*differentially private* if for all datasets D_1 and D_2 that differ on at most one element, and all $\mathcal{S} \subseteq \text{Range}(\mathcal{M})$:

$$\Pr[\mathcal{M}(D_1) \in \mathcal{S}] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(D_2) \in \mathcal{S}] + \delta \tag{2.9}$$

where the probability is calculated over the coin tosses of $\mathcal{M}$ and $\text{Range}(\mathcal{M})$ denotes the set of possible outcomes of $\mathcal{M}$.

In other words, the outcome of a randomized algorithm $\mathcal{M}$ has a very small dependence on the members of the dataset. If $\delta = 0$, we say that the algorithm $\mathcal{M}$ is ϵ-differentially private.

A fundamental concept of differential privacy is the notion of *sensitivity*, which captures how much one record affects the output of a function and therefore the uncertainty in the response one must introduce to hide the participation of a single dataset row. More formally:

Definition 2.2. For any function $f : \mathcal{D} \rightarrow \mathbb{R}^d$, the sensitivity of f is:

$$\Delta f_p = \max_{D_1, D_2} \| f(D_1) - f(D_2) \|_p \tag{2.10}$$

for all datasets D_1, D_2 differing on at most one element, with $\| \cdot \|_p$ indicating the L$_p$ norm.

Finally, differential privacy enjoys the property of composition, which specifies the privacy guarantees in a sequence of computation. More specifically:

Theorem 2.3. Let $\mathcal{M}_1$ be an ϵ_1-differentially private mechanism and $\mathcal{M}_2$ an ϵ_2-differentially private one. Then, their sequential combination is $(\epsilon_1 + \epsilon_2)$-differentially private.

Other qualitative properties of differential privacy are protection against arbitrary risks, neutralization of linkage attacks, quantification of privacy loss, as well as closure under post-processing [DR+14].

2.2.2 Mechanisms

Next, we describe mechanisms that are widely used to achieve the guarantees of differential privacy.

Generic Mechanisms

We first review generic mechanisms that can be applied in various settings.

Laplacian Mechanism (LPM) [DR+14]: A commonly used mechanism to achieve differential privacy is to randomize the true output of a function using random noise independently drawn from the Laplace distribution with probability density function (PDF) $p(x|\sigma) = \frac{1}{2\sigma} \cdot e^{-\frac{|x-\mu|}{\sigma}}$, $\mu = 0$, and scale σ calibrated as follows:

Theorem 2.4. For any function $f : \mathcal{D} \rightarrow \mathbb{R}^d$, the mechanism $\mathcal{M}$ defined as $\mathcal{M}(\mathcal{D}) = f(\mathcal{D}) + \text{Lap}(\sigma)^d$, guarantees ϵ-DP if $\text{Lap}(\sigma)$ are independent and identically distributed Laplace random variables with $\sigma \geq \frac{\Delta f_1}{\epsilon}$.

Gaussian Mechanism (GSM) [DR+14]: Another mechanism widely used to achieve probabilistic differential privacy is to randomize the true output of a function using random noise independently drawn from the Gaussian distribution with probability density function (PDF) $p(x|\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$, $\mu = 0$, and scale σ configured as follows:

Theorem 2.5. For any function $f : \mathcal{D} \rightarrow \mathbb{R}^d$, the mechanism $\mathcal{M}$ defined as $\mathcal{M}(\mathcal{D}) = f(\mathcal{D}) + \mathcal{N}(\sigma^2)^d$, guarantees (ϵ, δ)-DP, if $\mathcal{N}(\sigma^2)$ are independent and identically distributed Gaussian random variables with $\sigma \geq \frac{\sqrt{2 \cdot ln(2/\delta)}}{\epsilon} \cdot \Delta f_2$.

Exponential Mechanism (EM) [DR+14]: The exponential mechanism is designed for answering queries with arbitrary utilities, while preserving differential privacy. In particular, this mechanism assigns to outputs with higher utility exponentially greater probabilities of being selected, such that the mechanism's output is closer to optimal.

Theorem 2.6. Given a utility function $u : (\mathcal{D} \times \mathcal{R}) \rightarrow \mathbb{R}$ for a database $\mathcal{D}$, the randomized algorithm $\mathcal{M}$ defined as:

$$\mathcal{M}(\mathcal{D}, u) = \left\{ \text{return } r \in \mathcal{R} \text{ with probability} \propto exp\left(\frac{\epsilon u(\mathcal{D}, r)}{2\Delta_u}\right) \right\} \tag{2.11}$$

satisfies ϵ-differential privacy, where $\Delta_u = \max\limits_{\forall r, \mathcal{D}_1, \mathcal{D}_2} |u(\mathcal{D}_1, r) - u(\mathcal{D}_2, r)|$.

Mechanisms for Time-Series

We now review mechanisms that have been specifically proposed for settings of time-series, which we investigate in this thesis.

Fourier Perturbation Algorithm (FPA) [RN10]: FPA performs the noise addition on the compressed frequency domain: a time-series is decomposed into frequencies using the Discrete Fourier Transform (DFT) and the first κ Fourier coefficients F_κ are kept. Then F_κ is perturbed with noise distributed according to $\text{Lap}(\sqrt{\kappa} \cdot \Delta f_2/\epsilon)$ and padded with zeros to the size of the original time-series. Finally, the inverse DFT is applied to obtain the perturbed time-series. As per [RN10], FPA provably guarantees ϵ-DP.

Enhanced Fourier Perturbation Algorithm with Gaussian Noise (EFPAG) [AC14]: EFPAG improves on FPA by probabilistically choosing the number of coefficients (κ) to be perturbed, and using the exponential mechanism (EM) to assign larger probability to values that minimize the root-sum-squared error between the input time-series and its noisy version. Moreover, rather than DFT, it uses the Discrete Cosine Transform (DCT) and employs Gaussian noise instead of Laplacian to achieve better accuracy. As a result, EFPAG guarantees (ϵ, δ)-DP.

Simple Counting Mechanism (SCM) [Dwo+10; CSS11]: Finally, a *weaker* version of the Laplacian mechanism has been proposed for settings of time-series. According to it, the counts of a time-series are perturbed with noise distributed according to $\text{Lap}(1/\epsilon)$ (assuming that the sensitivity of count queries is 1). However, unlike the previous mechanisms which provide *user-level* privacy this one provides *event-level* privacy, i.e., in the setting of aggregate location time-series, it can only protect single location visits.

2.3 Metrics

Finally, in this section, we review various metrics that we employ throughout this thesis.

2.3.1 Error Metrics

As discussed in Section 2.1.1, error metrics are commonly used to assess the quality of a forecasting regression model. In Chapters 6, 7, and 8, we also employ such metrics to capture the effect of a defense mechanism on the precision of the time-series data release. In particular, we utilize the Mean Relative Error (MRE) and the Mean Absolute Error (MAE), defined below.

Given two time-series Y and Y', of length n, denoting respectively an aggregate location time-series before and after a defense has been applied, we calculate:

$$\text{MAE}(Y, Y') = \frac{1}{n} \cdot \sum_{t=1}^{n} |Y'_t - Y_t| \tag{2.12}$$

$$\text{MRE}(Y, Y') = \frac{1}{n} \cdot \sum_{t=1}^{n} \frac{|Y'_t - Y_t|}{\max(\gamma, Y_t)} \tag{2.13}$$

where γ is a sanity bound mitigating the effect of very small counts.

2.3.2 Correlation Coefficients

Correlation coefficients are statistical measures that capture a relationship between two variables. In this thesis, we calculate such measures for various purposes. In Chapter 5, we employ them to discover correlated ROIs based on their aggregate traffic time-series aiming to improve forecasting tasks. Whereas, in Chapter 8, we measure the correlation between the original and perturbed location time-series to quantify how well they retain utility after a defense mechanism has been applied.

Pearson's r

Pearson's correlation coefficient, r, is a measure of linear correlation between two variables [Ben+09]. It varies between -1 and $+1$, with values closer to 1 indicating positive linear correlation, and to -1 total negative correlation (values close to 0 imply no linear correlation). Given two signals Y and Y', the Pearson correlation is calculated as:

$$r(Y, Y') = \frac{\sum(Y - \mu_Y) \cdot (Y' - \mu_{Y'})}{\sqrt{(Y - \mu_Y)^2 \cdot (Y' - \mu_{Y'})^2}} \tag{2.14}$$

where μ_X is the mean of a signal X.

Spearman's ρ

Spearman's correlation coefficient, ρ, is a non-parametric measure of the statistical dependence between the ranking of two variables [WM03]. It provides an estimate of how well the relationship between two variables can be described with a monotonic function and, unlike Pearson's r, it does not assume that both variables are normally distributed. Given two signals Y and Y', the Spearman's correlation coefficient, ρ, is defined as:

$$\rho = 1 - \frac{6 \cdot \sum d_i^2}{n \cdot (n^2 - 1)} \tag{2.15}$$

where $d_i = rg(Y_i) - rg(Y_i')$ is the difference between the two ranks of each observation and n is the number of observations. Similar to other correlation measures, Spearman's obtains values between -1 and $+1$, with 0 implying no correlation, and -1 or $+1$ implying an exact monotonic relationship. Intuitively, positive correlations imply that as Y increases, so does Y', while negative correlations mean that as Y increases, Y' decreases.

Kendall Tau

The Kendall Tau rank correlation coefficient is a measure of correspondence between two rankings, whereby values closer to 1 indicate strong agreement and those closer to -1 strong disagreement. More precisely, given two rankings Y and Y', the Kendall rank correlation

$\tau(Y, Y')$ is computed as:

$$\tau(Y, Y') = \frac{\mathrm{P} - \mathrm{Q}}{\sqrt{(\mathrm{P} + \mathrm{Q} + \mathrm{T}) \cdot (\mathrm{P} + \mathrm{Q} + \mathrm{U})}} \tag{2.16}$$

where P is the number of concordant pairs, Q that of discordant pairs, T the number of ties only in Y, and U the number of ties only in Y'. If a tie occurs for the same pair in both Y and Y', it is not added to either T or U [Ken45].

2.3.3 Statistical Distance Measures

Such measures capture the distance between statistical objects which can be random variables or probability distributions. In this thesis, we employ distance measures between probability distributions for various purposes. In Chapter 6, we use them to calculate the adversarial error in extracting users' mobility profiles from aggregate location statistics, while in Chapter 8, we employ them to evaluate how well the perturbed aggregate location time-series preserve the distribution of location visits, over time.

Kullback-Leibler divergence

Also known as discrimination information, the Kullback-Leibler (KL) divergence [KL51] captures the *distance* between two probability distributions. Specifically, for two discrete probability distributions V and W, the KL-divergence from V to W is defined as:

$$\mathrm{KL}(W||V) = \sum_i W(i) \cdot \log \frac{W(i)}{V(i)} \tag{2.17}$$

where W usually represents the *true* distribution of data and V an approximation of W. In other words, the KL-divergence from V to W measures the information lost when V is used to approximate W. We here note that KL is not a *metric*, as it does not satisfy the triangle equality and in general is not symmetric in W and V.

Jensen-Shannon divergence

Jensen-Shannon (JS) divergence [ES03] is used to calculate the similarity between two probability distributions. It is based on Kullback-Leibler (KL) divergence, but unlike the latter, it is symmetric and always obtains a finite value. More precisely, given two probability distributions V and W, the JS-divergence is a smoothed version of their KL-divergence, defined by:

$$\mathrm{JS}(W||V) = \frac{1}{2} \cdot \mathrm{KL}(W||Z) + \frac{1}{2} \cdot \mathrm{KL}(V||Z) \tag{2.18}$$

where $Z = \frac{1}{2} \cdot (W + V)$. When employing the base 2 logarithm for calculating the KL-divergence, the JS-divergence is bounded by 1, thus $0 \leq \mathrm{JS}(W||V) \leq 1$. Note that the square root of the JS-divergence is a *metric* denoted as Jensen-Shannon distance (also bounded by 1).

Chapter 3

Literature Review

The work of this thesis revolves around the realization of mobility analytics on crowdsourced aggregate location data as well as the evaluation of their implications on the privacy of individual users who contribute their location data to the aggregation process. In this chapter, we review work related to our line of research. In particular, we present the related literature on mobility analytics and their applications, as well as describe techniques that are commonly used for the privacy-preserving collection or release of aggregate location statistics. Moreover, we discuss about relevant work on membership inference attacks, as well as attacks and defenses known from the location privacy literature, which are connected to our research.

3.1 Mobility Analytics

Mobility analytics are widely used to improve urban planning, traffic congestion, and overall the quality of life in the context of modern smart cities. In this section, we review related work on the most prominent applications of mobility analytics, namely, mobility forecasting and anomaly detection.

3.1.1 Mobility Forecasting

Previous work on mobility forecasting, e.g., [Yav+05; ZN12; Fan+15], uses data mining techniques to predict the movements of *individual* users in a city. This can be useful for facilitating an individual's daily life via personalized services, e.g., by recommending optimal routes for her daily trips or by advertising promotion coupons. Unlike this line of research, in Chapter 5, we focus on forecasting *aggregate* crowd flows within Regions of Interest (ROIs) of a city.

To understand aggregate human mobility, relevant work employs *visualization* techniques. For instance, Sagl et al. [SLB12] study spatio-temporal patterns of human mobility using mobile network traffic data obtained from an Italian telecommunication provider. In particular, they analyze the characteristics of mobility patterns in four cities of Northern Italy and study them under *exceptional* events like concerts and soccer matches. Similarly, Senaratne et al. [Sen+18] extract aggregate spatio-temporal patterns of mobile users in Santiago (Chile) and assess their uncertainty. While visualization techniques help towards understanding human mobility, they are only a first step to forecasting it.

Specific to forecasting flows on *aggregate* statistics is the work of Garzó et al. [Gar+13] which uses distributed streaming algorithms to process large scale mobility data and make mobility predictions on large metropolitan areas or JamBayes [Hor+12], a probabilistic traffic forecasting system deployed in the Seattle Greater Area which uses Bayesian structure search on historical and contextual data to model bottlenecks. Furthermore, recent work [HZS16; ZZQ17] employs advanced statistical techniques, like Markov random fields or neural networks, to forecast the flows of city regions taking into account not only historical models but also other factors such as weather conditions and information about events.

Finally, another line of work focuses on understanding and predicting aggregate *commuting* mobility patterns. Zhong et al. [Zho+16] analyze mobility patterns in London, Singapore, and Beijing, using one-week worth of data containing user journeys using the underground. They show that aggregate mobility patterns exhibit regularity when considering time intervals longer than 15 minutes, and demonstrate that peak hours are those with the least variability during a day. Although their analysis results provide useful insights for our work, the authors do not present any methodology for predicting mobility. Then, Silva et al. [SKA15] introduce a general framework for predicting traffic volumes in the London underground: they build a predictive model for each pair of stations under normal conditions (called the natural regime) and then extend it to model disruptions like station or line closures. Their approach is substantially different from ours since the disruption information is actually part of a *ground truth* dataset obtained from London's transport authority. Overall, prior work on mobility analytics differs from ours as they do not consider the collection of data directly from users, nor the privacy implications thereof.

3.1.2 Detecting Mobility Anomalies

Anomaly detection is the process of identifying events or observations that do not conform to the expected patterns in a dataset [CBK09]. It is particularly useful in various domains ranging from intrusion detection [Laz+03] and fraud detection [Kou+04], to event identification in sensor networks [DFN05] and video streams [Sab+15].

Specifically focused on detecting traffic anomalies in the context of smart cities is the work by Barria et al. [BT11] which presents a detection algorithm tailored to road traffic using microscopic traffic variables such as relative speed of vehicles, inter-vehicle time gap, and lane changing. Similarly, Chawla et al. [CZH12] model the traffic between the regions of a city using dimensionality reduction and link-route optimization techniques aiming to detect the root cause of mobility anomalies.

Other types of work attempt to combine various data sources for the anomaly detection task. For instance, Thom et al. [Tho+12] present a system geared to detect spatio-temporal anomalies by performing clustering on geolocated messages originating from Twitter and visualize them using tag clouds. To demonstrate its usability, they experiment with three case-studies: an earthquake on the US East Coast, London riots, and hurricane Irene. Likewise, Pan et al. [Pan+13] combine mobility data along with that from social media to uncover the road network sub-graph associated with an anomaly, based on the routing behavior of drivers. Then, Zheng et al. [ZZY15] investigate whether collective detection of anomalies using information from multiple spatio-temporal datasets is possible. More precisely, they propose a probabilistic anomaly detection method based on a spatio-temporal likelihood ratio test and evaluate it on five datasets that capture various types of mobility patterns in New York City. Finally, Sun et al. [Sun+04] build Markov models on user mobility patterns within a cellular network, aiming to detect network intrusions. Note that although in Chapter 5, we also focus on identifying event mobility anomalies, unlike this line of work, we do so using aggregate crowdsourced location data – specifically, collected directly from users in a privacy-friendly way.

3.2 Privacy-Preserving Location Statistics

We now review techniques employed for privately collecting or releasing aggregate location statistics.

3.2.1 Privacy-Preserving Aggregation

The work discussed here focuses on the *collection* of aggregate location statistics while hiding the inputs of individual users contributing to the aggregation.

Prior work builds on cryptographic techniques like homomorphic encryption or secret sharing to collect aggregate statistics in a privacy-preserving manner [KDK11; Shi+11; Bil+14]. Specific to the collection of aggregate location statistics is PrivStats [Pop+11], a system that allows the computation of aggregate statistics over location data achieving privacy and accountability. In particular, the authors combine homomorphic encryption with zero-knowledge proofs to ensure accountability against malicious clients and support simple statistics like sum, average, and standard deviation. Prisense [Shi+10] allows privacy-preserving data aggregation in urban sensing environments employing data slicing and mixing techniques to support various statistics as sum, average, variance, and median of sensory data. Kopp et al. [KMM12] propose a framework that enables the collection of quantitative visits to sets of locations following a distributed approach that assumes the presence of an anonymizer between the clients and the server. Then, Hermann et al. [Her+14] use Identity Based Broadcast Encryption (IBBE) to enable users of a location based service to privately share their locations while allowing the service provider to gather aggregate statistics on the locations shared. Finally, Melis et al. [MDD16] demonstrate how to combine privacy-preserving aggregation with succinct data structures (i.e., Count-Min Sketches [CM05]) to efficiently compute location counts whilst provably protecting privacy of single data points. Their protocol offers scalability, independence from trusted third-parties or key distribution centers, and fault-tolerance, which are crucial properties in a mobile setting, thus, we employ it in Chapter 5 to build and evaluate crowdsourced mobility analytics on aggregate locations.

Another line of work focuses on applying perturbation to the individual user inputs, such that a *noisy* result that preserves user privacy is obtained at the aggregator side. Examples of this approach are the protocols introduced by Bassily and Smith [BS15] which produce succinct histograms of user inputs under local differential privacy or RAPPOR [EPK14] which enables the collection of aggregate browser statistics by relying on randomized responses [War65]. Specific to the aggregate location setting is SpotMe [Que+11] which employs randomized response to estimate the number of people in a geographic location or Chen et al.'s work [Che+16] which focuses on spatial data aggregation in the local setting. More precisely, they propose a framework that allows an untrusted server to learn the user distribution over a spatial domain relying on a personalized count estimation protocol and clustering.

Finally, other work achieves privacy-preserving aggregation by combining cryptographic techniques with the guarantees of differential privacy [Dwo08]. As an example, Rastogi and Nath propose PASTE [RN10] which combines encryption with distributed noise generation to run aggregate queries on private time-series inputs. Similarly, Shi et al. [Shi+11] show how an untrusted data aggregator can learn statistics over multiple participants' private data using cryptographic techniques along with a data randomization procedure aiming to achieve distributed differential privacy. Moreover, Brown et al. [BOT13] propose Haze, a system that enables privacy-preserving real time traffic statistics (e.g., predicting traffic flows or detecting road works) based on jury voting protocols and a customized differentially private mechanism.

While the above related work enables the privacy-friendly collection of location statistics, it differs substantially from the work performed in this thesis since: (a) it does not demonstrate how the collected aggregate location statistics can be used to perform mobility analytics (cf. Chapter 5), and (b) it does not consider or study potential privacy loss stemming from learning or releasing either *exact* or *pertrubed* aggregate location statistics (cf. Chapters 6, 7, and 8).

3.2.2 Private Location Data Publishing

This line of work aims at *publishing* aggregate location statistics with the guarantees of differential privacy [Dwo08]. For instance, Ho et al. [HR11] introduce an algorithm based on quadtree decomposition and clustering that allows the discovery of *interesting* geographic locations on aggregate location data. Similarly, Cormode et al. [Cor+12] develop differentially private spatial decomposition techniques to answer range queries over arbitrary geographic regions while Qardaji et al. [QYL13] propose a method for estimating the optimal spatial grid granularity to guarantee privacy.

Next, Machanavajjhala et al. [Mac+08] rely on synthetic data generation techniques to publish statistical information about commuting patterns such as origin destination commute distances. Accordingly, Mir et al. [Mir+13] present a differentially private approach to model human mobility based on Call Detail Records (CDRs). More precisely, they model various noisy distributions (e.g., hourly calls per location) and use them to estimate synthetic population densities.

Fan et al. propose FAST [FX12], an adaptive system for releasing real-time aggregate statistics with differential privacy. Their approach relies on sampling and filtering to improve the accuracy of data release at each time slot. In follow-up work, Fan et al. apply their framework for anomaly detection tasks on epidemic outbreaks [FX13] or they combine it with spatial decomposition techniques to enable traffic monitoring with differential privacy [FXS13]. Acs and Castellucia [AC14] present a differentially private algorithm tailored to the spatio-temporal density of Paris. More precisely, they propose a number of optimizations which are based on public characteristics of the dataset and combine sampling, clustering, and time-series compression, to release one week worth of aggregate location statistics. Furthermore, Wang et al. [Wan+16] combine the techniques of FAST [FX12] with dynamic grouping and adaptive budget allocation to release real-time spatio-temporal statistics satisfying a weaker privacy notion called *w-event* differential privacy [Kel+14]. Finally, To et al. [TNS16] attempt to release the entropy of geographic regions with differential privacy guarantees and demonstrate that the large amounts of required noise render the published results unusable. To this end, they demonstrate how to achieve better utility with a weaker privacy notion, namely, crowd blending privacy [Geh+12].

The work of this thesis is orthogonal to this line of research since our proposed frameworks can be used to evaluate the protection of such defenses against inferences on aggregate location statistics as well as tune the privacy/utility tradeoff that they achieve towards specific analytics tasks.

3.3 Membership Inference

In Chapters 7 and 8, we will investigate the feasibility of membership inference attacks on aggregate location time-series. Thus, in this section, we review prior work on this type of privacy attack as well as discuss potential defenses against it.

3.3.1 Membership Inference on Aggregate Statistics

Membership inference attacks on aggregate statistics aim at determining the presence of a target individual's data points within an aggregate version of a dataset. This type of privacy attack has been shown to be successful in various data domains ranging from genomic and biomedical data [Hom+08; Bac+16], to smart metering [Bue+17].

Homer et al. [Hom+08] show that aggregate genomic statistics leak information about the inclusion of a target's genome in a case study group (e.g., patients with a specific disease). In particular, their attack is based on the statistical distance between the allele frequencies of the aggregates and the victim's profile. Homer et al.'s attack was extended by Wang et al. [Wan+09] who demonstrated that membership inference is also feasible with reduced prior knowledge about the target, if one takes into account correlations within the human genome. Next, Backes et al. [Bac+16] show that membership inference can be mounted against individuals contributing their microRNA expressions to scientific studies, relying on a likelihood-ratio test. Furthermore, they propose and evaluate mitigation mechanisms based on differential privacy or hiding. Finally, Buescher et al. [Bue+17] study membership inference in the context of smart metering and show that aggregating a small number of household readings does not protect the privacy of individual (house) profiles. Overall, previous work differs from ours since it examines the feasibility of membership inference attacks on a different data domain than the one we investigate, namely, aggregate location time-series.

Moreover, most of the related work (except for [Bac+16]) does not evaluate defenses against the proposed attack, which we do in Chapters 7 and 8.

3.3.2 Membership Inference on Machine Learning Models

Another line of research investigates membership inference attacks on machine learning models, whereby the adversary's aim is to infer if a data sample was part of the model's training set. Shokri et al. [Sho+17] are the first to demonstrate the feasibility of membership attacks on machine learning models proposing the technique of shadow model training (i.e., training a model that mimics that target model's behavior) and using the intuition that a model ends up overfitting its training data. Then, Salem et al. [Sal+18] demonstrate the feasibility of the attack under less adversarial assumptions. Hayes et al. [Hay+17] show that membership inference attacks are also possible against generative models, while other work [HAPC17; Mel+18] does so for settings of collaborative and federated learning. Potential defenses against membership inference on machine learning models include coarsening the precision of the model's testing output (e.g., its prediction vectors) [Sho+17], techniques that prevent overfitting such as regularization, dropout, and model stacking [Sho+17; Sal+18], differentially private [Hay+17] or adversarial training [NSH18]. The work of this thesis focuses on a different problem than this line of research, i.e., we use machine learning techniques to realize and understand membership inference on aggregate location data, as well as to evaluate potential defenses against it (see Chapters 7 and 8).

3.4 Location Privacy

The focus of this thesis is the privacy risks stemming from the collection or publication of aggregate location statistics, thus, it builds on the existing location privacy literature, which we briefly review.

3.4.1 Privacy in Location Based Services (LBS)

This line of work operates in a different setting than the one considered in this thesis, i.e., it focuses on the protection of a users' location while interacting with a location based service provider, e.g., to obtain the nearest Points of Interest (POIs) such as restaurants, gas stations, or hospitals [JW08].

Traditionally, privacy protection in this setting is achieved by relying on notions originating from the areas of data mining and databases, e.g., *k-anonymity* [Swe02] or *l-diversity* [Mac+06]. To achieve these privacy properties, techniques like mix-zones [BS03], spatio-temporal cloaking [GG03; GL05; XKP09], and dummy queries [KYS05; SGI09; MRC09], are commonly employed. However, such techniques have a few limitations as either they require a trusted third-party during query processing, or the generation of plausible fake locations, which is a challenging task [CG09]. Furthermore, other research efforts have shown that *k-anonymity* underestimates potential location privacy risks [Sho+10; Sho+11a].

To diminish the need for a trusted *anonymizer*, prior work relies on cryptographic techniques to protect users' location privacy in the LBS setting. Some work, e.g., [Ghi+08; KS09; PBP10; Olu+10], employs Private Information Retrieval (PIR) protocols which allow users to obtain information from the LBS without revealing their locations. Whereas, other work, e.g., [Mar+05; WDR12], deploys secret sharing techniques such that a compromised server can only recover a user's location with degraded precision.

More recently, Andres et al. [And+13] present *Geo-Indistinguishability*, a variant of differential privacy adapted to the setting of LBS. The core idea is that a user reports to a service provider an obfuscated location which is indistinguishable from other locations within a radius of her real whereabouts. To this end, Andres et al. [And+13] propose a perturbation technique which adds random noise drawn from a planar Laplace distribution to the user's location and evaluate it against tasks like retrieving POIs from a LBS and sanitizing datasets that contain geographic information. Furthermore, *Geo-Indistinguishability* has been employed to design new mechanisms with semantic geographic privacy guarantees [CPS15] or has been combined with other privacy metrics (i.e., inference error) to derive optimal location privacy-preserving mechanisms [YLP17]. However, follow-up work has questioned its practicality. For instance, Primault et al. [Pri+14] show that *Geo-Indistinguishability* is insufficient to protect against the extraction of users' Points of Interest (POIs), while Oya et al. demonstrate that obfuscated locations might destroy the quality of service [OTPG17].

3.4.2 Privacy of Mobility Trajectories

A number of research efforts highlight how the inherent nature of mobility data can be harmful for users' privacy. For example, Golle and Partridge [GP09] demonstrate the feasibility of user re-identification leveraging the uniqueness of their home and work places. Similarly, Zang and Bolot [ZB11] examine the uniqueness of users' top visited locations and conclude that anonymization of mobility trajectories is an extremely difficult task. Then, De Montjoye et al. [DM+13] show that 4 spatio-temporal points are sufficient to identify 95% of individual users in a Call Detail Records (CDRs) dataset while coarsening their trajectories, both spatially and temporally, does not offer significant privacy protection. Finally, Rossi et al. [RM14] indicate that spatio-temporal trajectories emerging from check-ins in location-based social networks can be used to re-identify users while in [RWM15] they show how mobility features like speed, direction, and distance of travel, can link trajectories to specific users when GPS data is available.

The previous line of research unravels characteristics of mobility trajectories that can harm users' privacy, thus, inspired other work which aims at de-anonymizing users across datasets. For example, the work by De Mulder et al. [DM+08] or Gambs et al. [GKPC14] builds profiles of users' movements and subsequently matches them against anonymized versions of the datasets (mobile network data and GPS trajectories, respectively) proving that anonymous location data can violate users' privacy. Srivatsa et al. [SH12] de-anonymize mobility traces using social networks as a side-channel along with the intuition that encounters in mobility traces can be mapped to relationships on a social network graph. Then, Naini et al. [Nai+16] demonstrate that statistics about user behaviors can serve as *fingerprints*, while Cao et al. [Cao+16] propose a similarity metric to identify users across different mobility data sources. Finally, Wang et al. [Wan+18] de-anonymize mobility trajectories of cellular network users using as auxiliary information social network check-ins, and propose algorithms that account for the spatio-temporal mismatches between the datasets.

To mitigate inferences on mobility trajectories a few techniques have been proposed in the literature among which are spatio-temporal generalization [GF15; Gra+17], suppression [Hoh+07; TM08; Che+13], time distortion [HHC14; Pri+15], and the insertion of *dummy* locations in the traces [Kru09; Kat+12]. More principled approaches are those by Chatzikokolakis et al. [CPS14] who propose a predictive mechanism that extends *Geo-Indistinguishability* to protect the privacy of mobility traces, or by He et al. [He+15] who present a system that generates synthetic trajectories with differential privacy guarantees. Finally, recent work by Bindschaedler and Shokri [BS16] employs generative models to synthesize mobility trajectories that satisfy *plausible deniability*, while maintaining the location semantics of the original trajectories.

The research conducted in this thesis differs from this line of work as our aim is to investigate whether inferences about individual users are feasible on *aggregate* location data, a problem which has received little attention by the research community (see Section 3.4.4). Nonetheless, the insights obtained by previous work as well as the investigation of mobility characteristics that make inferences feasible are crucial for the realization of our research (see Chapters 6, 7, and 8).

3.4.3 Quantifying Location Privacy

Previous work on location privacy quantification has studied the privacy loss incurred from disclosing (possibly obfuscated) mobility traces of individual users. The main work in this area is the quantification framework by Shokri et al. [Sho+11a; Sho+11b] which considers a strategic adversary that has prior information about users' mobility patterns, knows the location privacy protection mechanism in use, and deploys inference attacks based on the prior and the observed obfuscated traces. As a privacy metric, they employ the adversary's *correctness* (or estimation error) towards her inference goal. Using their framework to evaluate defenses, the authors show that metrics like entropy or *k-anonymity* are not suitable for quantifying location privacy. While inspirational for our work, the techniques proposed by Shokri et al. [Sho+11a; Sho+11b] were conceived to evaluate privacy-preserving mechanisms applied to individuals' mobility traces, thus, they are not applicable in the aggregate location setting considered in this dissertation.

Other relevant work of this category is that by Manousakas et al. [Man+18] who propose graph kernel methods to quantify privacy loss stemming from mobility traces represented as mobility networks, or by Olteanu et al. [Olt+14; Olt+17] who quantify the effect of co-location information on the users' location privacy employing approximation algorithms executed on Bayesian network models. Once again, this line of work studies privacy leakage in different settings than that considered in this thesis.

3.4.4 Aggregate Location Privacy

Finally, most relevant to the work of this thesis is the independent research conducted by Xu et al. [Xu+17] which demonstrates that individual user trajectories can be extracted from aggregate location statistics. In particular, they present an attack which exploits the inherent characteristics of human mobility, namely, uniqueness and regularity. They model the mobility patterns within different times of day (e.g., night time or day time) as well as across different days of the week and combine these models to extract user trajectories using only public characteristics of the dataset as prior knowledge. While the results presented in [Xu+17] are in agreement with those of this thesis, i.e., both research efforts find that aggregate location statistics can harm the privacy of individual users, there are some notable dissimilarities between the two: (a) the adversarial goals considered in this work are different, i.e., rather than extracting user trajectories we focus on user profiling or localization (cf. Chapter 6) and membership inference attacks (cf. Chapter 7), (b) we contribute methodologies that allow quantification of privacy loss from the release or collection of aggregate location statistics, and (c) we utilize them to measure the protection provided by potential defenses against the inferences that we consider.

Chapter 4

Datasets

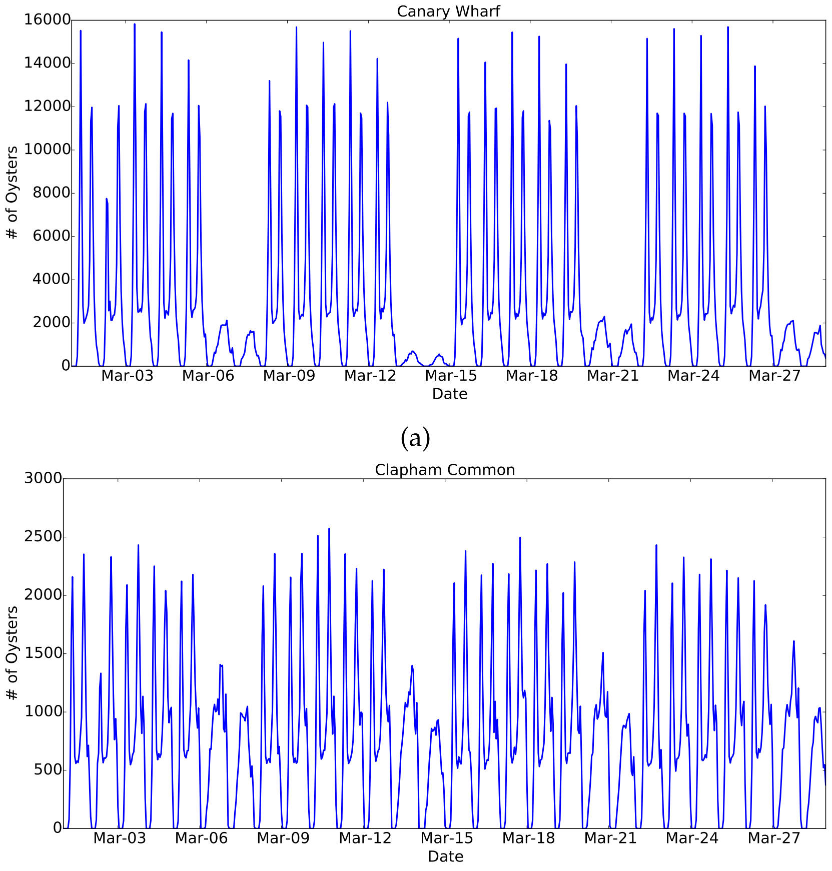
To perform mobility analytics on aggregate location data as well as evaluate their privacy implications, we employ two real-world mobility datasets, obtained respectively, from the Transport for London (TFL) authority and the San Francisco Cab (SFC) network. Both datasets contain a few weeks worth of location data and are often used in ubiquitous computing [CSC12; SKA15] and location privacy research [Sho+11a; Sho+11b; GKPC14; RWM15]. We select these datasets as they capture different mobility characteristics in modern cities: commuting patterns (TFL) are sparse and regular, i.e., they contain a few but repetitive data points for each traveler, while GPS trajectories (SFC) are dense and irregular, i.e., cabs generate lots of variable data points as they move around the San Francisco bay.

4.1 Notation

We first introduce notation that is used throughout this dissertation. We denote the set of mobile users as $U = \{u_1, u_2, \cdots, u_{|U|}\}$ and the set of Regions of Interest (ROIs) in which they move as $S = \{s_1, s_2, \cdots, s_{|S|}\}$. Moreover, we use $T = \{t_1, t_2, \cdots, t_{|T|}\}$, to represent the set of time intervals during which location data is available. We model the location time-series of a user $u \in U$ as a binary matrix L_u[1] of size $|S| \times |T|$, where an element $l_{s,t} \in L_u$ is 1 if user u is in location $s \in S$, at time $t \in T$, and 0 otherwise. Finally, we denote the *aggregate location time-series* as a matrix A of size $|S| \times |T|$, where each element $a_{s,t} \in A$ represents the number of users that are in location s, at time t.

Remark. In the beginning of each chapter, we will describe notation that is specific to it.

[1] We omit the subindex when there is no ambiguity.



(a)

(b)

FIGURE 4.1: TFL dataset – Hourly traffic volumes of two stations: (a) Canary Wharf and (b) Clapham Common.

4.2 Transport for London

London's transportation network consists of various connected sub-networks including the: London Underground (LUL), London Overground (LRC), Docklands Light Railway (DLR), National Rail (NR), Tramlink (TRAM), and London Transport Buses (LTB), all of which operate in the city under the umbrella of Transport for London (TFL). The most common payment method for using the TFL network is the Oyster card, a pre-paid RFID card. We have obtained from the TFL authority a dataset containing Oyster card trips on the transportation network from March 2010. Each entry in the dataset describes a unique trip and consists of the following fields: *oyster card id*, *start time*, *tap-in station id*, *end time*, and *tap-out station id*. Overall, the dataset contains approximately 60M trips performed by 4M oyster cards covering 582 train/tube stations which we consider as Regions of Interest (ROIs).

Dataset Pre-processing

First, we discard trips from TRAM and LTB due to their scarcity in the dataset. We also discard trips from March 29-31, 2010 to obtain exactly four weeks worth of data, i.e., from Monday, March 1st to Sunday, March 28th. Finally, we set the temporal resolution to one hour aiming to achieve regularity in commuting patterns as suggested in [Zho+16], thus, overall, the dataset contains 672 time slots (28 days × 24 hours).

Users' Location Time-Series

We consider each Oyster card as a user that moves within TFL's transportation network and we use its trips to populate the matrix L. More specifically, $l_{s,t} \in L$ is 1 if the user touched-in or out at station $s \in S$, during time $t \in T$, and 0 otherwise. When an Oyster card does not report any location at a particular time slot, we assign it to a special ROI denoted as *null*. As a result, the user's location time-series L is a matrix of size $|S| \times |T| = 583 \times 672$.

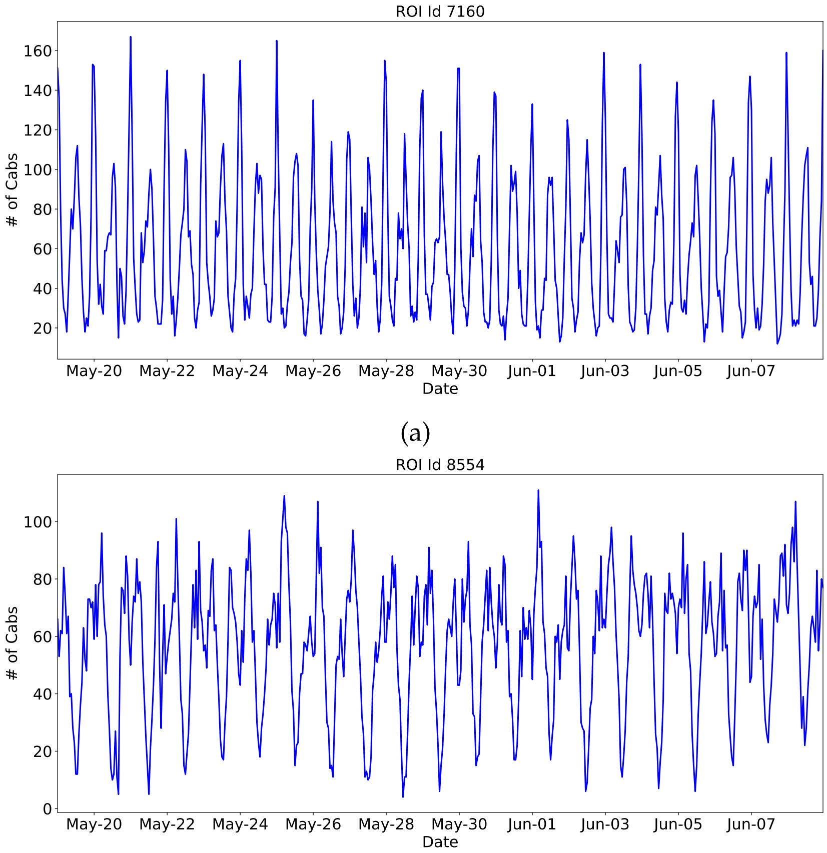
Aggregate Location Time-Series

We aggregate the users' location time-series to count the number of oyster cards that tap-in and out at each ROI and obtain hourly time-series, which capture the traffic evolution at TFL stations. In particular, each element $a_{s,t}$ of the aggregate matrix A, also of size $|S| \times |T| = 583 \times 672$, is calculated as $\sum_{u \in U} l_{s,t}$, where $l_{s,t}$ are the entries of the users' location time-series.

In Figure 4.1, we plot the *hourly* aggregate time-series of two TFL stations – Canary Wharf (one of the busiest stations of London) and Clapham Common (a residential station) – showing different patterns during weekdays and weekends, as well as peak commuting (morning and evening) hours. In general, we note that there is weekly and daily seasonality in the stations' time-series as well as stationarity (i.e., no particular trend). We verify the latter by performing the Augmented Dickey-Fuller test [DF79] which indicates that 90% of TFL tube stations have stationary time-series with 95% confidence.

Remark. For the privacy evaluations performed in Chapters 6, 7, and 8, we sample the TFL dataset and retain data for the 10K Oyster cards with the largest amount of trips. This yields a total of 7.3M events, with the top 10K oysters reporting on average 728 ± 16[2] tap-in and

[2] We here note that ± indicates one standard deviation from the mean value.



(a)

(b)

FIGURE 4.2: SFC dataset – Hourly traffic volumes of two ROIs with ids: (a) 7160 and (b) 8554.

out events, over 20 ± 9 unique ROIs. Overall, our sampled dataset is relatively sparse as the commuters are in the transportation system, on average, for 115 ± 21 out of the 672 hourly slots (28 days). Finally, we explicitly specify when we employ a temporal resolution different than one hour for some of our experiments (e.g., as we will do in Chapter 8).

4.3 San Francisco Cabs

The San Francisco Cab dataset [PSDG09] contains GPS mobility traces recorded by taxis in the San Francisco area from May 17 to June 10, 2008. Each record consists of: *cab identifier, latitude, longitude,* and a UNIX epoch *timestamp*. Overall, the dataset contains approximately 11M GPS coordinates, generated by 536 cabs.

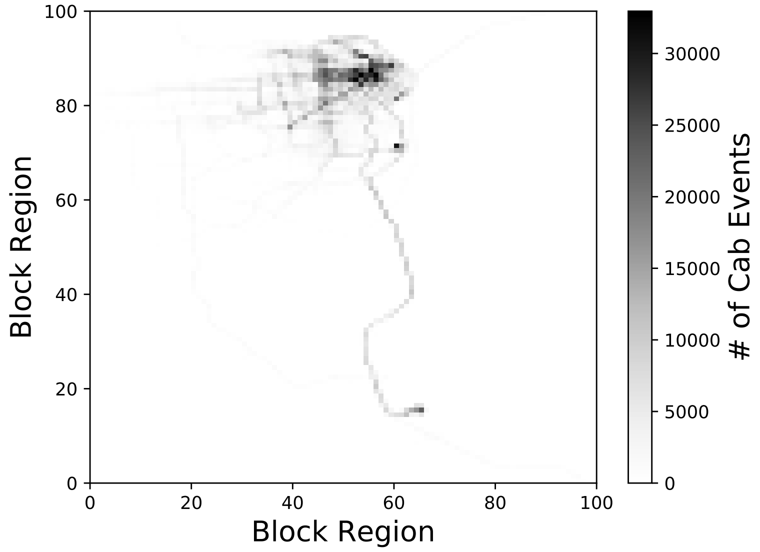


FIGURE 4.3: # of cab events on the 100×100 San Francisco grid (May 19 – June 8, 2008).

Dataset Pre-processing

Since locations in this dataset are GPS coordinates we divide the wider area of San Francisco in a grid consisting of Regions of Interest (ROIs). In particular, the grid consists of 100×100 ROIs, each covering an area of 0.02mi^2. Moreover, to facilitate our experiments, we keep exactly 3 weeks of data, i.e., from Monday, May 19 to Sunday, June 8, and we group traces in one hour epochs. Thus, this dataset contains cabs' location information for 504 hourly time slots (21 days × 24 hours). Finally, we remove duplicates, i.e., a cab reporting the same ROI multiple times during a time slot.

Users' Location Time-Series

For each cab, we populate its location time-series matrix L by setting $l_{s,t}$ to 1 if the cab reports its presence in the cell s of the spatial grid, at time t, and 0 otherwise. As for the TFL data, if a cab does not report any location at a specific time slot we assign it to a special ROI, which we denote as *null*. Thus, the L matrix is of size $|S| \times |T| = 10,001 \times 504$.

Aggregate Location Time-Series

To perform aggregation we count the number of taxis that report their presence in a ROI during a time slot and we create time-series that captures each ROI's traffic volume over time. More specifically, each item $a_{s,t}$ of matrix A is computed as $\sum_{u \in U} l_{s,t}$, where $l_{s,t}$ are the entries of each cab's location time-series. We here note that A is also a matrix of size $|S| \times |T| = 10,001 \times 504$.

In Figure 4.2, we plot the aggregate time-series of two representative ROIs in the San Francisco area, one of the busiest ones (id = 7160) and a moderately busy one (id = 8554). We observe weekly and daily patterns along with stationarity (96 out of the 100 busiest regions have stationary time-series with 99% confidence as indicated by the Augmented Dickey-Fuller test) while no particular trend is visible. We also aggregate the traffic from our 3-week dataset for each ROI and, in Figure 4.3, plot the resulting heatmap. Unsurprisingly, we find that the downtown area of San Francisco exhibits the highest traffic volumes, while the route to/from SFO airport is also clearly visible.

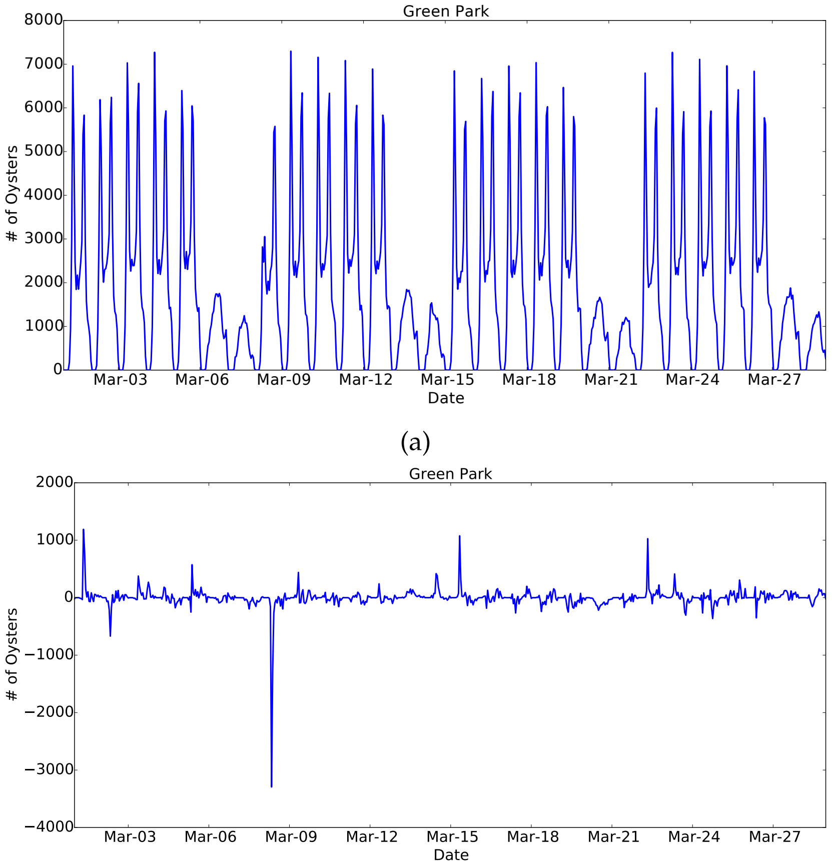
Remark. For the privacy evaluations performed in Chapters 6, 7, and 8, we focus *only* on the downtown of San Francisco. In particular we select an area of 30.3mi^2, which we split in a 10×10 grid. This leaves us with 2M events generated by 534 cabs, each reporting on average $3,826 \pm 1,069$ events over 78 ± 6 unique ROIs. Overall, this dataset is less sparse than the TFL one as taxis report ROIs for 340 ± 94 out of the 504 time slots (21 days). Finally, we explicitly specify when we use a different spatial or temporal resolution for some of our experiments (e.g., as we will do in Chapter 8).

Chapter 5

Privacy-Friendly Mobility Analytics on Aggregate Location Data

Location data enables a wide range of mobility analytics such as real-time traffic statistics [Gar+13; Waz18], forecasting events [HZS16], and detecting mobility anomalies [Pan+13], that can improve journey and transportation planning in the context of modern smart-cities. However, the large-scale collection of individual users' whereabouts raises important privacy concerns as these reveal sensitive information such as their home and work places, lifestyles, political or religious inclinations [Kru07; Ber15; Kul14], and can be used to re-identify them [GP09; ZB11; DM+13].

In this chapter, we investigate the feasibility of performing mobility analytics on crowd-sourced *aggregate* location data. In particular, we experiment with the real-world mobility datasets described in Chapter 4, and present a methodology based on time-series modeling that is geared to forecast traffic volumes in regions of interest (ROIs) and to detect mobility anomalies in them. In the presence of anomalies, we also make enhanced traffic volume predictions by training our model with additional information from correlated locations. Furthermore, we demonstrate how to build a privacy-respecting system for crowdsourcing the aggregate location data from users' mobile devices, such that an aggregator learns how many – but not which – users are in a location, at a given time. To this end, we present a mobile application prototype, called Mobility Data Donors (MDD), and present an empirical evaluation of its computation, communication, and energy complexities, which attest to the practicality of our vision.



(a)

(b)

FIGURE 5.1: TFL: Green Park station's time-series without (a) and with (b) deseasonalization.

5.1 Forecasting Traffic in Regions of Interest

We first investigate how to use aggregate location data for the purpose of forecasting traffic volumes in Regions of Interest (ROIs). Such predictions are particularly useful in modern cities for journey planning [Lam+05; SK03], congestion prevention [SKA15], as well as improving transportation service levels and adjusting staff needs at stations [Tra16]. We employ the aggregate location time-series computed on the TFL and SFC datasets (see Chapter 4) which we utilize to make hourly predictions in their ROIs (train or tube stations and grid cells, respectively).

5.1.1 Removing Seasonality

Our preliminary analysis of the TFL and SFC datasets shows that the aggregate time-series of their ROIs exhibit no particular trend but do preserve weekly and daily seasonality. Therefore, as proposed in previous work, e.g., [Hyl14], we de-seasonalize each region's aggregate time-series via additive decomposition. More specifically, we de-seasonalize the time-series of a ROI $s \in S$ as:

$$D_s = A_s - \bar{A}_s \tag{5.1}$$

where A_s is the original time-series of s and $\bar{A}_s$ is its seasonal part which is another matrix whose elements $\bar{a}_{s,t} \in \bar{A}_s$ are computed as:

$$\bar{a}_{s,t} = \frac{\sum_{k=0}^{T/c-1} a_{s,t+k \cdot c}}{T/c} \tag{5.2}$$

with c denoting the seasonality cycle, e.g., if $c = 24$ the daily seasonality of each region's hourly time-series is captured, whereas if $c = 168$ its weekly one. To this end, $\bar{A}_s$ is also a time-series containing the average count value of location s at each specific time slot (e.g., Mondays 3pm – 4pm, if weekly seasonality is considered). As an example, Figure 5.1 shows Green Park station's (a station among the busiest TFL ones) aggregate time-series in both its original and de-seasonalized form. Note a negative spike on the morning hours of March 8, as the station must have probably had reduced access (e.g., due to partial closure). In general, we observe that the de-seasonalized ROIs' time-series show strong auto-regressive structure.

5.1.2 Predicting Traffic Volumes in ROIs

To make our predictions, we focus on the 100 busiest TFL stations and the 100 most popular SFC grid cells. Since our preliminary analysis shows that ROIs' time-series are stationary and exhibit strong auto-regressive structure, we turn to ARMA modeling (see Section 2.1.1). For each ROI s, we feed the ARMA model with the values of the last 5 days of its aggregate and de-seasonalized time-series, A_s and D_s, respectively. We train the model using the first 4 days of D_s and test it against the last day (i.e., test day) of A_s following a recursive approach with a sliding time window to predict its hourly traffic. To do so, for each time slot we combine the ARMA model's predictions on D_s with ROI's seasonality $\bar{A}_s$, therefore, our predictions $\hat{A}_s$ are obtained by:

$$\hat{A}_s = \hat{D}_s + \bar{A}_s \tag{5.3}$$

where $\hat{D}_s$ is the ARMA model's prediction on the de-seasonalized time-series of ROI s. We compare our approach against a baseline, i.e., a *black-box* approach where we fit the ARMA model directly on each ROI's time-series A_s (i.e., without considering seasonal effects). Then, we evaluate the accuracy of the predictions using the absolute forecast error e_t as well as its percentage version c_t (see Section 2.1.1).

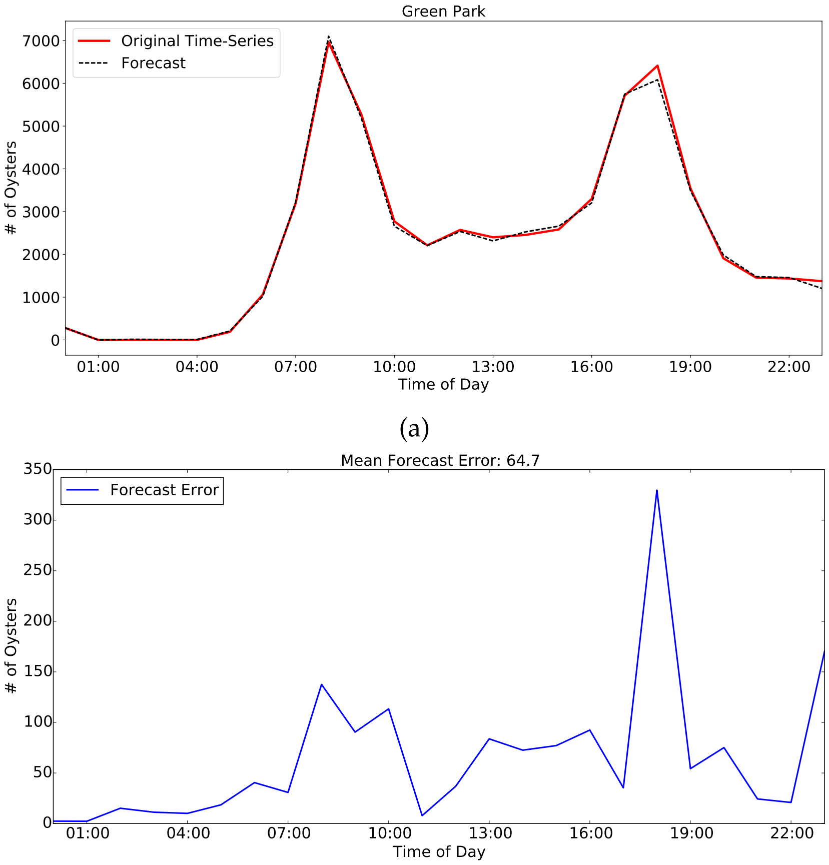
Figure 5.2a plots the traffic volume forecast for Green Park station on March 25, while Figure 5.2b shows the absolute forecast error. Overall, on the TFL dataset, the mean absolute forecast error for March 25, over the 100 busiest stations, is 59.53 ± 42.48 oysters, compared to 545.9 ± 376.8 oysters with the baseline approach. This corresponds to an error of $19.6\% \pm 59.5\%$ vs. $638\% \pm 1,619\%$, showing that the seasonality-based methodology significantly outperforms the baseline.

We follow the same approach for the SFC dataset, predicting the traffic volume of the most popular locations. As an example, Figure 5.3 shows the predictions and the forecast error for the region with identifier 8755, on June 5. The average forecast error over the 100 busiest regions is 5.62 ± 3.12 taxis ($19.7\% \pm 10.3\%$), whereas, the baseline error is 9.07 ± 2.95 ($35.4\% \pm 13.4\%$), once again showing that predictions can be improved when considering seasonal effects.

5.2 Detecting Traffic Anomalies

Next, we focus on detecting traffic volume anomalies on the ROIs' time-series. This is particularly important for traffic provisioning and travel planning as trip recommendations can be made to drivers or commuters during road accidents, incidents, or events that cause overcrowding in transportation stations [Tra18; Waz18; Pan+13].

Once again, we utilize ARMA modeling: our intuition is to train the model for each ROI and rely on the absolute forecast error as a confidence interval for detecting anomalies. More



(a)

(b)

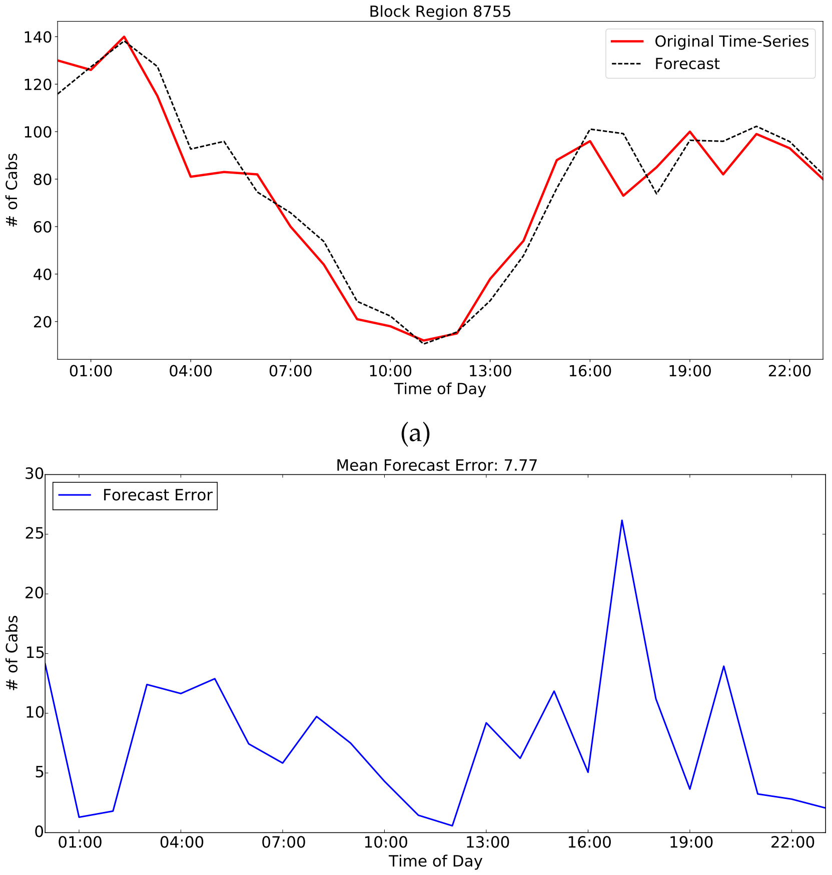
FIGURE 5.2: Hourly traffic forecasts for TFL's Green Park station on March 25: (a) predictions with the seasonal ARMA model and (b) forecast error.

precisely, assuming that the forecast error is normally distributed, we apply the 3σ rule and set a confidence interval λ. Thus, we detect an anomaly when:

$$e_t > \lambda \tag{5.4}$$

where $\lambda = \mu + 3\sigma$, with μ and σ being, respectively, the average and standard deviation of the forecast error e_t. In a way, we flag as anomalies those time slots that our model could not predict with good accuracy.

Subsequently, we experiment with our anomaly detection technique using a similar approach to that described in Section 5.1: we train the ARMA model using data of the first week of each ROI's time-series and test it against the remaining weeks (i.e., for the TFL dataset we



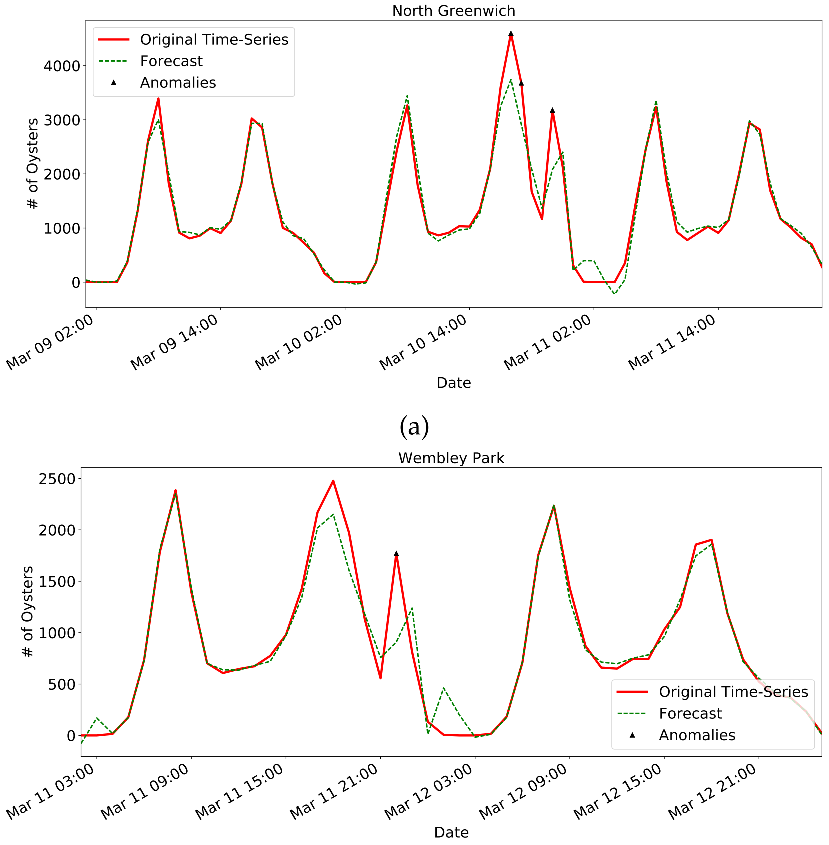
(a)

(b)

FIGURE 5.3: Hourly traffic forecasts for SFC's region with id 8755 on June 5: (a) predictions with the seasonal ARMA model and (b) forecast error.

have 3 test weeks while for the SFC we have 2), using a sliding window, and aiming to identify traffic volume anomalies. We focus on the 100 busiest stations of the TFL dataset and detect 896 anomalies, which corresponds to less than 2% of all one hour time slots in the 3 test weeks. On the SFC dataset, over the 100 busiest regions, we find 366 anomalies (i.e., 1% of the 2 test week time slots). We rank each anomaly based on its deviation from the confidence interval as a measure of its magnitude and keep track of the top 10% of anomalies, i.e., 90 anomalies for the TFL dataset and 30 for SFC: in Section 5.3, we will investigate whether or not we can enhance traffic predictions in the presence of anomalies by taking into account information from correlated ROIs.

Note that we do not have *ground truth* as to what constitutes an actual *anomaly* in our datasets, so we cannot empirically evaluate how well our approach corresponds to detecting, e.g., events, strikes, disruptions, etc. In general, we consider an anomaly to be a pattern that



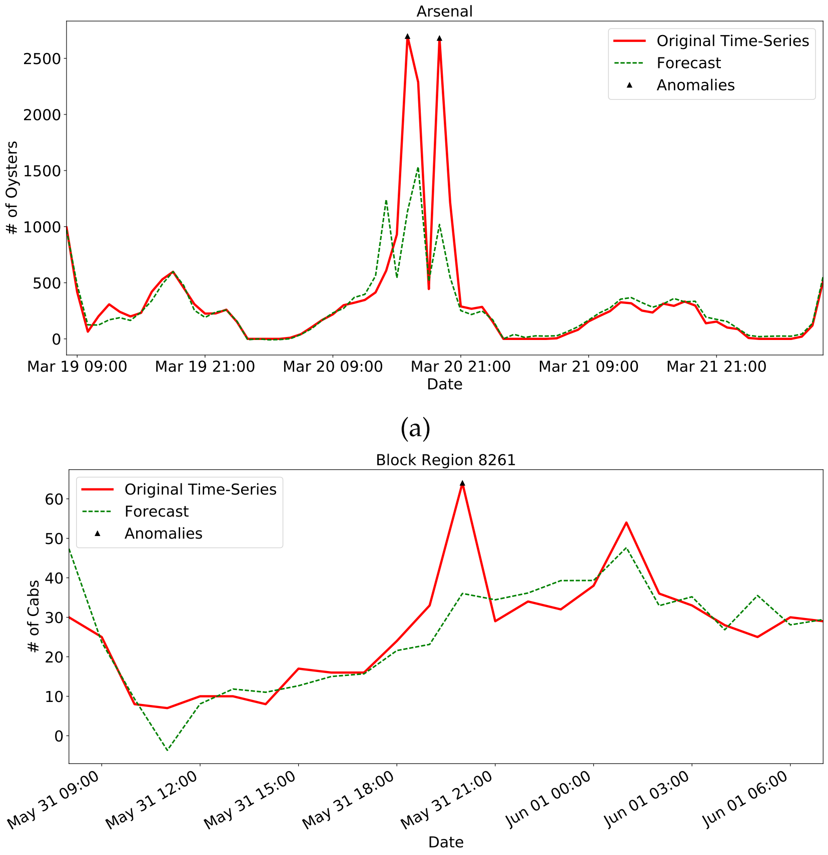
(a)

(b)

FIGURE 5.4: Aggregate time-series, forecasts, and detected anomalies, on TFL's (a) North Greenwich station, March 9–11, and (b) Wembley Park station, March 11–12.

does not conform to expected normal behavior [CBK09] and, as such, our anomaly detection techniques really aim at automatically flagging such patterns using aggregate locations. By focusing on the top events in terms of deviation from the confidence interval (i.e., unexpectedly increased or decreased traffic patterns in ROIs), we aim to investigate whether the collection of information from multiple ROIs can improve traffic volume predictions in the presence of anomalies (see Section 5.3).

Nonetheless, we discuss some case studies among the top anomalies that we were able to correlate with external events. As shown in previous work [SLB12], distinct human mobility patterns are observed during events that attract big crowds like football matches or music concerts. In Figures 5.4a and 5.4b, we plot anomalies that our method detects in North Greenwich and Wembley Park tube stations, on the evening hours of March 10 and March 11, respectively. These seem to correspond to concerts taking place in the O2 and Wembley



(a)

(b)

FIGURE 5.5: Aggregate time-series, forecasts, and detected anomalies, on (a) TFL's Arsenal station, March 19–22, and (b) SFC's region with id 8261, May 31–June 1.

arenas, which are venues close to these stations. Similarly, Figure 5.5a shows the aggregate time-series of Arsenal station as well as the anomalies detected on it when fitting our model. We can observe that the model detects anomalies on the evening of March 20, when an Arsenal FC soccer game was taking place. Finally, Figure 5.5b does the same for a region (id 8261) in the SFC dataset that is nearby AT&T Park, showing increased taxi traffic on the evening of May 31, when the San Francisco Giants were playing a baseball match.

5.3 Enhancing Predictions in Presence of Anomalies

We now investigate whether it is possible to improve ROI traffic volume forecasts in the presence of anomalies, using additional information from correlated ROIs. To this end, for each ROI, we use Spearman correlation (see Section 2.3.2) in order to discover those ROIs whose traffic can be useful for enhancing its predictions. Subsequently, we train a VAR model – geared to capture linear dependencies among multiple time-series (see Section 2.1.1) – with the time-series of a ROI as well as those of its correlated ROIs, and we compare the prediction results against a *local* model, i.e., an ARMA model trained only with ROI's past information (note that the ARMA model described in Sections 5.1 and 5.2 is now our *baseline*).

In the rest of this section, for each of our datasets, we start by describing our approach on a specific case study and then we generalize our results by focusing on the top events (90 for TFL and 30 for SFC) that our anomaly detection technique (Section 5.2) flagged as possible anomalies.

TFL. We focus on Saturday March 20, when our anomaly detection module spots anomalies, i.e., increased traffic volume, on the Arsenal station, likely caused by an Arsenal FC soccer game. We zoom in on the two hours before and after the game (15:00–17:00 and 19:00–21:00, respectively) when the majority of Arsenal fans, exit from, resp., enter Arsenal station. We follow a similar aggregation approach as that described in Chapter 4, although now, for each station, we keep two separate time-series: one counting passengers entering the station and one counting those exiting it. Once again, we de-seasonalize each station's entering and exiting time-series, as discussed in Section 5.1.1.

To discover stations correlated with Arsenal, we compute the Spearman correlation between the de-seasonalized time-series of passengers entering/exiting Arsenal as well as the de-seasonalized time-series of all the remaining stations in the TFL network, by sliding them up to 1 hour earlier/later. Our results show that the traffic exiting at Arsenal is highly correlated with the traffic entering at various other TFL stations including Arnos Grove, King's Cross, Leicester Square, Blackhorse Road, and Cockfosters (i.e., stations on the same line with Arsenal or on a line connected with that of Arsenal). We then set to improve the traffic volume predictions of passengers exiting at Arsenal station before the match, by feeding our model with the de-seasonalized entering time-series of the correlated stations. To do so, we use a vector auto regression model (VAR) which describes the evolution of a set of variables over the same sample period as a linear function of their past values.

Figure 5.6 shows the traffic volume predictions for passengers exiting Arsenal station

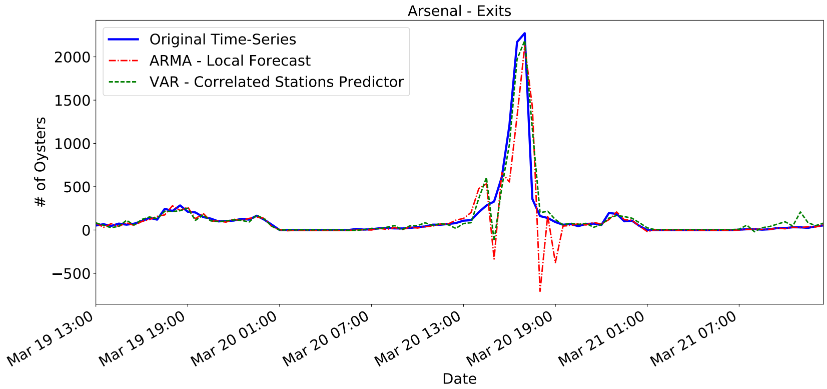


FIGURE 5.6: Forecast of passengers exiting TFL's Arsenal station on March 19–21 based on a local ARMA model vs. the VAR model trained with information from correlated stations.

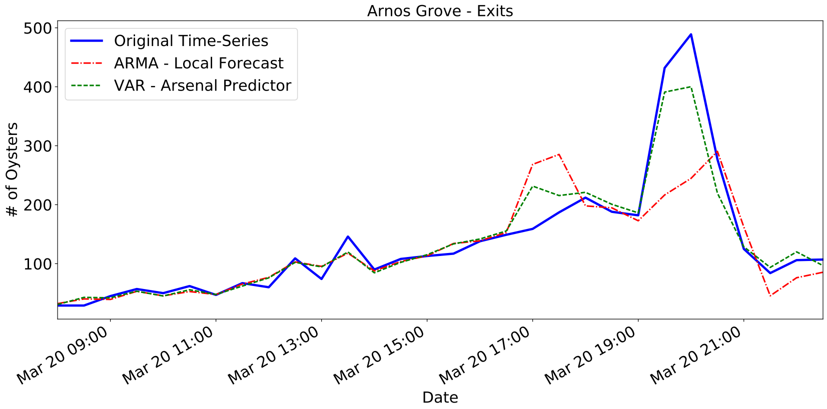


FIGURE 5.7: Forecast of passengers exiting TFL's Arnos Grove station on March 20 based on a local ARMA model vs. the VAR model trained with information from Arsenal station.

using the ARMA model trained with local information (i.e., the station's past exiting time-series) – which is now our *baseline* – and using the VAR model enhanced with additional information from the 10 most correlated stations. We observe that the *enhanced* model makes significantly better predictions between 15:00–17:00 on March 20, when there is increased traffic due to the game. We measure the average forecast error of the exit traffic predictions of Arsenal station on that day as 133.9 ± 270.6 oysters (i.e., 93% ± 185%) when making predictions with local information only and 65.26 ± 135.04 oysters (or 59.1% ± 57.4%) when enhancing it with that from correlated stations.

In relation to the same event, we discover high correlation between the time-series of passengers entering at Arsenal and those exiting at Arnos Grove a time slot later. Thus, we enhance the local ARMA prediction of traffic exiting volume at Arnos Grove by feeding the

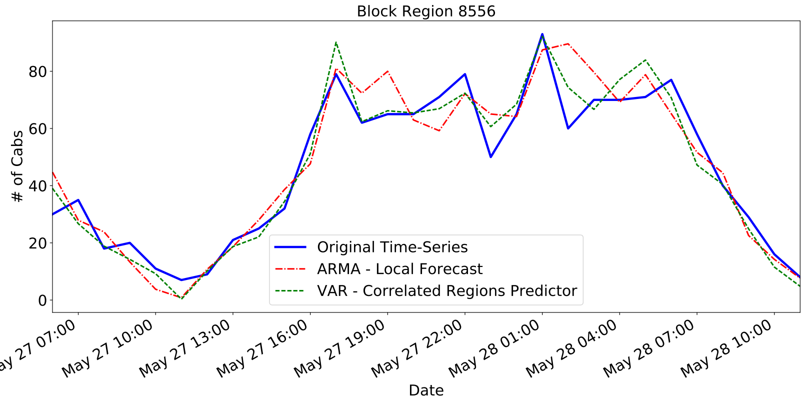


FIGURE 5.8: Taxi volume forecast on SFC region with id 8556 on May 27–28 based on a local ARMA model vs. the VAR model trained with information from correlated ROIs.

model with the de-seasonalized traffic entering Arsenal. The enhanced VAR model makes better predictions between 19:00 and 21:00 when there was increased exiting traffic after the Arsenal game was finished (see Fig. 5.7). Indeed, the average forecast error of Arnos Grove station for March 20 decreases from 23.38 ± 48.79 oysters to 11.22 ± 18.61, a 52% average improvement in traffic volume predictions.

Next, we apply our approach for all the *top 90* anomalies detected on the TFL dataset (Section 5.2), i.e., for each station under the presence of an anomaly, we predict its exiting or entering traffic (depending on which direction the anomaly has been detected) using a VAR model trained with the exiting or entering time-series (again, depending on the anomaly direction) from 10 correlated stations. We measure the average forecast error over the day of the anomaly and we compare it against a local approach (ARMA model), where predictions are made using only past station's information. Overall, for the 90 anomalies of the TFL dataset, we observe a 29% ± 13% improvement in our predictions when employing the VAR model, indicating that sharing information between correlated ROIs improves the quality of predictions.

SFC. We follow a similar approach for anomalies detected on the SFC dataset. Our correlation analysis shows that the de-seasonalized time-series of neighboring regions (i.e., grid cells) have high correlation, as it is likely that they are connected by the same roads. For instance, if we focus on an anomaly detected in region 8556 between May 27–28 (see Fig. 5.8), we observe a 40% improvement in predictions when training a VAR model including additional information from 5 correlated regions (i.e., block regions with ids 8557, 8657, 8558, 8655, and 8555) compared to the baseline, i.e., the ARMA model that predicts using only local information. Similar to our TFL experiments, we generalize this approach by trying to improve the predictions for the *top 30* anomalies detected on the SFC dataset, enhancing our model with information from 5 correlated regions. In this case, we obtain a 18% ± 14% average improvement on the predictions.

Discussion. We observe that the techniques described above yield better improvements on the TFL dataset compared to the SFC one. A possible explanation arises from the nature of the TFL dataset and the way passenger trips are aggregated, i.e., at station level while preserving the notion of *direction* (number of passengers exiting or entering a station). This allows us to perform a more fine-grained correlation analysis in comparison to the SFC dataset, where as we aggregate the GPS locations, we lose the granularity of each taxi's trajectories (i.e., a taxi moving from one region to another). While this is probably a good feature vis-à-vis privacy protection, it highlights that some analytics require the collection of (privacy-friendly) aggregate location statistics while preserving directions.

5.4 Crowdsourcing Privacy-Friendly Mobility Analytics

After having assessed the usefulness of aggregate location data for mobility analytics, we now investigate how to *enable* its collection in a privacy-friendly way. To this end, we design a distributed, collaborative framework whereby users install a mobile application – called *Mobility Data Donors* (MDD) – that regularly monitors their locations, stores it locally, and periodically reports it to a server in a privacy-friendly way. Privacy is respected through aggregation, by means of the scalable private aggregation protocol presented in [MDD16], thus, the server only learns aggregate information, i.e., how many (but not which) users are in a particular region or entered/exited a particular underground station in an interval of time. Once the aggregate location data (i.e., counts of users' presence in ROIs) has been received at the server, it can be used for mobility analytics applications similar to those described in Sections 5.1, 5.2, and 5.3.

As discussed in Chapter 1, protecting privacy of user locations is critical, since sensitive data about individuals, such as their home and work places [Kru07], political or religion inclinations [Ber15], can be inferred, and even a few locations are enough to re-identify users from anonymized traces [GP09; ZB11]. The ability to privately collect location reports enables applications that would otherwise be impossible due to privacy concerns. For instance, obtaining data from TFL typically requires several rounds of NDAs and the promise not to re-distribute the data: although TFL could publish aggregate statistics, it is unlikely they would do so in real-time (a crucial aspect for mobility analytics) and anyway this would only capture one aspect of urban mobility—i.e., underground, overground, and train station trips, but not, e.g., cars, taxis, or bikes. In general, collecting locations directly from mobile users, without requiring them to forego their privacy, paves the way for a number of novel and interesting analytics.

5.4.1 Data Collection

To support private data collection, we require a secure aggregation protocol that allows a server to only learn aggregate locations. We choose the protocol proposed by Melis et al. [MDD16], which we review in more detail in Appendix A, as it supports scalability, independence from trusted third-parties, and fault-tolerance, i.e., properties which are fundamental for the success of a distributed crowdsourcing system.

Our system model mirrors to that of Melis et al. [MDD16], i.e., it consists of a server, or aggregator, that facilitates networking and collects aggregate location counts from a set of mobile users running the MDD app. There is no other trusted authority. As in [MDD16], we assume the aggregator and the users to be *honest but curious*, i.e., they follow protocol specifications and do not misrepresent their inputs, but try to extract information from the other parties' inputs. When installed, the MDD app generates a private/public key pair and communicates its public part to the aggregator. After setup, the app runs in the background, regularly collecting GPS coordinates. At predefined time slots (by default, every hour), the privacy-preserving aggregation is triggered by the server, provided that there are at least χ users connected, which are randomly assigned to groups of m users. In the default setting, the app maps GPS coordinates to a grid of $p \times p$ cells of ψ square miles, and builds a $p \times p$

matrix corresponding to the grid, setting to 1 items corresponding to ROIs the user has visited in the specified time slot (and 0 otherwise).[1] The values of χ, m, and ψ are passed onto the user to inform them of the granularity of the data collection, and give them the option to withdraw (minimum acceptable values can be adjusted from the MDD's settings). Next, as per [MDD16], the app generates blinding factors (summing up to zero) based on the keys of the users in the same group, and encrypts each entry in the matrix. Finally, it sends the encrypted matrix to the aggregator who obliviously aggregates all (encrypted) matrices and decrypts the aggregate location counts which can be used for the analytic tasks.

Besides recording coordinates, and mapping them onto a grid, the app can also recognize points of interest, such as train or underground stations, which is particularly useful for mobility analytics on transport datasets such as the TFL one. In this case, the aggregation takes place on a vector where each item corresponds to a point of interest and is set to 1 if the user has visited it, during the specified time slot.

5.4.2 Experimental Evaluation

Next, aiming to assess the real-world deployability of our techniques, we empirically evaluate the performance of the MDD application, in terms of computation, communication, and energy overhead. Specifically, we evaluate the overhead imposed by the cryptographic operations needed for the privacy-respecting data collection. We use the TFL and SFC mobility datasets as guidelines for simulating the system. For our experiments, we use the prototype implementation, in Javascript/Node.js, of the secure aggregation protocol by Melis et al. [MDD16] without input compression (i.e., Count-Min Sketches), and adapt its client-side to run on Android using Apache Cordova[2]. The cryptographic operations are implemented using elliptic curve cryptography, specifically, the Ed25519 elliptic curve [Ber+12] (supporting 256-bit points and offering 128-bit security) from the Elliptic.js library[3]. Finally, for the cryptographic hash operations of the protocol, the implementation employs the SHA-256 algorithm.

[1] Note that the app is easily tunable so that, instead of binary values, the matrix encodes, e.g., duration of user's presence in each ROI, whether the user has entered or exited a cell, etc.

[2] https://cordova.apache.org/

[3] https://github.com/indutny/elliptic

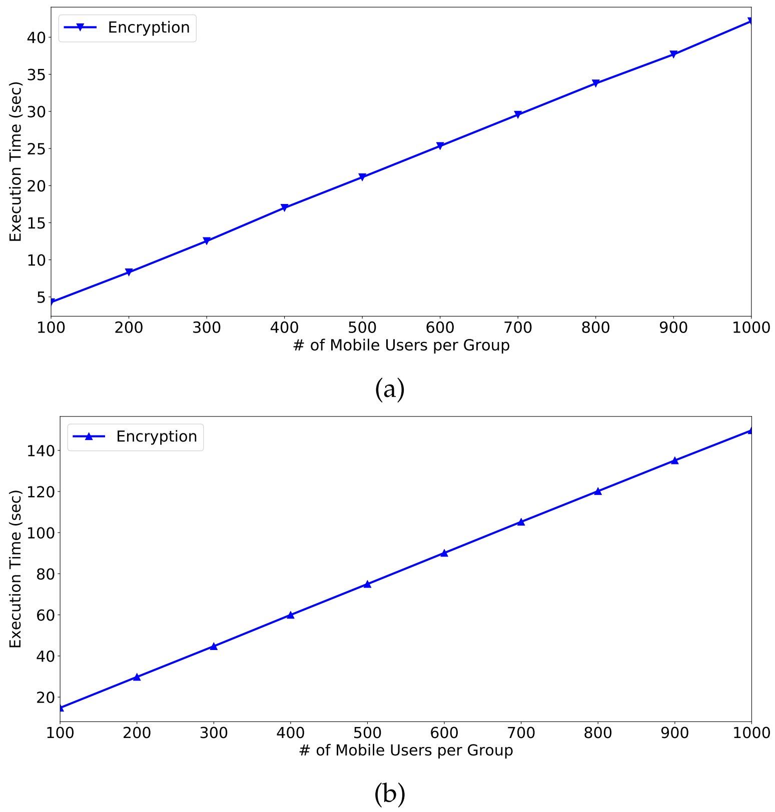


FIGURE 5.9: Execution times of the encryption phase for increasing number of users per group in the (a) TFL and (b) SFC settings.

For the sake of our evaluation, we run the experiments on a mid-range (rather than a high-end) Android device, as we do not want to limit deployment only to (possibly higher-income) users that have newer mobile phones. We use a Samsung Galaxy A3 device, equipped with a 1.2 GHz quad-core Snapdragon 410 processor and 1.5GB RAM, running Lollipop v. 5.0.2. For our energy consumption analysis, we utilize PowerTutor [Pow18], an Android app for power monitoring. Note that, although a Javascript implementation of the cryptographic operations might not be optimal in terms of efficiency, it offers portability among different mobile OSes.

TFL. We start our experiments with the TFL use-case. Recall that the TFL data involves 583 ROIs (stations), so each user device, for each time slot, encrypts a matrix of size $2 \cdot 583$, with the first row indicating entering the station and the second exiting it. We here remind that the complexity of the aggregation protocol depends on how many users are assigned to the same group, since the blinding factors are derived from public keys of other users in the group. In Figure 5.9a, we plot the execution time, measured on our Android device, of the encryption phase vis-à-vis the number of users in the aggregation group. As expected, running times grow linearly in the size of the group. For instance, the encryption performed by each mobile device takes 4.2s with 100 mobile users and 42s with 1,000. Therefore, keeping group sizes at a few hundreds of users, e.g., 200, could offer a reasonable trade-off between granularity and efficiency. Obviously, even if, say, 1 million London commuters were to participate, the system simply scales by running multiple parallel instances with each group, and combining multiple aggregates from 5,000 200-user groups.

Assuming groups of 200 mobile users, in each round of the protocol, each device has to download 10.7KB worth of public keys. Similarly, the transmission of the encrypted values for $2 \cdot 583$ entries in the location matrix results in an overhead of 4.54KB. Finally, we measure the energy consumption to be 862mJ for the encryption part, 609mJ to download public keys via Wi-Fi, and 322mJ to transmit the encrypted matrix. We observe that the energy overhead is quite small for a modern mobile phone (for reference we note that downloading a 20KB web page with Google Chrome via Wi-Fi consumes approximately 800mJ).

SFC. Next, we evaluate complexities considering the SFC use-case, for which we divide the city of San Francisco into a grid of 100×100 cells. In this setting, our mobile app, for each time slot, encrypts a matrix of size 10,001 (including the *null* ROI). Figure 5.9b displays the resulting computational overhead for an increasing number of users in the groups. Once again, we observe that execution times are linear to the number of users involved in the aggregation, i.e., the encryption phase requires about 14s with 100 and 149s with 1,000 users. We also measure communication and energy overhead, assuming groups of 100 users. Obtaining the public keys of users in the group requires downloading 5.37KB, while transferring the encrypted matrix (of 10,001 elements) requires 39KB, which, once again, is acceptable for a mobile app. The energy consumption of the cryptographic operations in each protocol round is 485mJ. Finally, the communication operations (via Wi-Fi) require 306mJ and $2,769$mJ, respectively.

Discussion. We here note that employing the secure aggregation protocol of Melis et al. [MDD16] *without* input compression (i.e., Count-Min Sketches) does not affect at all the utility of the analytics (i.e., the prediction tasks described in Sections 5.1, 5.2, and 5.3). Moreover, we observe that the number of users in each group mainly affects the computation cost, while the communication and energy overheads are primarily influenced by the size of the input. In particular, from our experiments above, we notice that the overhead is appreciably low when groups are in the order of hundreds of users and matrices are in the order of thousands. This means that if aggregation is performed over larger inputs, the protocol would quickly incur high energy, communication, and computation costs, and this would remarkably limit the deployability of the MDD app.

While for the use-cases considered in this chapter, the overhead of the secure aggregation protocol is appreciably low, it would not be able to scale to, e.g., privately building origin-destination matrices [SKA15; Zho+16] which are particularly useful to obtain finer grained statistics for discovering similar locations for personalized recommendations [Cle+11], or modeling the effects of disruptions in a transportation network [SKA15]. For such type of applications, as discussed in [MDD16], one could employ Count-Min Sketches to reduce complexities from linear to logarithmic in the size of the input while introducing minimal accuracy loss for the analytics tasks.

Chapter 6

Quantifying Privacy Leakage from Aggregate Location Data

In Chapter 5, we demonstrated how privacy-friendly crowdsourced aggregate location data can be useful for mobility analytics tasks, such as forecasting traffic volumes and detecting mobility anomalies, in the context of modern smart-cities. While aggregating location data is often considered a privacy-respecting strategy to enable such mobility analytics [Shi+10; Pop+11; Ube18; Waz18], there is no sound methodology which allows to reason about privacy loss for individual users that stems from the release or collection of aggregate location statistics.

In this chapter, we present a framework geared to address this gap and use it to perform a thorough evaluation of aggregate location privacy. More precisely, we consider an adversary which given some prior knowledge about the users and the aggregate location time-series aims at performing user *profiling*, i.e., inferring users' mobility patterns, as well as *localization*, i.e., recovering their punctual locations. We introduce various strategies to build the adversarial prior knowledge and discuss inference strategies which employ Bayesian reasoning or greedy approaches to improve it, given the aggregate location data. To this end, we define appropriate metrics to capture privacy leakage from aggregate location statistics. We apply our framework on the real-world mobility datasets described in Chapter 4 to quantify the privacy leakage originating from raw aggregate location time-series as well as to evaluate the privacy protection offered by output and input perturbation techniques based on differential privacy (DP), with respect to the utility loss towards the analytics.

Symbol	Description
Adv	Adversary
U	Set of mobile users
S	Set of locations (ROIs)
T	Time period considered
T_O	Observation period
T_I	Inference period
L_u	User u's location time-series
L_u^p	User u's mobility profile
A	Aggregate location time-series
A^p	Aggregate mobility profile
A'	Perturbed aggregate location time-series
P	Adv's prior knowledge
$\hat{P}$	Adv's inference output

TABLE 6.1: Notation.

6.1 Evaluating Aggregation Based Location Privacy

We first describe the basic components of our framework for quantifying privacy leakage from aggregate location data.

6.1.1 Framework Description

We employ the notation summarized in Table 6.1. As described in Chapter 4, we consider a set of users U that move among a set S of regions of interest (ROIs), e.g., landmarks, neighborhoods, or stations, at time instances in the set T. This set represents the time frame in which users' locations are collected (e.g., 1 week, 1 month, etc.), while they can be aggregated in epochs of different granularity (e.g., 15 mins, 1 hour, etc.).

Users' location time-series and mobility profiles. We denote the location time-series of a user $u \in U$, during T, with a matrix L of size $|S| \times |T|$, in which rows are ROIs and columns are epochs. L is a binary matrix such that $l_{s,t} \in L$ is 1 if the user was in location $s \in S$ during epoch $t \in T$, and 0 otherwise. We also define a user's *mobility profile*, L^p, where $l^p_{s,t} \in L^p$ represents the probability that a user is in location s, at time t, and is computed as $l_{s,t} / \sum_{s \in S} l_{s,t}$.

Aggregate location time-series and mobility profile. The aggregate location time-series is represented by the matrix A, of size $|S| \times |T|$. Each item $a_{s,t} \in A$ denotes the number of users in location s at epoch t, and is calculated as $a_{s,t} = \sum_{u \in U} l_{s,t}$, where $l_{s,t}$ are the entries of each user's L. The aggregation can be performed by a trusted aggregator or via a cryptographic protocol, as described in Chapter 5. We also define A^p, the *aggregate mobility profile*, as a probability distribution matrix whose entries $a^p_{s,t} \in A^p$ are computed as $a_{s,t} / \sum_{s \in S} a_{s,t}$. This represents the *likelihood* of a user being in a ROI, at an epoch, while observing the aggregates. For instance, $a^p_{s,t} = 0.1$ indicates that at time t, 10% of the user observations are in ROI s.

Adversarial prior knowledge. We model the prior knowledge the adversary, denoted as Adv, may have about a user $u \in U$, as P_u[1]. P is built over an *observation* period denoted as $T_O \subset T$ and is used by Adv to perform her adversarial task during the *inference* period $T_I \subset T$, when the aggregate location time-series are available. As a result, P is a matrix of size $|S| \times |T_I|$ and can be probabilistic (i.e., describing how likely a user is to visit a ROI) or binary (i.e., indicating whether a user will visit a ROI or not). In Section 6.1.2, we describe various approaches to build the adversarial prior knowledge.

Quantifying aggregate location privacy. Given the observation of the aggregate location time-series A during the inference period T_I, and the prior knowledge about each user P, Adv aims to infer information about individual users from A. We model the output of this inference as a matrix $\hat{P}$, which is of size $|S| \times |T_I|$ for each mobile user. We do so to quantify the privacy loss for individuals given the adversary's prior knowledge and her capability to exploit the aggregates. More specifically, we measure the adversary's *error* vis-à-vis the users' actual location time-series L during the inference period, after executing inference attacks, considering two goals: user profiling and user localization.

User profiling: Adv aims at inferring the mobility profile of a user $u \in U$. Given P and A (or A^p), Adv outputs a matrix $\hat{P}$. For each epoch of the inference period, $\hat{P}$ contains a probability distribution reflecting how likely the user is in each ROI. To compute Adv's error, we compare the user's mobility profile L^p to Adv's inference output $\hat{P}$ if we consider the result of the inference, or to the prior P if the aggregate location data is not available. For each $t \in T_I$, we use the Jensen-Shannon (JS) metric (see Section 2.3.3), to measure the distance between the probability distributions. In particular, for a user profile, we measure Adv's total error over the inference period T_I as:

[1] We omit the subindex when there is no ambiguity.

$$\text{AdvErr}_{\text{JS}} = \frac{\sum_{t' \in T_I} \text{JS}(L^p_{t'} || \hat{P}_{t'})}{|T_I|} \tag{6.1}$$

Intuitively, at each time slot, JS computes the distance between the inferred and the user's actual mobility profile. To this end, Equation 6.1 averages the distance over all time slots, i.e., it computes the adversary's mean error during the inference period.

User localization: Adv aims at inferring the punctual locations of a user over time. More formally, given P and A (or A^p), Adv outputs a binary matrix $\hat{P}$, with 1's for ROIs Adv predicts the user to be in, and 0's elsewhere. To measure Adv's performance we compare her predictive matrix (either prior P or posterior $\hat{P}$) against the user's actual location matrix L. Concretely, we use Adv's precision and recall when predicting the user's locations to derive the F1 score (reviewed in Section 2.1.2), and measure the total adversarial error as:

$$\text{AdvErr}_{\text{F1}} = 1 - \text{F1} \tag{6.2}$$

The F1 score captures Adv's performance in predicting the user's locations over the time slots of T_I, thus, Equation 6.2 reflects the adversarial error towards her localization goal over the inference period.

We here note that both adversarial goals have been considered in the location privacy literature [DM+08; Kru07; Sho+11a; Wer+14], although in different contexts, namely, reconstructing user traces, or recovering users' locations from obfuscated individual data.

Privacy Loss (PL). For both adversarial goals, we measure the privacy loss for an individual user from the aggregate location time-series as the normalized difference between Adv's error when using her prior knowledge (P), with and without A ($\text{AdvErr}_{P,A}$ and AdvErr_P, resp.). More specifically, for each user we define her privacy loss (PL) from the aggregate location time-series as:

$$\text{PL} = \begin{cases} \frac{|\text{AdvErr}_{P,A} - \text{AdvErr}_P|}{\text{AdvErr}_P} & \text{if } \text{AdvErr}_P \neq 0 \ \wedge \ \text{AdvErr}_{P,A} < \text{AdvErr}_P \\ 0 & \text{otherwise} \end{cases} \tag{6.3}$$

PL is a value between 0 and 1 and captures Adv's improvement towards her goal (i.e., either profiling or localizing the user).

6.1.2 Adversarial Prior Knowledge

We now present a few different approaches to build the adversary's prior knowledge, which we divide in *probabilistic* priors, i.e., user profiles averaging location reports over time, and *assignment* ones, i.e., binary matrices representing users' location visits at certain times. Essentially, they differ in how the P matrix is populated, depending on what information is available to the adversary and which strategy is employed to extract prior knowledge about each user.

In real life, adversarial prior knowledge may originate from, e.g., social networks, data leaks, location traces released by providers, or personal knowledge. Our aim is to describe a generic quantification framework, comparing different adversarial strategies, hence, we opt to construct priors from a subset of the users' location time-series (L), during the *observation* period T_O. We follow intuitive strategies, based on a sensible threat model in which Adv obtains information about users' routines and punctual locations (e.g., where one works and lives) over a certain period of time. Nonetheless, our framework is generic enough so that new ways of building Adv's priors can be easily incorporated.

Probabilistic Priors

Probabilistic priors model prior information that represent knowledge of user profiles.

ROI frequency. We start by considering that Adv knows the probability of a user visiting a given ROI during the observation period. We assume that Adv has access to a vector of size $|S|$, indicating how frequently the user visits each ROI during T_O. Adv then populates P by: (a) transforming the vector into a probability distribution using the total number of user's observations, M, as normalizing factor, and (b) copying the distribution onto P, for all time slots of the inference period T_I. More specifically, using the entries $l_{s,t}$ in the user's location time-series matrix L, we populate the elements $p_{s,t} \in P, \forall s \in S, \forall t' \in T_I$, as:

$$p_{s,t'}^{\text{FREQ\_ROI}} = \sum_{t \in T_O} l_{s,t} / M \tag{6.4}$$

ROI seasonality. This prior models the case that Adv knows the seasonal probability of a user visiting a ROI during the observation period T_O, for a given seasonal time period

SEAS. For instance, if SEAS corresponds to one day, and epochs are of one hour, we assume that Adv obtains a user's probability distribution over the ROIs for every hour in a day. If seasonality is computed on days of the week, the probability distribution over ROIs available to the adversary is for each hour, for each day of the week. More formally, if c denotes the seasonality cycle of SEAS (e.g., $c = 24$ for daily or $c = 7 \cdot 24$ for weekly seasonalities), then the seasonality profile is, $\forall s \in S, \forall i \in \{1, \ldots, c\}$:

$$\text{ROI\_SEAS}_{s,i} = \frac{\sum_{k=0}^{T_O/c-1} l_{s,i+k\cdot c}}{\sum_{s\in S}\sum_{k=0}^{T_O/c-1} l_{s,i+k\cdot c}} \tag{6.5}$$

Then, we calculate the elements $p_{s,t} \in P, \forall s \in S, \forall t' \in T_I$, as:

$$p_{s,t'}^{\text{ROI\_SEAS}} = \text{ROI\_SEAS}_{s,t' \bmod c} \tag{6.6}$$

Time seasonality. We assume that Adv knows the seasonal probability of a user reporting her location (without any information about which concrete ROIs) during the observation period T_O, for a given seasonal time period SEAS. For instance, if SEAS corresponds to one day, and the granularity is one hour, Adv learns which hours of a day a user is likely to report locations. More formally, if c denotes the seasonality cycle of SEAS, the time seasonality profile is, $\forall i \in \{1, \ldots, c\}$:

$$\text{TIME\_SEAS}_{i} = \frac{\sum_{s\in S}\sum_{k=0}^{T_O/c-1} l_{s,i+k\cdot c}}{M} \tag{6.7}$$

where M is the total number of user's observations within the period T_O. Then, the elements $p_{s,t} \in P$ are built, $\forall s \in S, \forall t' \in T_I$, as:

$$p_{s,t'}^{\text{TIME\_SEAS}} = \begin{cases} 1/|S| & \text{if } \text{TIME\_SEAS}_{t' \bmod c} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{6.8}$$

i.e., it is a *uniform* probability distribution over the ROIs $s \in S$ for the time slots when the user is likely to report locations.

Prior	Description
FREQ_ROI	Frequent ROIs, over time
ROI_DAY	Most frequent ROIs, for each time instance of a day
ROI_DAY_WEEK	Most frequent ROIs, for each time instance of a week
TIME_DAY	Most frequent time instances of a day, reporting ROIs
TIME_DAY_WEEK	Most frequent time instances of a week, reporting ROIs
LAST_WEEK	Last week's ROIs
LAST_DAY	Last day's ROIs
LAST_HOUR	Last hour's ROIs

TABLE 6.2: Different ways to build adversarial prior knowledge.

Assignment Priors

Next, we describe strategies to compute prior information that represents knowledge of users' punctual locations. An assignment prior is modeled as a binary matrix that predicts whether or not a user will be in a location $s \in S$, at time $t' \in T_I$.

Most popular prior ROIs. We model the case that Adv only considers users' favorite locations (POP). Given a probabilistic prior knowledge P', and a threshold λ' modeling what the adversary considers to be *favorite*, Adv builds a binary location matrix P so that its elements $p_{s,t} \in P$ are calculated $\forall s \in S, \forall t' \in T_I$, as:

$$p^{\text{POP}}_{s,t'} = \begin{cases} 1 & \text{if } p'_{s,t'} \geq \lambda' \\ 0 & \text{otherwise} \end{cases} \tag{6.9}$$

All prior ROIs. Next, we consider a scenario where Adv considers every location that a user visits, but not their frequency (ALL). Given a probabilistic prior knowledge P', Adv builds a binary location matrix P whose elements $p_{s,t} \in P$ are populated $\forall s \in S, \forall t' \in T_I$, as:

$$p^{\text{ALL}}_{s,t'} = \text{CEIL}(p'_{s,t'}) \tag{6.10}$$

where CEIL is the ceiling function, thus $p^{\text{ALL}}_{s,t'} = 1$ iff the probability of visiting s at t' is greater than 0 (i.e., the user has visited that location during time slots of T_O).

Last season. We assume that Adv has access to the last seasonal information for each user, i.e., the last season SEAS constitutes the observation period T_O. For instance, if SEAS corresponds to 1 day and the time granularity is 1 hour, Adv knows the locations visited in each hour of

Algorithm 6.1.1: BAYES

```
Input: $P$, $A^p$
for each $u \in U$ do
    for each $t' \in T_I$ do
        $\hat{P}(:,t') = P(:,t') \times A^p(:,t')$
        $\hat{P}(:,t') = \hat{P}(:,t') / \sum_{s \in S} \hat{p}_{s,t'}$
    return $\hat{P}$;
```

the last day. Formally, if c denotes the seasonality cycle, e.g., $c = 7 \cdot 24$ hours for weekly seasonality, the elements $p_{s,t} \in P$ are calculated utilizing a sliding window on the user's location time-series L, i.e., $\forall s \in S, \forall t' \in T_I$:

$$p_{s,t'}^{\text{LAST\_SEAS}} = l_{s,t'-c} \tag{6.11}$$

Summary. Table 6.2 summarizes our approaches to construct the adversarial prior knowledge. For priors taking seasonality into account, SEAS takes a value indicating the seasonal period we consider to build Adv's initial knowledge.

6.1.3 Inference Strategies

We now describe possible strategies that the adversary can follow to exploit aggregate locations in order to make inferences about individuals. We present algorithms that, taking as input Adv's prior knowledge about a given user (P) and the aggregate location time-series (A or A^p), output an updated matrix $\hat{P}$. This matrix represents Adv's posterior knowledge about the user's whereabouts over the inference period T_I by virtue of the availability of the aggregate location time-series. The proposed strategies can be used for both profiling and localization attacks, the difference being the nature of the output matrix $\hat{P}$, which is probabilistic in the former case and binary in the latter. In the following we use the symbol ":" to denote all the instances of a dimension in a matrix. For instance, $W(x,:)$ is a vector containing all the column values of row x, while $W(:,y)$ represents all the row values of column y, in matrix W.

Bayesian inference. The first strategy, summarized in Algorithm 6.1.1, computes the posterior probability distribution (i.e., mobility profile) of a user $u \in U$ over the locations $s \in S$, during each time slot of the inference period $t' \in T_I$, given the adversarial prior knowledge

```
Algorithm 6.1.2: MAX_ROI
   Input: $P, A$
1  $P_U(:,:,:) = \emptyset$
2  $\mathrm{LOC}_U(:,:,:) = \emptyset$
3  for each $u \in U$ do
4      $P_U = P_U || P_u$
5  for each $s \in S, t' \in T_I$ do
6      if $a_{s,t'} == 0$ then
7          $\mathrm{LOC}_U(:,s,t') = 0$
8      else
9          $U^* = \mathrm{SORT}(P_U(:,s,t'), a_{s,t'})$
10         for each $u \in U^*$ do
11             $\mathrm{LOC}_U(u,s,t') = 1$
12 for each $u \in U$ do
13     $\hat{P} = \mathrm{LOC}_U(u,:,:)$
14     return $\hat{P}$;
```

(P) and the aggregate mobility profile A^p which is computed on the observed location aggregates. More precisely, for each $t' \in T_I$, we apply the Bayes' rule which states that the posterior distribution of a user over the locations is proportional to the prior times the likelihood, as:

$$\hat{P}(:,t') \propto P(:,t') \times A^p(:,t') \tag{6.12}$$

This results to the user's un-normalized distribution over the ROIs at time t' and can be normalized by employing the law of total probability, i.e., by dividing with the factor $\sum_{s \in S} \hat{p}_{s,t'}$ (line 4, Algorithm 6.1.1).

Max-ROI. The Bayesian approach is well-principled but it considers the users independently, thus, it loses information related to the fact that at most $a_{s,t'}$ users can be assigned to location s, at time t'. We now describe a *greedy* alternative that accounts for this constraint. The algorithm aims at maximizing the total probability for each ROI by assigning the most probable users to each location. It is summarized in Algorithm 6.1.2, where $\mathrm{SORT}(V,x)$ denotes a function that returns the indexes of the *top* x values of a vector V.

Specifically, Adv first concatenates the probabilistic prior P matrices of all users $u \in U$ and creates a 3-dimensional matrix of size $|U| \times |S| \times |T|$, which we denote as P_U (lines 3–4, Algorithm 6.1.2). Additionally, she creates a localization matrix of the same size, denoted as LOC_U. Next, Adv selects all s, t' such that $a_{s,t'} = 0$ and sets the corresponding indexes of LOC_U to zero, i.e., she discards locations where no users were observed during the aggregation period (lines 6–7). Then, for all non-zero entries in A, she selects the $a_{s,t'}$ most probable

Algorithm 6.1.3: MAX_USER

```
   Input: $P, A$
1  $\mathrm{LOC}_U(:,:,:) = \emptyset$
2  for each $t' \in T_I$ do
3      for each $u \in U$ do
4          $S^* = \mathrm{INDEX}(P(:,t') > 0)$
5          for each $s \in S^*$ do
6              if $\sum LOC_U(:,s,t') < a_{s,t'}$ then
7                  $\mathrm{LOC}_U(u,s,t') = 1$
8          if $\sum LOC_U(:,:,t') == \sum A(:,t')$ then
9              break;
10 for each $u \in U$ do
11     $\hat{P} = \mathrm{LOC}_U(u,:,:)$
12     return $\hat{P}$;
```

users according to her prior, P_U, setting the corresponding indexes in LOC_U to 1 (lines 9–11). If there are users with equal probability, Adv can use any criterion, e.g., the total number of location reports (as we do in our experiments that are presented in Section 6.2) to make a decision. Finally, Adv outputs the location assignment profile of each user as her $\hat{P}$ matrix (lines 12–14).

Max-User. Our final inference attack is similar in spirit to the previous *greedy* strategy but, rather than maximizing the probability over the ROIs, it maximizes each user's probability over the ROIs by assigning them to their most likely locations. The algorithm is summarized in Algorithm 6.1.3, where $\mathrm{INDEX}(V > x)$ denotes a function that returns the indexes of a vector V whose values are larger than x.

More precisely, Adv first sorts users by some criterion, e.g., the total number of locations that they report (as we will do in our experiments in Section 6.2). Then, at each time slot $t' \in T_I$, Adv iterates over the users and assigns each of them to their most likely ROIs, provided that each ROI's aggregate count $a_{s,t'}$ is still not consumed (lines 3–7, Algorithm 6.1.3). The procedure is repeated until the assignments cover all the revealed aggregate information for the time slot (lines 8–9).

Remark. Our strategies are suitable for both adversarial inference goals, i.e., profiling and localization. For instance, if Adv is given a probabilistic prior, she can follow MAX_ROI or MAX_USER strategies and transform their assignment outputs to probability distributions that can be used for her profiling goal. Similarly, she can run BAYES on the prior knowledge and evaluate POP and ALL on its output to localize users.

6.2 Evaluation of Raw Aggregate Location Time-Series

We now use our framework to experimentally evaluate aggregate location privacy from the release of raw aggregate data. We compare different approaches to build priors (Section 6.1.2) as well as strategies to perform inference attacks (Section 6.1.3), using the TFL and SFC mobility datasets described in Chapter 4.

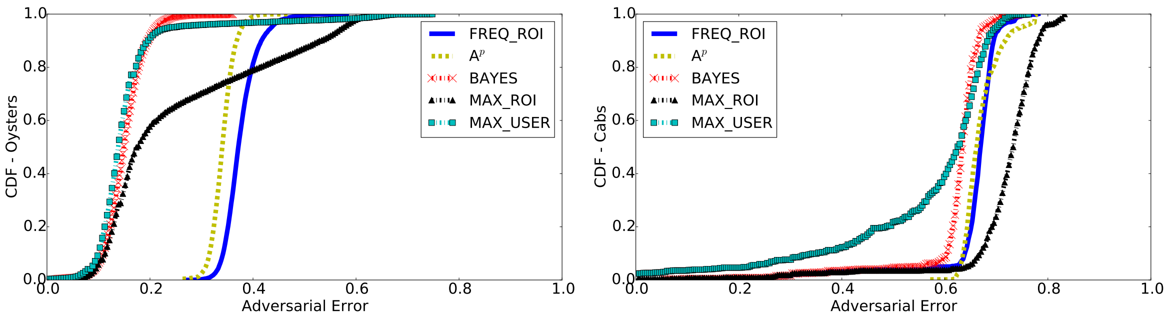
6.2.1 Training and Testing Data

To perform the inferences presented in our framework, we split the TFL and SFC datasets into *observation* and *inference* periods. More specifically, for TFL, we build the probabilistic prior knowledge using the first *3 weeks* of L, i.e., 75% of the data is used for training. As a result, the observation period T_O consists of $21 \times 24 = 504$ hourly time slots. Similarly, for SFC, we use the first *2 weeks* of data to derive the adversarial knowledge (i.e., 67% of the data is used for training), thus, the observation period consists of $14 \times 24 = 336$ time slots. In both cases, for the seasonal assignment priors we utilize a sliding window on L, as described in Section 6.1.2.

We evaluate Adv's performance in profiling and localizing users against the last week of data, i.e., 25% (33%) of data is used for testing in the TFL (SFC) dataset. As a result, the inference period T_I, during which the aggregate location time-series are available, consists of $7 \times 24 = 168$ hourly time slots. For TFL, in T_I, each station is reported $3,117 \pm 57,831$ times while stations have commuters touching in or out for 71 ± 54 out of the 168 hourly time slots. Similarly, for SFC, during T_I, each ROI is reported $6,920 \pm 7,860$ times while ROIs have taxis in them for 135 ± 62 out of the 168 time slots.

6.2.2 User Profiling

We start our experimental evaluation by quantifying aggregate location privacy against *user profiling* (Section 6.1). We study the impact of the information used to build Adv's prior vis-à-vis the strategy employed to exploit the aggregate location data. Specifically, we measure Adv's performance using the JS distance from the users' actual mobility profiles (Eq. 6.1), and use this metric in our plots (Figures 6.1–6.3). During our analysis, we also discuss the privacy



(a) TFL

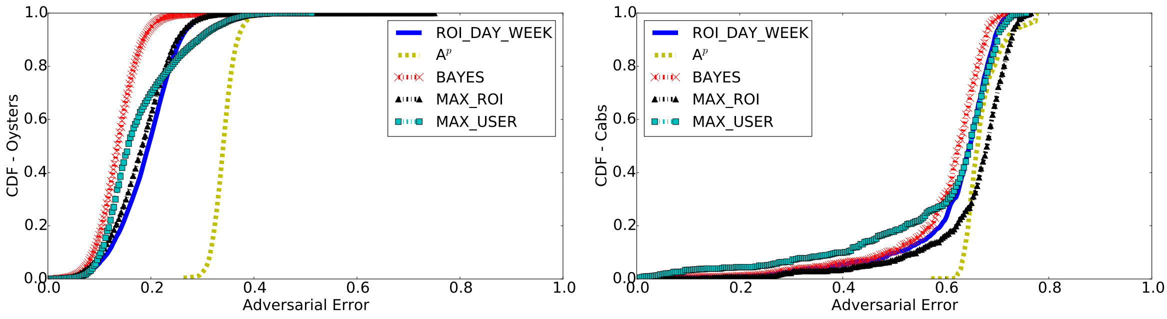
(b) SFC

FIGURE 6.1: FREQ_ROI prior – Adv's profiling error.

loss (PL, Eq. 6.3), allowing us to better understand the effect of aggregate data publication on privacy, independently of the prior mobility pattern of the user, as PL reflects how much the adversary has learned with respect to her initial knowledge.

Probabilistic Priors

FREQ_ROI. In Figure 6.1, we plot the CDF of the adversarial error over the user population (i.e., oysters or cabs), over the testing week with the FREQ_ROI prior, i.e., each user's frequent ROIs over time, for both datasets, and using different inference strategies. On the TFL dataset (Fig. 6.1a), a baseline attack where Adv uses only her prior knowledge (blue line) has an average error of 0.37, while with A^p (i.e., the population profile extracted from the aggregates, indicated by the yellow dotted line in the plot) the adversarial error is 0.34. When Adv uses both her prior and the aggregates for the inference, the error is notably reduced, yielding average errors amounting to $0.15, 0.25$, and 0.15, respectively, with the BAYES, MAX_ROI, and MAX_USER strategies. More specifically, this corresponds to an average privacy loss (i.e., adversarial improvement towards the profiling goal given the aggregates – indicated by the area between the blue line and the red, black or cyan line) of, resp., 0.6, 0.41, and 0.59, for individuals whose locations are used to compute the aggregate time-series. We also observe that inferences affect users in different ways, i.e., with BAYES, the adversarial error is reduced for all users, while with MAX_ROI and MAX_USER for 77% and 95% of users, respectively. This confirms that MAX_ROI and MAX_USER are somewhat greedy strategies and may end up selecting users that either report few (MAX_ROI) or many (MAX_USER) ROIs overall, to consume the aggregates.



(a) TFL

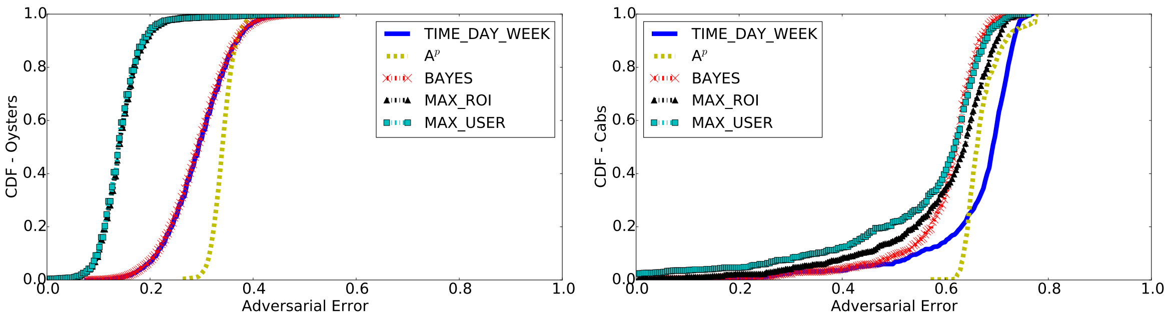
(b) SFC

FIGURE 6.2: ROI_DAY_WEEK prior – Adv's profiling error.

On the SFC data (cf. Fig. 6.1b), the adversarial error only relying on cabs' frequent ROIs prior (FREQ_ROI) is higher compared to that of TFL – 0.65 on average, and in this case it is quite similar to that owing to the aggregates (A^p). It drops to 0.62 with the Bayesian updating (corresponding to 0.06 privacy loss) and to 0.56 with MAX_USER (0.16 PL), indicating that taxis reporting the most locations are regular within them and end up losing more privacy. We also observe that, unlike in the TFL experiments, the greedy MAX_ROI strategy actually deteriorates Adv's mean error (0.71), owing to the bias introduced by taxis visiting few ROIs (i.e., cabs having high probability to appear in a ROI). Overall, we find that profiling commuters based on their frequent ROIs is more effective than profiling cabs, as cabs report more locations and follow variable routes during their shifts.

ROI_DAY_WEEK. Next, we report Adv's error with the ROI_DAY_WEEK prior, i.e., a weekly profile that takes into account location frequency as well as time and day semantics (e.g., users' locations on Mondays, 3pm). The results are plotted in Figure 6.2, for both datasets. Note that we also experiment with location frequency and time only (and not day) semantics to build the prior (ROI_DAY), which yields larger errors, as less information is considered. To ease presentation, we defer details to Appendix B.1.

With the TFL data (Fig. 6.2a), it is clear that commuters' most frequent ROIs for the time instances of a week (ROI_DAY_WEEK) are a more informative prior than their frequent ROIs (FREQ_ROI), with an average prior error as low as 0.19. This shows how time and day semantics help Adv profile commuters. MAX_ROI and MAX_USER strategies slightly enhance Adv's posterior knowledge and result in 0.08 and 0.14 mean privacy loss, respectively. Whereas, the Bayesian inference significantly improves Adv's performance towards



(a) TFL (b) SFC

FIGURE 6.3: TIME_DAY_WEEK prior – Adv's profiling error.

her profiling goal, yielding an average of 0.27 privacy loss for the users. With the SFC dataset (Fig. 6.2b), the average prior error is lower than with FREQ_ROI as time and day semantics enhance Adv's performance, but it still remains relatively high (0.61). Two of the inference strategies reduce Adv's error, although not dramatically: BAYES and MAX_USER help Adv to profile cabs' mobility and yield, resp., 0.03 and 0.07 privacy loss. In contrast, MAX_ROI actually deteriorates Adv's performance and does not harm the cabs' privacy. Overall, we notice that profiling cabs using their weekly profiles as prior knowledge is more challenging than profiling commuters whose mobility patterns are more regular.

TIME_DAY_WEEK. Our last experiments with probabilistic priors measure Adv's error (see Fig. 6.3) when her prior knowledge consists only of time information for the users, i.e., she knows which time slots of the inference week a user is likely to report ROIs, but not which specific ROIs. Similar experiments in which Adv knows which time slots of *any* day a user reports ROIs (TIME_DAY) result in larger error and are discussed in Appendix B.1. On the TFL dataset (Fig. 6.3a), the error based on this prior is larger than with ROI_DAY_WEEK, namely, 0.3. Greedy strategies (i.e., MAX_ROI and MAX_USER) remarkably improve Adv's performance (i.e., they result in 0.5 privacy loss on average), as in this case the users reporting the most ROIs are chosen to consume the aggregates (due to the prior, users have equal probability to appear in ROIs). On the other hand, the Bayesian inference only slightly decreases the adversarial error, due to the small probabilities of her prior, which consists of a uniform distribution over the tube stations for the users' most frequent time slots of a week.

With the SFC data (Fig. 6.3b), when Adv knows the cabs' most frequent time slots reporting ROIs, her prior error is larger compared to that of cabs' frequent ROIs over the time instances of a week (ROI_DAY_WEEK), i.e., 0.66. However, exploiting the aggregate knowledge the error is reduced and BAYES, MAX_ROI, and MAX_USER, yield $0.09, 0.1$, and 0.2, mean privacy loss, respectively. Overall, we point out that due to the different nature of the datasets (sparse TFL vs. dense SFC), user profiling with time information as prior knowledge yields different amounts of privacy leakage.

Assignment Priors

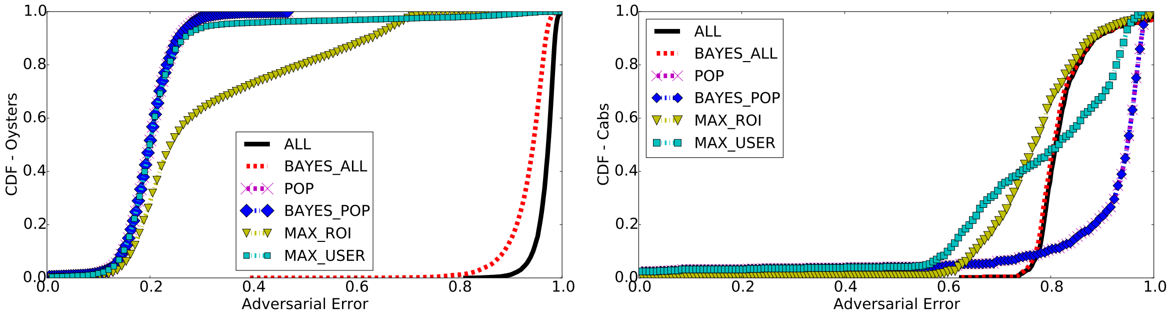
Next, we evaluate Adv's performance with assignment priors, i.e., when she obtains a historical location profile as her prior knowledge for the users. We experiment with LAST_WEEK, LAST_DAY, and LAST_HOUR, described in Section 6.1.2. Unlike probabilistic ones, the privacy loss from aggregates with assignment priors is very small, as the sliding window on the location time-series of commuters or cabs already yields highly informative priors. Since the CDF plots are less illustrative in this setting, we defer them to Appendix B.1 (Figures B.3–B.4).

Discussion

Overall, our experiments show that aggregates do help the adversary on the profiling inference goal. The actual degree of privacy loss for the users depends on the prior: assignment ones yield smaller privacy leakages, as they are already quite informative for the adversary compared to probabilistic ones. We also observe that inferring the mobility profiles of commuters from aggregates is significantly easier than profiling cabs. In other words, cabs' patterns are not as regular as those of tube passengers, who exhibit high seasonality. As a consequence, commuters lose much more privacy than cabs from aggregate locations.

6.2.3 User Localization

We now measure privacy loss in the context of *localization* attacks, i.e., as Adv attempts to predict users' future locations. Our experimental setup is the same as with profiling. However, Adv's output is not a probability distribution, but a binary localization matrix, and Adv's main performance metric is based on the F1 score of her predictions (recall Eq. 6.2).



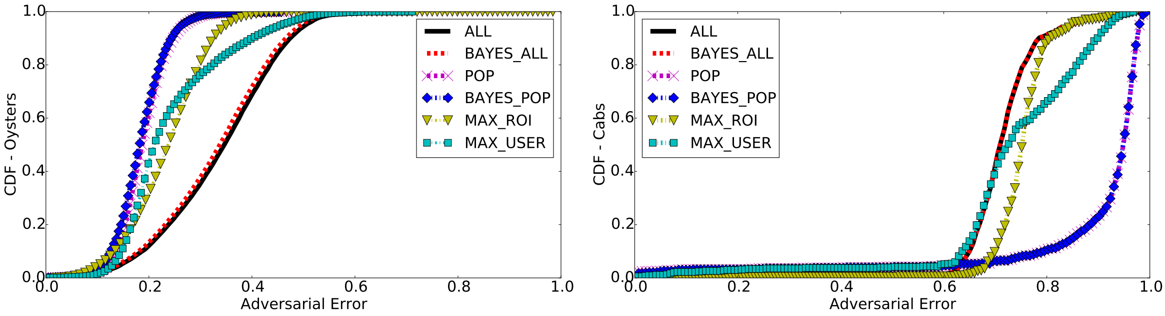
(a) TFL (b) SFC

FIGURE 6.4: FREQ_ROI prior – Adv's localization error.

Probabilistic Priors

We quantify Adv's error in localizing users when the prior knowledge matrix P is built according to users' most frequent ROIs over time (FREQ_ROI) and their most frequent ROIs for each time slot of a week (ROI_DAY_WEEK). Since the prior is a probability distribution over ROIs for each time slot of the inference period, Adv's baseline prediction is to extract the users' most popular prior ROIs (POP) or all prior ROIs (ALL) (Section 6.1.2). For POP, we set the threshold λ' to 0.5, i.e., we consider users' favorite ROIs those with more than 50% chance of visiting. As part of her inference strategy, Adv (a) applies BAYES and evaluates POP and ALL on its output, and (b) employs MAX_ROI and MAX_USER. Figures 6.4–6.5 plot the corresponding results, while additional experiments with Adv knowing users' most frequent time slots of a week (TIME_DAY_WEEK) are deferred to Appendix B.2.

FREQ_ROI. Figure 6.4a plots the CDF of Adv's error in localizing TFL passengers with their frequent ROIs over time as her prior knowledge. Using only the prior, i.e., predicting that users will appear in all their frequent ROIs (ALL), we get a very large average error (0.97); evaluating ALL after applying the Bayesian inference slightly reduces the adversarial error (0.93) and yields very small privacy loss (on average, 0.04). When predicting that commuters will appear in their most popular ROIs (POP), Adv's mean error drops to 0.21. Again, BAYES does not improve Adv's performance, as the prior probabilities are so small that, after updating, they do not exceed λ'. We observe that with POP, Adv predicts users to be out of the transportation system during the time slots of T_I. Interestingly, such a conservative strategy yields a small adversarial error overall, however, this occurs due to the fact that the TFL dataset is relatively *sparse*. With the greedy inference strategies (MAX_ROI and MAX_USER),



(a) TFL

(b) SFC

FIGURE 6.5: ROI_DAY_WEEK prior – Adv's localization error.

Adv's mean error is much smaller than ALL, respectively, 0.32 and 0.23. Their error patterns are different as they select different sets of users to cover the aggregates. More precisely, MAX_ROI achieves an error of 0.5 or less for 70% of the users, while MAX_USER for 90%. In both cases, Adv's error is reduced notably in comparison with the ALL baseline strategy, and we find that the aggregates do indeed yield substantial privacy loss (resp., 0.66 and 0.77).

In Fig. 6.4b, we plot the CDF of Adv's error while attempting to localize SFC cabs over T_I, again with the FREQ_ROI prior. Similar to the TFL experiments, when Adv extracts cabs' most popular prior ROIs (POP), she predicts all of them to be outside the network, since the prior probabilities are smaller than the threshold ($\lambda' = 0.5$), and the Bayesian inference updates them negligibly. However, unlike TFL, Adv's error with POP is 0.9 on average, proving it to be a bad strategy for localizing cabs. Predicting that cabs will show up in all their prior ROIs (ALL) slightly improves her predictive power as the mean error drops to 0.83, while BAYES negligibly reduces it further. Both MAX_ROI and MAX_USER inference strategies improve Adv's predictions compared to the ALL baseline, and they yield, resp., 0.08 and 0.11 privacy loss. However, we observe that MAX_ROI behaves more consistently than MAX_USER (which reduces Adv's error only for 50% of the cabs), indicating ROI regularity. Overall, it is clear that localization strategies behave quite differently on datasets of dissimilar characteristics.

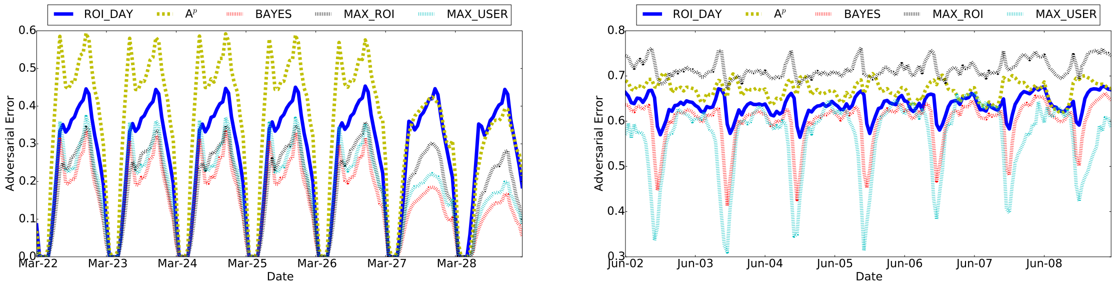
ROI_DAY_WEEK. Figure 6.5 displays the CDF of Adv's error localizing users with their most frequent ROIs for each time slot of a week (ROI_DAY_WEEK) as her prior knowledge. For TFL, we notice that all prior ROIs yield a mean adversarial error of 0.34 and Bayesian inference slightly reduces it, resulting to insignificant privacy loss (0.03). In this case, users' most popular prior ROIs (POP) reduce Adv's error to 0.19. The BAYES and POP inference results in a negligible mean privacy loss (0.06). In contrast, compared to ALL, MAX_ROI and MAX_USER generate a notable privacy loss (0.29 and 0.26 on average). MAX_USER yields larger errors for users that are selected to cover the aggregates, while the error gets smaller for those users that were not (i.e., because the aggregates were consumed). On the other hand, MAX_ROI predicts better than MAX_USER for 25% of users, who are highly regular in the ROIs they visit. In comparison to FREQ_ROI, ROI_DAY_WEEK enables Adv to localize commuters more efficiently, proving it to be a more informative prior.

With the SFC data (Fig. 6.5b), localizing cabs via all their prior ROIs (ALL) yields a mean error of 0.71, while BAYES reduces it insignificantly. Interestingly, with ROI_DAY_WEEK being an instructive prior, ALL proves to be the best strategy. Extracting the cabs most popular ROIs (POP) results in an average error of 0.9 confirming once again that this strategy does not perform well on the dense cab dataset. Furthermore, MAX_ROI and MAX_USER yield significant privacy loss (0.17 and 0.18, resp.), compared to the baseline POP.

Overall, our experiments demonstrate that Adv is more effective in localizing commuters or cabs with ROI_DAY_WEEK than FREQ_ROI, however, the privacy loss for individuals is smaller due to the more revealing prior knowledge.

Assignment Priors

Finally, we assume that Adv obtains a historical assignment prior for the users, i.e., we experiment with LAST_WEEK, LAST_DAY, and LAST_HOUR, priors in the context of the localization adversarial task. We defer the details of the corresponding results (and plots) to Appendix B.2. We find that TFL commuters are best localized with their last week's ROIs (average error is 0.24 with LAST_WEEK, 0.27 with LAST_DAY, and 0.31 with LAST_HOUR), whereas, SFC cabs with their last hour's ROIs (average error is 0.73 with LAST_WEEK, 0.71 with LAST_DAY, and 0.64 with LAST_HOUR). Moreover, as in the profiling case, the availability of aggregates yields limited privacy loss when the adversarial prior knowledge is built via assignments. This indicates that, since assignment priors are already quite instructive, the aggregates do not significantly improve Adv's knowledge of individual users' whereabouts.



(a) TFL

(b) SFC

FIGURE 6.6: ROI_DAY prior – Adv's hourly profiling error.

Discussion

Similar to profiling, localization inferences performed using the aggregates yield different degrees of loss in privacy for individual users, depending on Adv's prior knowledge. Assignment priors are more revealing than probabilistic ones, thus aggregates end up leaking less privacy overall. We also observe that commuters are best localized via their most popular ROIs (POP), while cabs by their most recent ROIs (LAST_HOUR). Once again, localizing commuters is easier than localizing cabs as the former ones exhibit seasonality, while the latter ones have irregular patterns.

6.2.4 Privacy Implications of Regular Mobility Patterns

Experimenting with our framework also provides some interesting considerations about Adv's error over the time slots of the inference week. One would expect the leakage to vary according to time of the day (e.g., peak hours vs. night) or days of the week (e.g., weekdays vs. weekends) since the number of users in the system, and their concentration, varies significantly. As such, users would have variable levels of privacy loss over time. In order to validate this intuition, we pick a case study out of our experimental setup and examine the patterns in Adv's mean error during the hourly time slots of the inference week. Figure 6.6 plots the evolution of the adversarial profiling error over time for tube passengers (TFL) and taxis (SFC), when Adv obtains their most frequent ROIs for the time slots of day (ROI_DAY) as prior knowledge.

For TFL (Fig. 6.6a), we observe different patterns with respect to hours of the day and weekdays, as expected. Only considering the prior (ROI_DAY), Adv's error is smaller in the morning hours than mid-day or evening hours, likely because tube passengers are regular in their commuting routines to work, while in the evening they might go to the gym, meet friends, or go shopping before traveling back home. As the aggregate time-series is available to the adversary, her error is reduced during morning hours but not as much as in mid-day and evening hours. In other words, commuters lose more privacy if they travel during mid-day, as there are fewer users in the transportation system, or in the evening hours, because the aggregates reflect their irregular mobility pattern. Similarly, we observe that the aggregates give Adv a much more significant advantage during the weekends than on weekdays, as commuters more likely follow variable routes.

Likewise, for SFC (Fig. 6.6b), we observe distinct patterns in Adv's error with respect to hours of the day and weekdays. With the prior (ROI_DAY), Adv's error has a spike in the morning peak hours of weekdays indicating that cabs follow variable routes at these times and are not highly predictable. We find that Adv's prior error is smaller (0.57) during mid-day hours (i.e., 12pm–4pm) as cabs might be parked waiting for clients, or fewer routes might be performed during that shift. Indeed, the availability of the aggregate time-series harms cabs' privacy more during mid-day time slots as BAYES and MAX_USER reduce the adversarial error significantly (i.e., they yield higher privacy loss). Finally, we note that, among the inference strategies, MAX_USER gives Adv remarkable advantage in profiling cabs during weekends as the cabs reporting the most ROIs are likely to follow routes that are reflected by the aggregates.

6.3 Evaluation of Defenses

In Section 6.2, we demonstrated that aggregate location time-series leak information about individuals' whereabouts, and have evaluated how, based on different types of prior knowledge and inference strategies. We now use our proposed framework to study whether mechanisms supporting the release of aggregate information in a privacy-respecting manner are effective at avoiding such privacy leakage, and to what extent. Specifically, we focus on the protection offered by Differential Privacy (DP, see Section 2.2), using either output or input perturbation techniques. The former add noise to the output of the aggregation process, whereas, with the latter, noise is added to users' inputs before aggregation.

In theory, one can assess the level of privacy provided by DP mechanisms as it is configured by the parameter ϵ, which determines the privacy risk incurred when releasing statistics computed on sensitive data (providing an upper bound). However, DP's definition bounds the privacy leakage resulting from the inclusion of any dataset record, thus, it is uncertain how to calculate this bound in the presence of adversaries who know which records were used for a computation and exploit its output to improve their prior knowledge. Nonetheless, since DP provides protection against arbitrary risks, evaluating it in our framework is a meaningful exercise. Furthermore, while ϵ expresses the relation between the level of privacy before and after the release, it is not an absolute measure of privacy and it is often not clear how to interpret, in practice, the actual level of privacy enjoyed by individuals in the dataset, nor is how to choose the value of ϵ to obtain the desired protection.

In the rest of this section, we use our framework to measure to which extent DP mechanisms reduce the privacy leakage compared to the release of raw aggregates, vis-à-vis the resulting utility of the data. That is, we quantify the protection that these mechanisms provide to users in presence of an adversary that, as in Section 6.2, has access to the aggregates (now perturbed via a DP mechanism) and uses that information to improve her prior knowledge about users' whereabouts.

6.3.1 Privacy and Utility Metrics

Privacy Gain (PG). We quantify the protection provided by DP techniques in terms of the privacy gain they yield, which we define as the difference between Adv's error when using her prior (P) with the noisy aggregates A' ($\text{AdvErr}_{P,A'}$) and that with the raw aggregates A ($\text{AdvErr}_{P,A}$), normalized by the maximum gain the mechanism can provide. That is, we measure privacy gain (PG) as:

$$\text{PG} = \begin{cases} \frac{\text{AdvErr}_{P,A'} - \text{AdvErr}_{P,A}}{1 - \text{AdvErr}_{P,A}} & \text{if } \text{AdvErr}_{P,A} \neq 1 \ \wedge \text{AdvErr}_{P,A'} > \text{AdvErr}_{P,A} \\ 0 & \text{otherwise} \end{cases} \tag{6.13}$$

PG is a value between 0 and 1 capturing Adv's deterioration towards her goal (e.g., profiling users) owing to the noise added by the DP techniques.

Mean Relative Error (MRE). We also use the MRE (Section 2.3.1) to measure utility loss, specifically, to capture the error between the original aggregate location time-series and their noisy version, which comes as the result of perturbation. We calculate the MRE over each ROI $s \in S$ in our datasets, and report the mean value.

6.3.2 Output Perturbation

We first evaluate differentially private mechanisms based on output perturbation, in which an entity adds noise to the statistics prior to their release. This entity can be trusted with the individual users' data [AC14; FX12] or only be allowed to compute aggregate statistics, e.g., using cryptographic protocols for private aggregation as done in previous work [Shi+10; Pop+11] and in Chapter 5. We evaluate the following defenses: the Laplace mechanism (LPM), the Fourier Perturbation Algorithm (FPA), and the Simple Counting Mechanism (SCM), which we review in Section 2.2.2.

We present the results of our evaluation on two case studies: (a) user profiling on the TFL dataset with Adv obtaining FREQ_ROI as her prior knowledge and following the greedy MAX_ROI strategy, and (b) user profiling on the SFC data when Adv knows FREQ_ROI and employs MAX_USER. Although we restrict to two cases, their choice is reasonable as our analysis in Section 6.2 shows that, in these settings, the aggregates yield significant privacy loss for individual users.

We calculate the sensitivity for the users in each dataset, i.e., the maximum number of events reported by an oyster or a cab during the inference period T_I (224 and 1,997 for TFL and SFC, resp.). Furthermore, we parameterize the LPM, FPA, and SCM, perturbation mechanisms with $\epsilon \in \{0.001, 0.01, 0.1, 1.0\}$. For FPA, as done in [RN10], we experiment with the parameter κ to minimize its total error, finding that $\kappa = 25$ yields the best results on TFL and $\kappa = 20$ on the SFC data.

Utility loss. Tables 6.3 and 6.4 report the utility loss for both datasets, in terms of MRE, of the mechanisms for different values of ϵ. Overall, as expected, for all mechanisms the higher the privacy (i.e., lower ϵ values), the higher the utility loss (i.e., bigger MRE).

ϵ	**0.001**	**0.01**	**0.1**	**1.0**
LPM	3,778.7	3,326.8	882.3	89.2
FPA	1,085.1	107.8	10.6	0.8
SCM	447.6	43.8	4.2	0.3

TABLE 6.3: MRE (utility loss) of output perturbation mechanisms on the TFL dataset.

ϵ	**0.001**	**0.01**	**0.1**	**1.0**
LPM	127.4	125.3	119.8	101.6
FPA	122.4	90.5	12.6	1.4
SCM	97.2	22.7	2.4	0.2

TABLE 6.4: MRE (utility loss) of output perturbation mechanisms on the SFC dataset.

In our first case study (Table 6.3), LPM yields the worse utility, with the perturbed aggregates being more than 80 times worse than the raw ones, even for a mild level of expected privacy ($\epsilon = 1.0$). The highest utility is provided by SCM, followed by FPA. Nonetheless, even with the former, the utility is at least 4 times worse than the raw aggregates (MRE $= 4.2$) for small ϵ values (0.1 or less). In our second case study (Table 6.4), we observe that LPM results in very large errors (MRE ≥ 101). Once again, FPA and SCM yield smaller utility loss, although for sensible levels of privacy (i.e., $\epsilon = 0.1$) the perturbed aggregates are about 12 and 2 times worse estimates than the raw ones, respectively.

Privacy gain. Figs. 6.7 and 6.8 display box plots of the privacy gain (PG, see Eq. 6.13) enjoyed by individual users in both datasets thanks to the perturbation mechanisms, for increasing values of ϵ. In the TFL case study (Fig. 6.7), the three mechanisms exhibit very different behaviors. LPM offers the best privacy protection, with an average privacy gain as high as 0.77 for $\epsilon \leq 0.1$, and 0.65 for $\epsilon = 1.0$. However, as discussed above (and shown in Table 6.3), this protection comes with very poor utility. We also find that FPA and SCM offer similar protection (PG $= 0.74$ on average) for $\epsilon = 0.001$, while, as ϵ grows, the privacy gain drops significantly, being negligible when $\epsilon = 1.0$. While this is somewhat expected for SCM which guarantees only *event-level* privacy, it is surprising for FPA, which in theory should provide as much protection as LPM (recall that both mechanisms achieve ϵ-DP). Similarly, in the SFC case (Fig. 6.8), we observe that LPM provides the best privacy gain (0.36 on avg.) for all values of ϵ. FPA and SCM behave similarly to LPM for $\epsilon \leq 0.01$, however, as ϵ increases the privacy

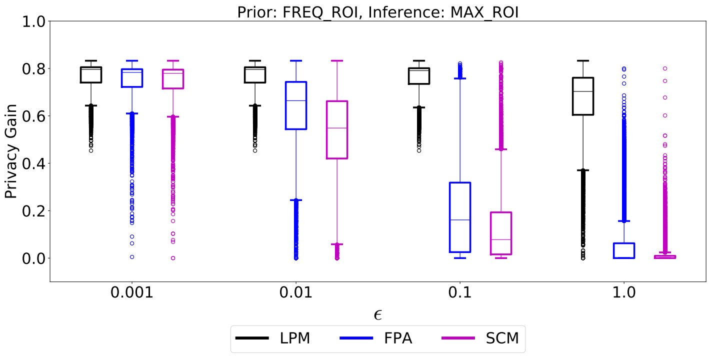


FIGURE 6.7: Privacy Gain (PG) achieved by output perturbation DP mechanisms on the TFL dataset.

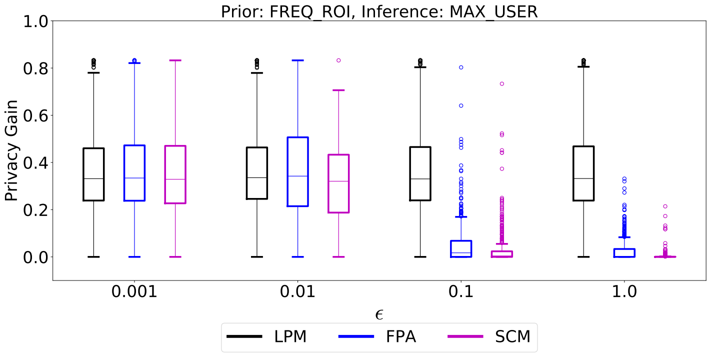


FIGURE 6.8: Privacy Gain (PG) achieved by output perturbation DP mechanisms on the SFC dataset.

gain approaches zero.

6.3.3 Input Perturbation

We now evaluate input perturbation DP techniques, whereby users add noise to their inputs prior to the aggregation process. In particular, we focus on randomized response techniques which are commonly used to privately collect statistics from users participating in surveys [War65], crowdsource statistics from client software [EPK14], as well as privately aggregate user locations in real-time [Que+11]. In particular, the SpotMe system [Que+11] lets users perturb their location at each time instance $t' \in T_I$ by claiming to be in a ROI $s \in S$ (a *yes* response) with some probability π, or report the truth (i.e., whether they are or not in location s) with probability $1 - \pi$. The aggregator collects the perturbed user inputs and computes the aggregation estimating the number of individuals in each location $s \in S$ and every time slot $t' \in T_I$, via $a_{s,t'} = \text{total}_{s,t'} \cdot \frac{\text{Pyes}_{s,t'} - \pi}{1-\pi}$, where $\text{total}_{s,t'}$ is the total number of responses

π	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
TFL	9.6	22.3	33.6	52.7	103.5
SFC	0.9	2.4	4.1	6.2	12.9

TABLE 6.5: SpotMe [Que+11]: MRE (utility loss) for increasing values of π on the TFL and SFC datasets.

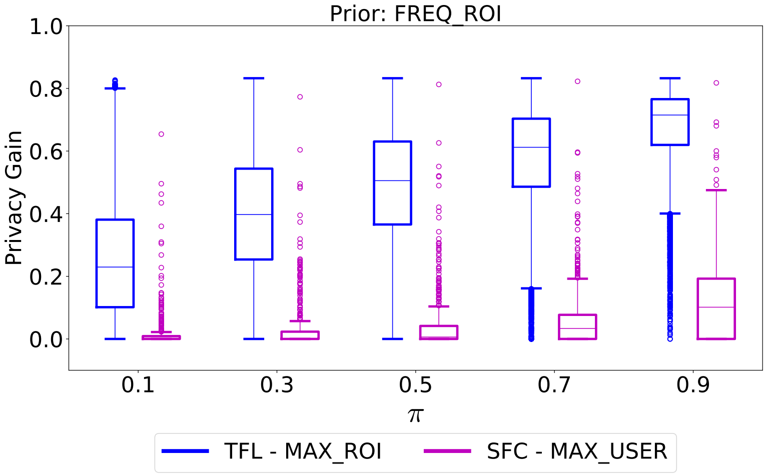


FIGURE 6.9: Privacy Gain (PG) achieved by SpotMe [Que+11] for increasing values of π, on the TFL and SFC datasets.

received for ROI s at time t' and $\text{Pyes}_{s,t'} = \frac{\text{yes}_{s,t'}}{\text{total}_{s,t'}}$ depicts the proportion of *yes* responses. This mechanism is $\ln \frac{|S|-(|S|-1)\cdot\pi}{\pi}$-DP at each time slot [WN16], thus, overall it guarantees $O(|T_I| \cdot \ln \frac{|S|-(|S|-1)\cdot\pi}{\pi})$ differential privacy due to the composition theorem (Section 2.2).

We evaluate SpotMe [Que+11], as a representative for randomized response based input perturbation mechanisms, using our framework. In this context, Adv is assumed to obtain the perturbed aggregates A' estimated by the aggregator after the mobile users apply the mechanism on their inputs. As in the output perturbation case, we focus on two user profiling case studies: (a) TFL data with FREQ_ROI adversarial prior knowledge and MAX_ROI inference, and (b) SFC dataset with FREQ_ROI prior and MAX_USER strategy.

Utility loss. Table 6.5 shows the MRE of the perturbed aggregates, highlighting that, as π grows (i.e., as commuters or cabs perturb their inputs with higher probability) the utility loss of the aggregates increases. For TFL, with $\pi = 0.1$, the MRE over all stations is 9.6, while it reaches 103.5 when π is set to 0.9. For SFC, the MRE over the ROIs is 0.9 with $\pi = 0.1$, while when $\pi = 0.9$ the perturbed aggregates are approximately 12 times worse than the raw ones.

Privacy gain. Fig. 6.9 plots the privacy gain provided by the SpotMe mechanism with respect to the parameter π, for the users of both TFL and SFC datasets. For TFL, we observe that, as π increases, PG also increments, reaching up to 0.6 on average, with the most conservative parameterization ($\pi = 0.9$). Interestingly, for SFC, we observe that, as π grows, the privacy gain only increases negligibly. For $\pi = 0.5$, the average PG is 0.04, while it's only 0.1 when $\pi = 0.9$, highlighting the challenges of using this mechanism on dense datasets with small number of users.

6.3.4 Discussion

Our evaluation of defense mechanisms based on differential privacy (DP) highlights the difficulty to fine-tune the trade-off between privacy and utility. More specifically, our case studies show that applying existing DP mechanisms in a straightforward manner yields poor utility in the context of aggregate location time-series and for the settings considered in this work, i.e., mobility analytics over transport data. Moreover, we find that mechanisms which reduce the required amount of noise do not provide acceptable privacy protection, which is not surprising given that the adversarial model considered in our framework is not in accordance with DP's indistinguishability based definition.

Moreover, although the generic framework of differential privacy abstracts from adversarial prior knowledge, our experimental evaluation indicates that the concrete nature of this prior should be taken into account when evaluating defense mechanisms. While some priors may not be useful for the adversary, our experiments show that realistic approaches of building adversarial prior knowledge, for example considering users' frequent locations, can help an adversary when performing inference attacks to extract knowledge, even from aggregates which have been perturbed with a differentially private mechanism.

Furthermore, we observe that the performance of DP mechanisms in terms of privacy and utility is also highly dependent on the intrinsic characteristics of the datasets used in our experiments. For instance, on the sparse TFL dataset containing thousands of users moving among a relatively large number of ROIs (583), output and input perturbation achieve reasonable levels of privacy, with the latter performing only slightly better than the former in terms of utility. On the other hand, on the dense SFC dataset, which includes fewer users and ROIs (101), output perturbation does not yield significant privacy protection, while input perturbation only a negligible one. Overall, in the SFC setting, the utility loss introduced by DP mechanisms is somewhat smaller compared to the TFL one.

Finally, data pre-processing techniques, e.g., sub-sampling and clustering which are employed in [AC14], could theoretically be used to improve the utility of DP mechanisms (e.g., by reducing the number of locations reported by the users or merging sparse ROIs together), however, such an approach is application dependent and cannot be considered a generalizable solution.

Chapter 7

Membership Inference on Aggregate Location Data

In Chapter 6, we demonstrated that aggregate location data can help an adversary, who has some prior knowledge about mobile users, to profile or localize them. In our proposed framework geared to quantify privacy loss stemming from the aggregates, we have made an implicit assumption that the adversary knows that a target user's location time-series has been used to compute the aggregates, i.e., she knows that the target user is part of the database under examination.

In this chapter, we investigate whether such an assumption is realistic. In particular, we study the feasibility of membership inference attacks on aggregate location time-series, whereby the aim of the adversary is to infer if a user's location data was used to compute a piece of aggregate location data. We model the problem as a distinguishability game and we present a methodology that instantiates the adversarial task with a machine learning classifier trained on the adversarial prior knowledge. We then deploy our methods on the real-world mobility datasets described in Chapter 4 aiming to quantify privacy leakage stemming from releasing raw aggregate location data. We then illustrate how our techniques can be used to evaluate the privacy protection offered by defense mechanisms that guarantee differential privacy with respect to the utility loss that they introduce.

Motivation. As discussed above, studying membership inference is motivated by research efforts that show how an adversary can exploit aggregate location data to infer user profiles (e.g., as we do in Chapter 6) or to extract mobility trajectories [Xu+17]. To mount such attacks, the adversary needs to know that the data of a user is part of the aggregate dataset.

Furthermore, the ability of an adversary to ascertain the presence of an individual in aggregate location time-series constitutes an obvious privacy threat if the aggregates relate to a group of users that share a sensitive characteristic. For instance, learning that an individual is part of a dataset aggregating movements of Alzheimer's patients or mobility patterns of users with certain income implies learning that she suffers from the disease or her income. Similarly, inferring that statistics collected over a sensitive timeframe or sensitive locations include a particular user also harm the individual's privacy.

Finally, membership inference can also be leveraged by providers to evaluate the quality of privacy protection on the aggregates *before* releasing them, and by regulators, to support enforcement of individual's rights (e.g., the right to be forgotten) or to detect violations. For instance, if a service provider is not allowed to release location data, or make it available to third-parties even in aggregate form, one can use membership inference attacks to verify possible misuse of the data.

7.1 Modeling Membership Inference on Aggregate Location Data

We first describe how we model the problem of membership inference on aggregate location data.

7.1.1 Notation

The notation used throughout this chapter is summarized in Table 7.1. As in previous chapters, we denote the set of mobile users as $U = \{u_1, u_2, \cdots, u_{|U|}\}$, and the set of regions of interest they visit as $S = \{s_1, s_2, \cdots, s_{|S|}\}$. We also use $T = \{t_1, t_2, \cdots, t_{|T|}\}$ to denote the set of time intervals on which aggregate location data is collected. We model the location of a user $u \in U$ over time as a binary matrix L_u[1] of size $|S| \times |T|$, where $l_{s,t} \in L_u$ is 1 if u is in location $s \in S$, at time $t \in T$, and 0 otherwise. That is, L_u contains the location time-series of u, while those of all users are depicted with the matrix L_U, which is of size $|U| \times |S| \times |T|$.

[1] We omit the subindex when there is no ambiguity.

Symbol	Description
Adv, Ch	Adversary, Challenger
P	Adv's prior knowledge
U	Set of mobile users
S	Set of locations (ROIs)
T	Time period considered
T_O	Observation period
T_I	Inference period
L_u	User u's location time-series
L_U	Location time-series of all users in U
$Y \subset_\$ U$	Random subset $Y \subset U$
A^X	Aggregate location time-series of users in $X \subset U$
m	Aggregation group size

TABLE 7.1: Notation.

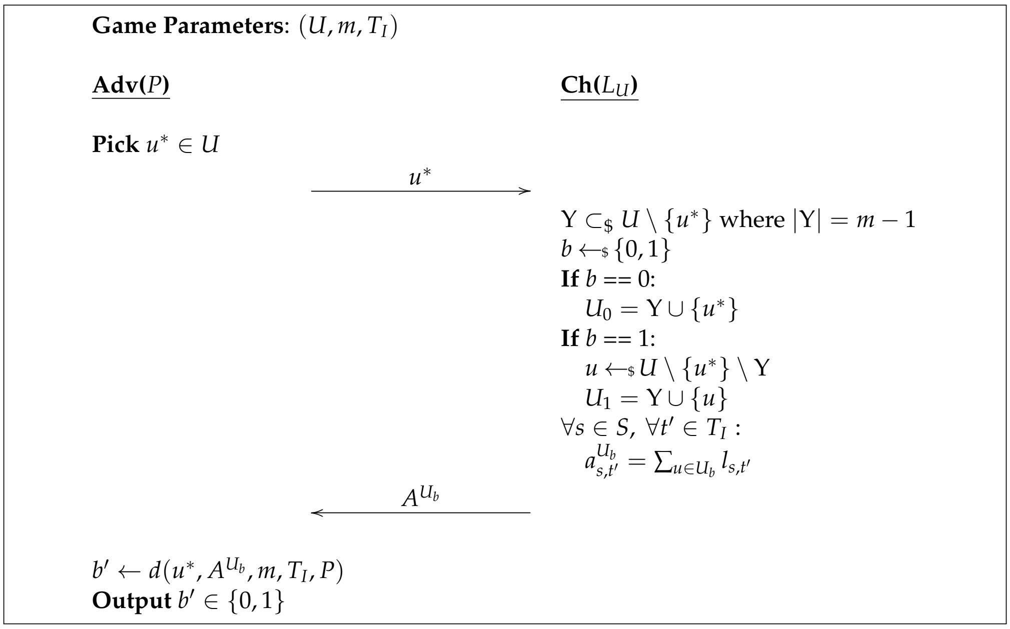
Furthermore, A^X denotes the aggregate location time-series computed over the users in $X \subset U$. A^X is modeled as a matrix of size $|S| \times |T|$, where each element $a^X_{s,t} \in A^X$ represents the number of users in X that are in ROI s, at time t. Finally, we denote the prior knowledge that an adversary (Adv) may have about users as P, which is built during an *observation* period, denoted as $T_O \subset T$. The prior knowledge is used by Adv to perform membership inference during the *inference* period, $T_I \subset T$, for which aggregate location data is available.

7.1.2 Distinguishability Game

We model membership inference by means of a distinguishability game (DG), played by the adversary Adv and a challenger Ch, which computes aggregate locations over various user groups. The former, having some prior knowledge about the users (P), tries to infer whether data of a particular user (u^*) is included in the aggregates. Naturally, Adv could be interested in multiple target users, however, to ease presentation, we describe the case of a single target user.

The game is parameterized by the set of users U, the number of users included in the aggregation group (m), and the inference period T_I. Note that m and T_I inherently affect Adv's performance, as we discuss in our experimental evaluation (Section 7.3).

We present the game in Figure 7.1. First, Adv selects the target user u^* from the set of users U, and sends it to Ch. The latter randomly selects a subset $Y \subset U$ of size $m - 1$, excluding u^*, and draws a random bit b. If $b = 0$, she aggregates the location matrices of all users in



```
Game Parameters: $(U, m, T_I)$

Adv($P$)                                   Ch($L_U$)

Pick $u^* \in U$
                 $\xrightarrow{u^*}$
                                           $Y \subset_{\$} U \setminus \{u^*\}$ where $|Y| = m - 1$
                                           $b \leftarrow_{\$} \{0,1\}$
                                           If $b == 0$:
                                             $U_0 = Y \cup \{u^*\}$
                                           If $b == 1$:
                                             $u \leftarrow_{\$} U \setminus \{u^*\} \setminus Y$
                                             $U_1 = Y \cup \{u\}$
                                           $\forall s \in S,\ \forall t' \in T_I$ :
                                             $a^{U_b}_{s,t'} = \sum_{u \in U_b} l_{s,t'}$
                 $\xleftarrow{A^{U_b}}$

$b' \leftarrow d(u^*, A^{U_b}, m, T_I, P)$
Output $b' \in \{0,1\}$
```

FIGURE 7.1: Distinguishability Game (DG) between the adversary (Adv) and the challenger (Ch), capturing membership inference on aggregate location time-series. The game is parameterized by the set of users (U), the aggregation group size (m), and the inference period (T_I).

Y along with that of u^*; whereas, if $b = 1$, she selects another random user $u \neq u^*$ not in Y and adds her data to the aggregates instead. The resulting matrix A^{U_b}, computed over all time slots of T_I, is sent back to Adv, which attempts to guess b. Adv wins if $b' = b$, i.e., she successfully distinguishes whether u^* is part of the aggregates or not; naturally, her goal is to win the game, over multiple iterations, with probability higher than 1/2 (i.e., a random guess).

We model Adv's guess as a *distinguishing function* d, with input $(u^*, A^{U_b}, m, T_I, P)$. How to instantiate the function is discussed in Section 7.1.4. Note that the parameters of the DG game include the set of users U, but this information is *not* used in the distinguishing function. In other words, we only assume that Adv knows that u^* is in the universe of possible mobile users, but not that she knows all users in U.

7.1.3 Adversarial Prior Knowledge

Our generic game-based definition of the adversarial goal enables the consideration of adversaries of variable strength, modeled by their prior knowledge, P. We consider two possible priors, discussed next.

Subset of Locations

We start with a setting in which Adv knows the real locations of a subset of users $Y \subset U$, *including* the target user (i.e., $u^* \in Y$), during the inference period T_I. Thus, in this case observation and inference periods coincide (i.e., $T_O = T_I$). We consider $|Y| = \alpha \cdot |U|$, where $\alpha \in [0, 1]$ models the percentage of users for which Adv knows their actual location time-series. Formally, we define the prior knowledge as:

$$P : \; l_{s,t} \in L_u \quad \forall u \in Y, \, \forall s \in S, \, \forall t' \in T_I \tag{7.1}$$

This type of prior knowledge represents the case of an adversary that has access to location information of some users at a point in time, e.g., a telecommunications service provider getting locations from cell towers, or a mobile app provider collecting location data. Using this information she attempts to infer membership of her target to an aggregate dataset published by another entity.

Participation in Past Groups

We then consider an adversary that knows aggregates computed during an observation period T_O, disjoint from the inference period T_I (i.e., $T_O \cap T_I = \emptyset$) for β groups W_i of size m, which may or may not include u^*. For each group W_i, we assume that Adv knows: (a) the aggregates of the observation period, i.e., $a_{s,t}^{W_i} \in A^{W_i} \;\; \forall s \in S, \, \forall t \in T_O$, and (b) u^*'s membership to the group. More formally:

$$P : A^{W_i} \, \wedge \, \mathbb{1}^{W_i}(u^*) \quad \forall \, i \in \{1, \cdots, \beta\} \tag{7.2}$$

where $\mathbb{1}^{W_i}(u^*)$ is the indicator function modeling the membership of the target user to the group W_i. In our experiments, we consider two different *flavors* of this prior:

– **(a) Same Groups as Released:** Adv knows the target user's participation in past groups which *are* also used to compute the aggregates released by Ch during the inference period;

- **(b) Different Groups than Released:** Adv knows the user's participation in past groups that *are not* used to compute the aggregates released in the inference period.

Observe that prior (a) simulates the case of continuous data release related to particular groups, where users are stable over time (e.g., statistics about a neighborhood), and with the adversary (e.g., a group member) having observed the participation of the target user in past aggregates of the same groups. Prior (b) is less restrictive, as it only assumes that the adversary has some aggregates of groups in which the target was previously included, but does not require these groups to be fixed over time – e.g., if the target user moves to a new neighborhood and her data is mixed with other users, Adv attempts to infer membership using her past information.

7.1.4 Distinguishing Function

In the distinguishability game (DG, Fig. 7.1), the adversary tries to guess whether or not the target user's data is part of the location aggregates using a distinguishing function, which we denote as d. This function takes as input the target user u^*, the *challenge* A^{U_b}, the parameters of the game m and T_I, as well as the prior knowledge P.

We opt to instantiate d with a *supervised machine learning classifier*, trained using data that is included in the adversarial prior knowledge. Our intuition is that the adversary's distinguishing goal can be modeled as a binary classification task, i.e., categorizing observations into two classes corresponding to whether or not the data of target user u^* is part of the location aggregates under examination.

7.1.5 Privacy Metric

Given our game-based definition, we reason about privacy leakage in terms of the adversarial performance in distinguishing whether or not u^*'s data is included in the aggregates. In particular, we introduce a privacy loss metric, capturing Adv's advantage in winning the DG game over a random guess (assuming that the adversary plays the distinguishability game for a specific user multiple times), while relying on the distinguishing function's (or classifier's, as described in Section 7.1.4) Area Under the Curve (AUC, see Section 2.1.2) to measure the adversarial performance.

AUC score. For a series of instances of the game for u^*, we count the Adv's guesses b' regarding the presence of u^*'s data in the released aggregate location time-series as:

- True Positive (TP), when $b = 0$ and $b' = 0$;
- True Negative (TN), when $b = 1$ and $b' = 1$;
- False Positive (FP), when $b = 1$ and $b' = 0$;
- False Negative (FN), when $b = 0$ and $b' = 1$.

We then calculate the classifier's True Positive and False Positive Rate (TPR and FPR, resp.), which we reviewed in Section 2.1.2. From these, we derive the Receiver Operating Characteristic (ROC) curve, which represents the TPR and FPR obtained at various discrimination classification thresholds, and compute the Area Under Curve (AUC). The AUC score captures the classifier's overall performance in the distinguishability game.

Privacy Loss (PL). As mentioned, we measure the privacy loss of u^* as the adversary's *improvement* over a random guess baseline (AUC $= 0.5$). Formally, we define PL as:

$$\mathrm{PL} = \begin{cases} \frac{\mathrm{AUC}-0.5}{0.5} & \text{if } \mathrm{AUC} > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{7.3}$$

Hence, PL is a value between 0 and 1 that captures the adversary's advantage over random guessing when distinguishing whether the target user's data is part of the aggregates.

7.2 Experiment Design

Datasets and sampling. To evaluate membership inference on aggregate location data, we employ the real-world TFL and SFC mobility datasets reviewed in Chapter 4. To select the users (i.e., victims) against which we run membership inference attacks we perform an analysis of the number of events reported by the users in the TFL and SFC datasets. We observe that for TFL (resp., SFC), the median is 727 (resp., 4,111), with a maximum of 881 (resp., 8,136) and a minimum of 673 (resp., 504). We sort the users in each dataset per total number of events and split them in 3 groups of equal size, capturing their mobility patterns as: *highly*, *mildly*, and *somewhat*, mobile. To avoid bias while selecting target users, we sample 50 users from each mobility group *at random*. Thus, we run membership inference attacks against a total of 150 users in each dataset.

Experiment setup. Our experiments aim to evaluate the effectiveness of the distinguishing function d, used in the DG game, to guess whether the target user u^* is in the aggregates or not. As mentioned in Section 7.1.4, we instantiate d using a machine learning classifier.

We train the classifier on a *balanced* dataset of *labeled* aggregate location time-series computed over user groups that include and groups that exclude the target user u^*, so that it learns patterns that distinguish her participation in the aggregates. The training dataset is generated using data from the prior knowledge P. We then play the game, i.e., we use the trained classifier to infer membership on a *balanced* testing set of aggregates previously *unseen*.

More specifically, our experimental setup goes through three phases: *aggregation*, *feature extraction*, and *classification*, which we describe in high-level. The concrete details of each phase depend on the adversarial prior knowledge, and are presented in Section 7.3 (where we evaluate membership inference attacks with different priors). The three phases are discussed next.

Aggregation. For a victim u^*, we create a dataset D by repeating these steps:

1. Randomly generate a group U_0 of m users, which *includes* u^*;
2. Aggregate the location matrices of users in U_0, for $|T_I|$ intervals;
3. Append a row with the aggregates A^{U_0} to dataset D, and attach the label *in*;
4. Randomly generate a group U_1 of m users, which *excludes* u^*;
5. Aggregate the location matrices of users in U_1, for $|T_I|$ intervals;
6. Append a row with the aggregates A^{U_1} to the dataset D, and attach the label *out*.

Feature extraction. For each row of the dataset, corresponding to the aggregates of a group with or without u^*, we extract statistics that are given as input to the classifier. Such statistics are calculated per location (ROI) and include variance, minimum, maximum, median, mean, standard deviation, as well as the sum of values of each location's time-series.

Classification. We first split the dataset D into the non-overlapping balanced training and testing sets mentioned earlier. We then train the classifier on the features extracted from the *training* set. Finally, we play the distinguishability game on the aggregates of the *testing* set (data previously unseen by the classifier), classifying them as including or excluding u^*'s data.

Implementation. Our experiments are implemented in Python using the *scikit-learn* machine learning suite[2]. We instantiate the distinguishing function with the following classifiers: (a) Logistic Regression (LR), for which we employ a linear solver using a coordinate descent optimization algorithm suitable for binary classification [Fan+08]; (b) Nearest Neighbors (k-NN), configured to use Euclidean distance, with k set to 5, i.e., to predict the output class based on the votes of the 5 nearest samples; (c) Random Forests (RF), set up to train 30 decision trees and to consider all the features during the node splits using the Gini criterion to measure their quality; and (d) Multi-Layer Perceptron (MLP), consisting of 1 hidden layer with 200 nodes, whose weights are calculated via a stochastic gradient-based optimizer. For more details about the classifiers, we refer to Section 2.1.2.

For the feature extraction, we use the tsfresh[3] Python package. For both datasets, and for all groups' aggregate location data, we extract the 7 statistical features mentioned above, for each ROI. We obtain 4,081 features for TFL (583 ROIs) and 707 features for SFC (101 ROIs). To avoid overfitting, we use Recursive Feature Elimination (RFE) to reduce the number of features to the number of samples we create for each user's dataset D. We then feed the features in their original form to all classifiers, except for MLP where we standardize them to have mean of 0 and variance 1.

7.3 Membership Inference on Raw Aggregate Location Data

We now present the results of our experimental evaluation, measuring the performance of different classifiers in instantiating the distinguishing function (i.e., performing membership inference) on raw aggregate location data. We do so with respect to the different priors discussed in Section 7.1.3, using the experimental design described in Section 7.2. Recall that, for

[2] http://scikit-learn.org/stable/

[3] http://tsfresh.readthedocs.io/en/latest/

our experiments, we perform membership inference attacks against 150 users sampled from high, mild, and somewhat, mobility profiles (50 from each case).

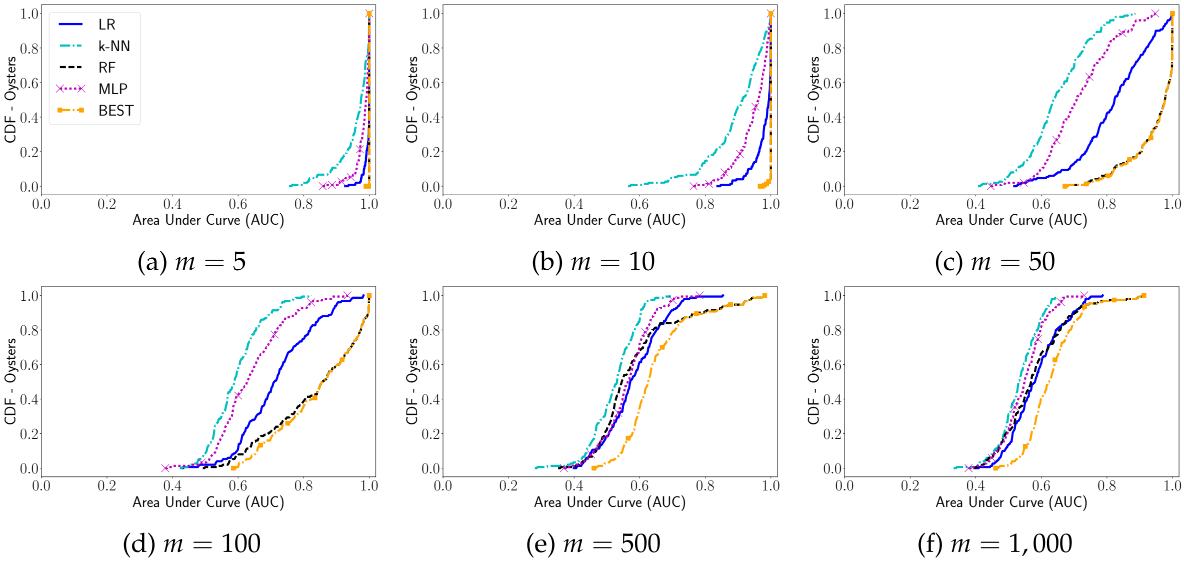
7.3.1 Subset of Locations

We start with the setting where the *observation* and *inference* periods coincide, and the adversary knows the location time-series for a subset of users, including the target, during this period. This information can then be used by Adv to create groups, with and without her target, and train a classifier. We consider as the observation and inference period, the *first week* of both TFL and SFC datasets, i.e., $|T_O| = |T_I| = 24 \cdot 7 = 168$ hourly time slots. We build Adv's prior by setting $\alpha = 0.11$ for TFL and $\alpha = 0.2$ for SFC, i.e, we randomly choose 1,100 out of 10,000 TFL users and 106 out of 534 SFC cabs. This represents a setting where Adv (e.g., a telecommunications service provider) knows location information for a small subset of users, including her target.

We then generate (a) a balanced training dataset by randomly sampling 400 *unique* user groups from Adv's prior knowledge, whereby half include the target user and half exclude it; and (b) a balanced testing set by sampling 100 unique user groups from the set of users that are not in the prior knowledge (apart from the target user). Our choice for training and testing sizes (400 and 100, resp.) is so that the datasets are large enough to enable learning and evaluation, and experiments to run in reasonable time. Finally, we extract features from the aggregates of both training and testing groups, labeling them as per the participation of the target in the group, and perform experiments with different values of m in order to evaluate the effect of aggregation group size.

TFL. Figure 7.2 plots the CDF, computed over the 150 target TFL users, of the AUC score achieved by the classifiers for different values of m. Limited by the adversarial knowledge (1,100 users), we examine aggregation group sizes up to 1,000. Note that the orange line labeled as BEST represents a hypothetical best case in which Adv chooses the classifier that yields the highest AUC score for each target user.

When groups are small, i.e., $m = 5$ or 10, all classifiers achieve very high AUC scores. For instance, with $m = 10$, Logistic Regression (LR) and Random Forest (RF) achieve a mean AUC score of 0.97 and 0.99, respectively. This indicates that for such small groups, where users'



(a) $m = 5$

(b) $m = 10$

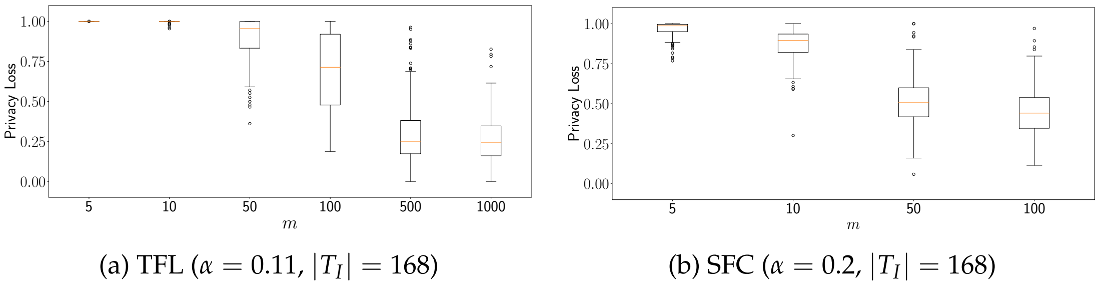
(c) $m = 50$

(d) $m = 100$

(e) $m = 500$

(f) $m = 1,000$

FIGURE 7.2: TFL, Subset of Locations prior ($\alpha = 0.11$, $|T_I| = 168$) – Adv's performance for different values of m.



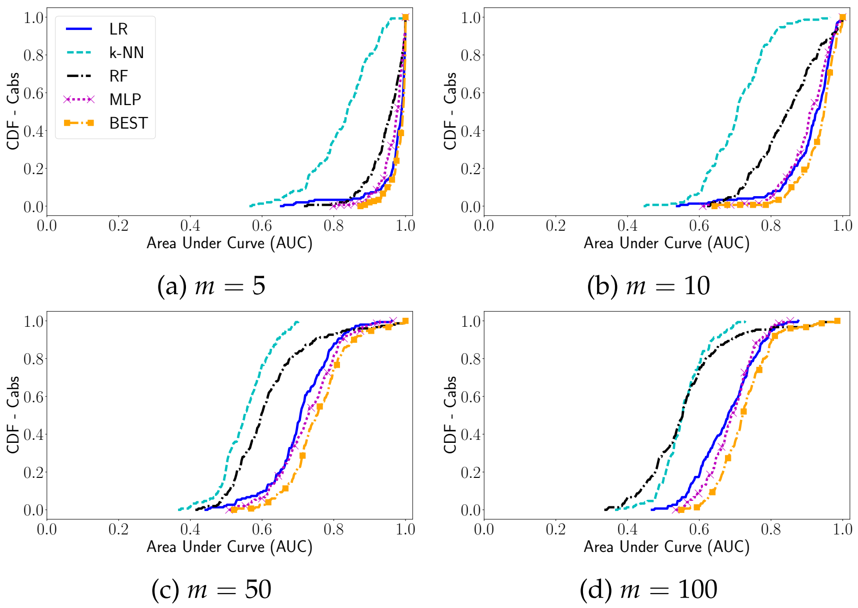
(a) TFL ($\alpha = 0.11$, $|T_I| = 168$)

(b) SFC ($\alpha = 0.2$, $|T_I| = 168$)

FIGURE 7.3: Subset of Locations prior – Privacy Loss (PL) for different values of m.

contribution to the aggregates is very significant, membership inference is very effective. As the size of the aggregation groups increases to $m = 50$ or 100, the performance only slightly decreases, with RF outperforming LR, Nearest Neighbors (k-NN), and Multi-Layer Perceptron (MLP), yielding 0.94 mean AUC for groups of 50 users, and 0.83 for 100. With larger aggregation sizes, $m = 500$ or 1,000, performance drops closer to the random guess baseline (AUC = 0.5). Nonetheless, even for groups of 1,000 users, Adv can still infer membership of 60% of the target population with an AUC score higher than 0.6.

We also measure the impact of the effectiveness of the distinguishing function on privacy using the privacy loss metric (PL, recall Eq. 7.3). More specifically, in Figure 7.3a, we report a box plot of the PL for different aggregation group sizes, when the adversary picks the best classifier for each target user (orange line in Fig. 7.2). For small groups, mean PL is very large,



(a) $m = 5$

(b) $m = 10$

(c) $m = 50$

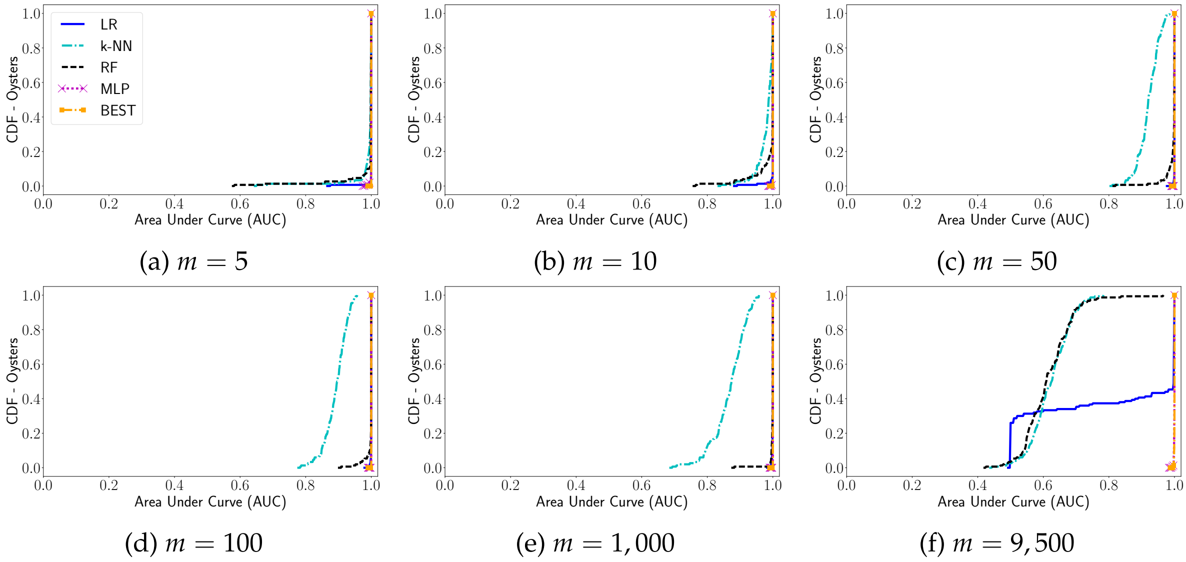
(d) $m = 100$

FIGURE 7.4: SFC, Subset of Locations prior ($\alpha = 0.2$, $|T_I| = 168$) – Adv's performance for different values of m.

e.g., 0.99 for $m = 10$, 0.89 for 50, and 0.68 for 100. Unsurprisingly, PL decreases as the group size increases, i.e., users enjoy better privacy when their data is aggregated in larger groups. Yet, even then they experience a 25% reduction of privacy with respect to a random guess ($m = 1,000$).

SFC. In Figure 7.4, we plot the classifiers' performance on the SFC dataset for groups of up to 100 users, as we are limited by the adversarial knowledge (106 cabs). As in the previous case, for small groups ($m = 5$ or 10) Adv can infer membership with high accuracy. For instance, for groups of 10 users, LR and MLP achieve mean AUC of 0.9, followed by RF (0.84) and k-NN (0.7). Again, performance decreases as group size increases: for groups of 50 cabs (resp., 100) MLP and LR yield mean AUC scores of 0.72 (resp., 0.68) and 0.7 (resp., 0.67). Nonetheless, when Adv picks the best classifier for each cab (orange line), the mean AUC score is still 0.72 even with 100 users per group.

PL over the different values of m is explored in Fig. 7.3b, using the best classifier for each target. Similar to the TFL case, the loss is very large for small groups (e.g., PL $= 0.86$ when $m = 10$), and remains significant in larger ones (e.g., PL $= 0.44$ when $m = 100$). Interestingly, for groups up to 100 users, PL is larger on TFL than on SFC data (e.g., PL $= 0.68$ on TFL vs. 0.44 on SFC, for $m = 100$), indicating that membership inference is easier on sparse data.



(a) $m = 5$

(b) $m = 10$

(c) $m = 50$

(d) $m = 100$

(e) $m = 1,000$

(f) $m = 9,500$

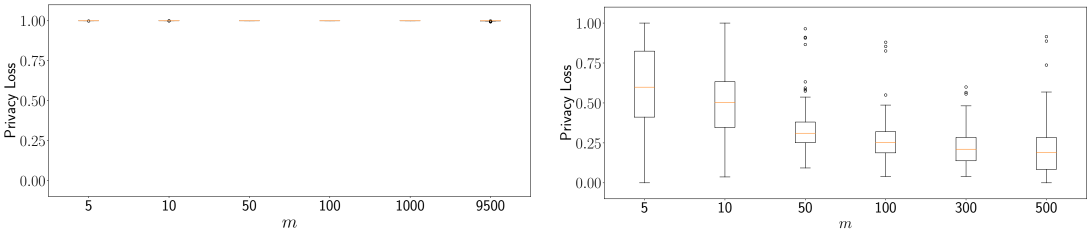
FIGURE 7.5: TFL, Same Groups as Released prior ($\beta = 150$, $|T_I| = 168$) – Adv's performance for different values of m.

7.3.2 Participation in Past Groups

Next, we simulate the setting where Adv's prior knowledge consists of aggregates of groups released in a past observation period, labeled as including data from the target user or not. As discussed in Section 7.1.3, we consider two variants: when Adv's prior knowledge is built on either (a) the *same* groups as or (b) *different* groups than those used to compute the inference period aggregates.

Same Groups as Released

In this setting, we generate D by computing the aggregates of $\beta = 150$ randomly sampled unique user groups – 75 that include and 75 that exclude the target – and set the corresponding label of participation. We split D *over time* to obtain the training and testing sets. As observation period, i.e., where Adv builds her prior knowledge, we consider the first *3 weeks* for TFL (i.e., $|T_O| = 3 \cdot 168 = 504$ hourly time slots) and the first *2 weeks* for SFC ($|T_O| = 336$). In both cases, the inference period is the *last week* of data, thus $|T_I| = 168$ hourly time slots, yielding a 75%-25% train/test split for TFL, and a 67%-33% one for SFC. Finally, we train the classifiers with features extracted from the aggregates of *each week* in the training set, and test them on those extracted from the aggregates of each group in the test set.



(a) TFL ($\beta = 150$, $|T_I| = 168$)

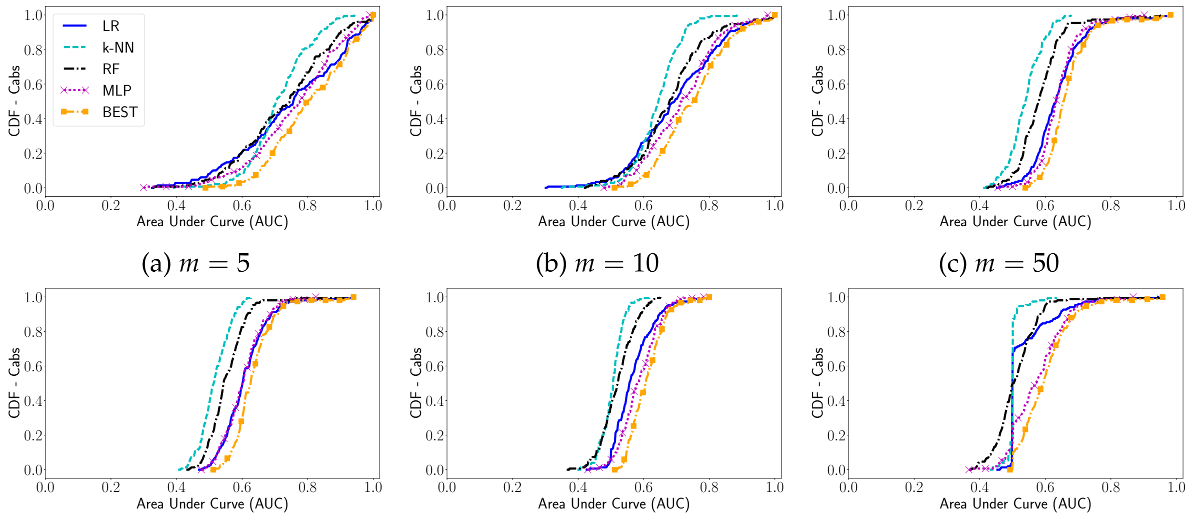
(b) SFC ($\beta = 150$, $|T_I| = 168$)

FIGURE 7.6: Same Groups as Released prior – Privacy Loss (PL) for different values of m.

TFL. Figure 7.5 shows the classifiers' performance for different aggregation group sizes (m). In this experiment, there is no limitation from the prior, thus, we can consider groups as large as the dataset. We observe that for group sizes up to 100 (Figs. 7.5a–7.5d), membership inference is very accurate (all classifiers yield mean AUC scores higher than 0.9). Interestingly, as the groups grow to 1,000 commuters (Fig. 7.5e), LR, RF, and MLP, still yield very high AUC scores (0.99 on average), while that of k-NN slightly decreases (0.86). For groups of 9,500 commuters (Fig. 7.5f), MLP clearly outperforms the other classifiers yielding an AUC score of 0.99 compared to 0.81 for LR, 0.62 for k-NN and 0.61 for RF. Overall, this experiment indicates that when mobility patterns are regular, as those of commuters, an adversary with prior knowledge about specific groups can successfully infer membership in the future if groups are maintained, even if they are large.

Figure 7.6a reports the privacy loss (PL) when the adversary picks the best classifier for each user. We see that, independently of the group size, commuters lose a huge amount of privacy when their data is aggregated in groups for which the adversary has prior knowledge. The results reinforce the previous intuition: the effect of regularity on aggregates is very strong, and makes commuters very susceptible to membership inference attacks.

SFC. Figure 7.7 illustrates the performance of the classifiers for variable aggregation group size on the SFC dataset. Recall that this is smaller than TFL, as it only contains the location data of 534 cabs. We observe that the lack of regularity in cabs' movements has a great impact on the ability of an adversary to infer membership, even when the groups are maintained over time. For small groups ($m = 5$ or 10), the classifiers' AUC ranges between 0.76 and 0.64, as opposed to 0.9 or more in TFL, with MLP now yielding the best results. As groups become larger (Figs. 7.7c–7.7e), irregularity has a bigger effect and, unexpectedly, performance drops



(a) $m = 5$ (b) $m = 10$ (c) $m = 50$

(d) $m = 100$ (e) $m = 300$ (f) $m = 500$

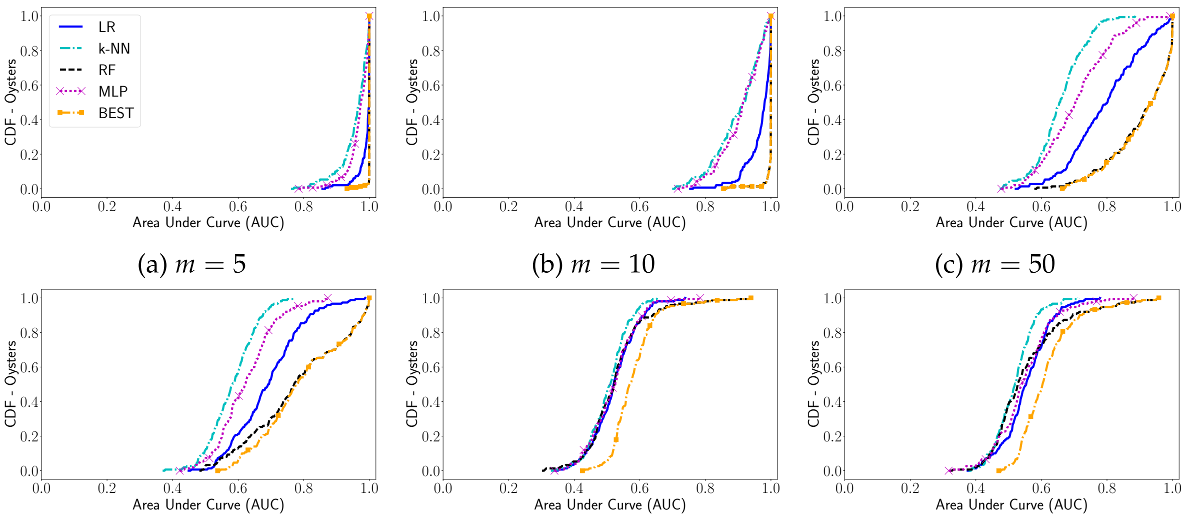
FIGURE 7.7: SFC, Same Groups as Released prior ($\beta = 150$, $|T_I| = 168$) – Adv's performance for different values of m.

further. Already for $m = 100$, RF and k-NN perform similar to the random guess baseline, and LR's AUC drops to 0.52 when group size reaches $m = 500$. MLP, however, is still somewhat better than random (0.57 mean AUC). Overall, if the adversary picks the best classifier for each cab (orange line), she can infer membership for half the cabs with AUC score larger than 0.6.

In terms of PL, Figure 7.6b shows that cabs lose privacy when they are aggregated in small groups. However, since cabs, as well as the groups they are aggregated in, are not as regular as TFL commuters, the loss drops drastically with larger groups (e.g., mean PL is 0.2 for groups of 500 cabs). In other words, irregularity makes inferring membership harder for the adversary. However, even though on average PL decreases, we observe that, for $m = 500$, some instances of our experiment exhibit larger privacy loss than for $m = 300$. This stems from the small size of the cab population. As there are only 534 cabs, when grouping them in batches of 500 elements, there inevitably is a big overlap across groups, which effectively creates a somewhat artificial regularity that increases the performance of the attack.

Different Groups than Released

In this setting, for each target user, we design a balanced experiment by generating a dataset D with the aggregates of 400 unique randomly sampled groups – half including the target and



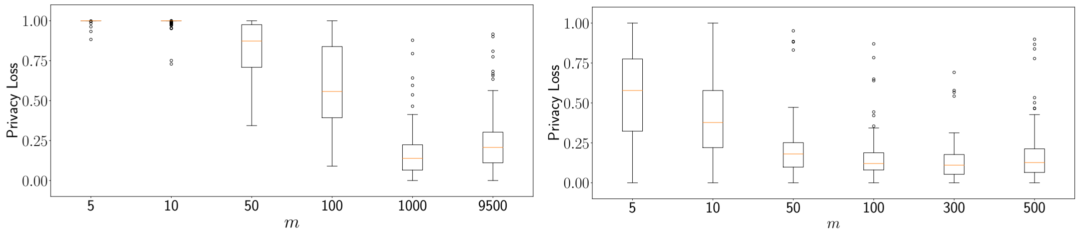
(a) $m = 5$ (b) $m = 10$ (c) $m = 50$

(d) $m = 100$ (e) $m = 1,000$ (f) $m = 9,500$

FIGURE 7.8: TFL, Different Groups than Released prior (β=300, $|T_I|$=168) – Adv's performance for different values of m.

half not – and set the corresponding participation label (in or out). Once again, the experiment size is chosen to provide enough data for our classifiers to learn patterns, while keeping the computation time reasonable on commodity hardware. To simulate the difference in groups between *observation* and *inference* period, we first perform a 75%-25% stratified random split of D, whereby we keep 300 groups for training and 100 for testing. Then, for TFL, we define the observation period to be the first *3 weeks* of data (i.e., $|T_O| = 504$) while for SFC the first *2 weeks* ($|T_O| = 336$) and in both cases, the inference period is the *last week* (i.e., $|T_I| = 168$). We then split the training and testing sets according to time: from the training set, we keep only the aggregates of the observation period, while, from the testing set, only those from the inference period (i.e., overall, we perform a 75%-25% split for TFL and 67%-33% one for SFC). That is, we let the adversary obtain knowledge for 300 user groups (i.e., $\beta = 300$), half including her target, whose aggregates are generated during the observation period. Finally, we train the classifiers on the features extracted from the aggregates of the groups in the *training* set for each week of the *observation* period, and test them against those extracted from the groups in the *testing* set (i.e., during the *inference* period).

TFL. Figure 7.8 illustrates the classifiers' performance for different aggregation group sizes (up to 9,500 commuters). Once again, for small groups ($m = 5$ or 10) membership can be easily inferred (AUC > 0.89 for all classifiers). As m increases, we first observe a small drop



(a) TFL ($\beta = 300$, $|T_I| = 168$) (b) SFC ($\beta = 300$, $|T_I| = 168$)

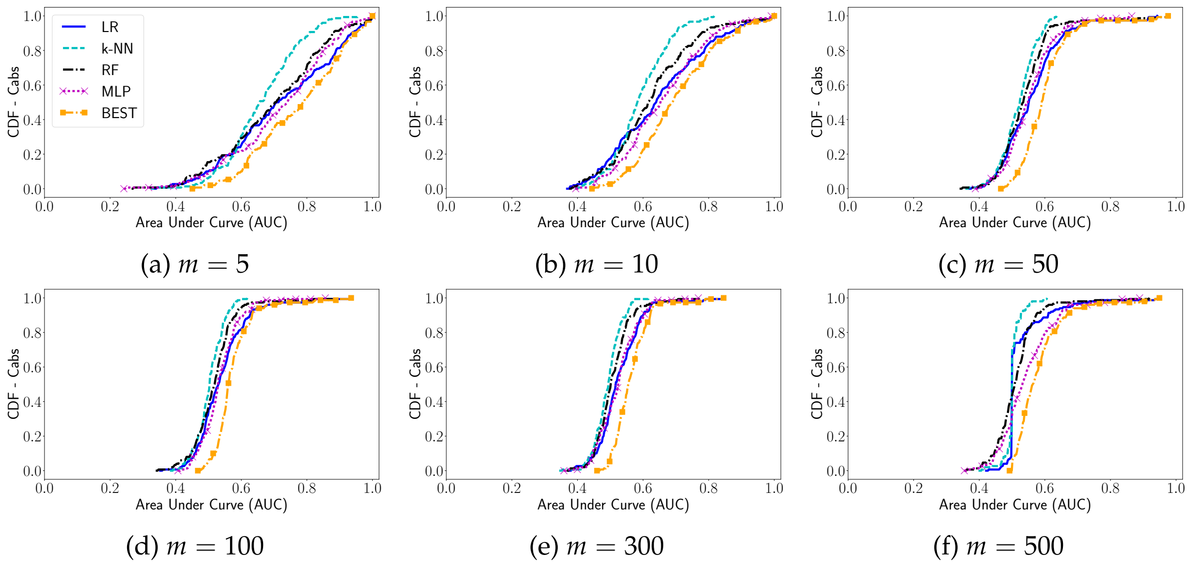
FIGURE 7.9: Different Groups than Released prior – Privacy Loss (PL) for different values of m.

in the adversarial performance, with RF achieving mean AUC of 0.89 and 0.78 for groups of 50 and 100 commuters, resp. This indicates that regularity still helps membership inference in small groups even when these groups change. However, when m reaches 1,000 all the classifiers perform, on average, similar to the baseline indicating that the effect of regularity dilutes. Interestingly, for $m = 9,500$, we note a small increase in the classifiers' AUC scores due to the big user overlap across training and testing groups, i.e., the *different groups* prior becomes more similar to the *same groups* one.

This effect can also be observed in terms of privacy loss (PL, Fig. 7.9a). Membership inference is quite effective for groups of size up to 100, where commuters suffer a privacy loss of at least 0.59. However, when the data of more commuters is aggregated, mean PL decreases to 0.17 for groups of 1,000, and it slightly increases to 0.22 when $m = 9,500$. Overall, we note that the privacy loss is smaller in this setting, however, this is not surprising, since this is a weaker adversarial setting than the previous ones.

SFC. Similar to the experiment with the same groups prior, we observe in Figure 7.10 that the classifiers perform worse for SFC than for TFL, due to the lack of regularity. Already for small groups ($m = 5$) the mean AUC drops to 0.71 for the best classifiers, LR and MLP. With larger groups, the performance is significantly lower, and all classifiers converge towards the random guess baseline. When $m = 500$, MLP and LR yield slightly better results and membership inference can be achieved with AUC larger than 0.6 for only a small percentage of cabs (about 20%).

From Fig. 7.9b, we see that, due to the weaker prior, PL values are smaller across the board compared to the previous setting. Overall, PL decreases with increasing aggregation group size, ranging from mean PL of 0.54 with $m = 5$ to 0.12 for $m = 300$. Similar to the



(a) $m = 5$ (b) $m = 10$ (c) $m = 50$

(d) $m = 100$ (e) $m = 300$ (f) $m = 500$

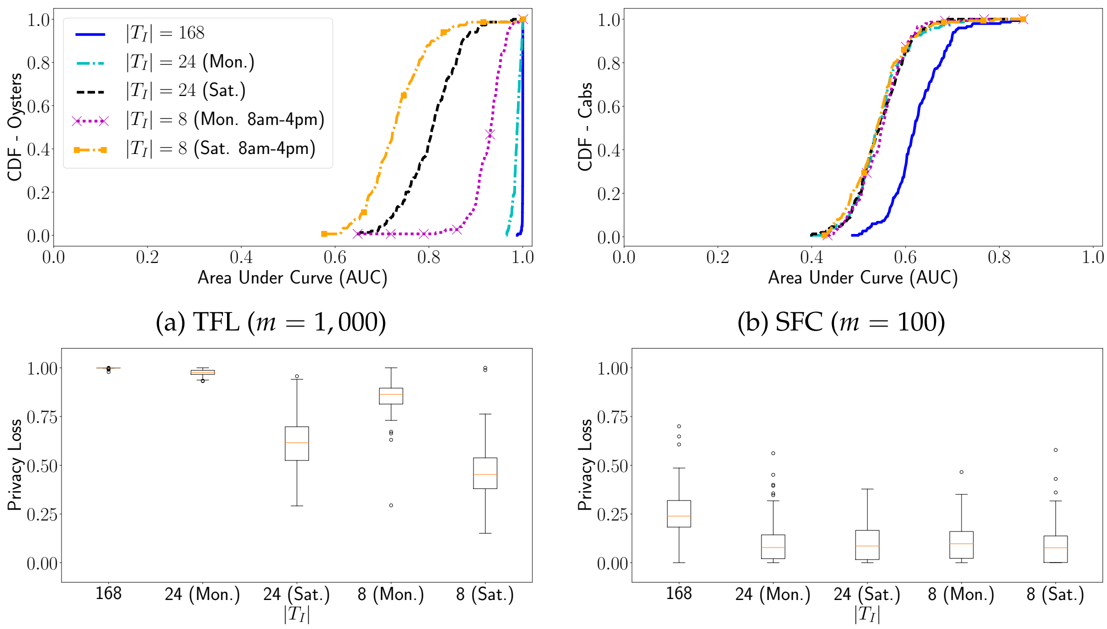
FIGURE 7.10: SFC, Different Groups than Released prior (β=300, $|T_I|$=168) – Adv's performance for different values of m.

TFL case, we observe a small increase for groups of 500 cabs. The reason is the same, i.e., the user overlaps between training and testing groups slightly improve the effectiveness of the membership inference attack.

7.3.3 Length of Inference Period

In our previous experiments, we studied the effect of the aggregation group size (m) on the success of membership inference, for various types of adversarial prior knowledge. In this section, we examine the effect of the inference period *length*, i.e., $|T_I|$. We consider lengths of 1 week (168 hourly time slots), 1 day (24 time slots), and 8 hours (8 time slots). In particular, for the last two, we also consider working vs. weekend days to account for the differences in mobility behavior.

We report experiments in the setting where Adv has prior knowledge about the groups that are released by Ch – i.e., the Same Groups as Released prior discussed in Section 7.1.3 – and fix the group size to 1,000 commuters for TFL and to 100 cabs for SFC. For each target user, we create a dataset of $\beta = 150$ randomly sampled unique groups, half of which include the user and half of which do not, and split their aggregates in training and testing sets according to time following a 75%-25% split for TFL, and a 67%-33% one for SFC. We choose RF as the classifier for TFL, and MLP for SFC, since they yield the best AUC scores in this setting,



(a) TFL ($m = 1,000$)

(b) SFC ($m = 100$)

(c) TFL ($m = 1,000$)

(d) SFC ($m = 100$)

FIGURE 7.11: Same Groups as Released prior ($\beta = 150$) – Adv's performance for variable inference period length ($|T_I|$), on (a) TFL and (b) SFC, and Privacy Loss (PL) on (c) TFL and (d) SFC.

as shown in Figs. 7.5e and 7.7d. For each $|T_I| \in \{8, 24, 168\}$, we train the classifiers on aggregates of that length, for each week in the training set (observation period), and evaluate them against the corresponding aggregates in the test set (inference period).

Figure 7.11a reports the results on the TFL dataset: as the number of points in the inference period $|T_I|$ decreases, the adversarial performance degrades as there is less information about mobility patterns to be exploited. Also, there is indeed a difference between working days and weekends. The mean AUC is 0.97 when training and testing on a Monday, and 0.8 on a Saturday. This seems to be due to regularity, as commuters' regular patterns during the week make them more susceptible to membership inference than sporadic or leisure activities over weekends. This is confirmed by the classifier's performance for $|T_I| = 8$, as we obtain much better results when the inference period is set to Monday 8am–4pm (AUC $= 0.91$) than on Saturday during the same timeframe (AUC $= 0.72$).

Once again, the lack of regularity affects negatively Adv's performance when attacking the SFC dataset (Fig. 7.11b). As for the length of the inference period, our results confirm that the inference task becomes harder with fewer points in time: the mean AUC drops from 0.62

to 0.54 when $|T_I|$ decreases from 1 week to 1 day. However, as cabs are overall not regular (their movements are mandated by client demand), we do not observe significant difference between working days and weekends, nor when considering full days vs. 8 hour slots.

The privacy loss (PL) exhibits similar trends. For TFL (see Fig. 7.11c), the highest loss is observed when more points are available (0.98 on average when $|T_I|$ is 1 week long), while the loss is reduced as the length of the inference period decreases and the adversary has less information. Also, we see how regularity in working days results in better membership inference attacks than during weekends, i.e., mean PL is 0.96 and 0.85 on Mondays vs. 0.61 and 0.46 on Saturdays for $|T_I|$ set at 24 and 8 hours, respectively. Finally, Figure 7.11d highlights smaller PL for the SFC cabs, for all period lengths, with a maximum mean PL of 0.25 when $|T_I|$ is 1 week, down to 0.1 and 0.09 for 1 day and 8 hours, respectively. There is no significant difference between Mondays and Saturdays, confirming that regularity has a strong influence on the problem.

7.3.4 Discussion

Overall, our evaluation showcases the effectiveness of modeling membership inference attacks on aggregate location time-series as a classification task, vis-à-vis different datasets and priors. We show that an adversary can build a machine learning model by extracting features from known aggregate location time-series and use it to guess whether a target user has contributed her data to a set of previously unseen aggregates. Our results evidence that the risks stemming from such attacks are significant, with the actual level of privacy leakage depending on the adversary's prior knowledge, the characteristics of the data, as well as the group size and the timeframe on which aggregation is performed.

We find that, up to certain aggregation group sizes, membership inference is very successful when the adversary knows the actual locations of a subset of users (including her target), or when she knows past aggregates for the same groups on which she tries to perform the inference. In the least restrictive setting, where the past groups known to the adversary are different than those whose location time-series are released, privacy leakage is relatively small, but still non-negligible. Moreover, the characteristics of the data used for aggregation also influence the adversarial performance: in general, privacy leakage on the dataset containing mobility of commuters (TFL) is larger than on the one including cab traces (SFC). This highlights that regularity in users' movements, as well as sparseness of the location signal, significantly ease the membership inference task.

Finally, the number of users that contribute to aggregation also has a profound effect on the adversarial performance. Unsurprisingly, membership inference is very successful when aggregation is performed over small groups, while users generally enjoy more privacy in larger groups. A notable exception is the TFL case, where due to the regularity of commuters, membership inference attacks are still very effective even for large groups. Also, the length, as well as the time semantics, of the inference period play a very important role. Inference is easier if the aggregates of longer periods are released (i.e., more information is available to extract patterns), and at times when mobility patterns are likely to be more regular (e.g., workdays or mornings).

7.4 Evaluation of Defenses

In this section, we evaluate the effectiveness of available defense mechanisms that can be employed to prevent privacy leakage from membership inference attacks on aggregate location time-series. To this end, we focus on the established framework to define private functions that are free from inferences, namely, Differential Privacy (DP, see Section 2.2). The application of a differentially private mechanism to a dataset ensures that only a bounded amount of information is disclosed upon its release. Such mechanisms can mitigate membership inference attacks, as DP's indistinguishability-based definition guarantees that the outcome of any computation on a dataset is insensitive to the inclusion of any record (or user) in the dataset.

7.4.1 Experiment Design

Intuition. Our evaluation of membership inference on raw aggregate location time-series (Section 7.3) showed that releasing such data poses a significant privacy threat for users who contribute to the aggregation, and more so in settings where the latter is performed over small groups. In this section, we present experiments aiming to evaluate the effectiveness of differentially private mechanisms in defending against membership inference. Note that we opt to evaluate them over large user groups, since we expect (and have also verified experimentally) that, for small groups, the utility loss incurred by DP-based mechanisms is prohibitively high. This is because the *sensitivity* of the location aggregation function, which directly affects the amount of noise to be employed, does not depend on the group size (m). As such, the aggregate time-series of groups with few users, which naturally have small counts, are affected more by the noise required by the DP-based mechanism.

As the large group size gives the defense mechanism an *advantage* in terms of utility, and since DP provides protection against arbitrary risks [Dwo08], we set to run our experiments considering a worst-case adversary that obtains *perfect* prior knowledge for the users, i.e., she knows the inference period aggregates of the groups that are released by the challenger as well as the target user's membership in these groups. With this knowledge, Adv is able to train an accurate machine learning classifier that, upon release of raw statistics, always guesses correctly the target user's membership, i.e., achieves an AUC score of 1. In other words, we evaluate the privacy and utility trade-off of differentially private mechanisms considering the best setting for utility and the worst one for privacy.

Experiment. We slightly modify the DG game described in Figure 7.1, such that Ch applies a differentially private mechanism on the aggregates, before sending her *challenge* to Adv. We evaluate the gain in privacy offered by the mechanisms on two cases, depending on whether Adv's classifier (which, once again, instantiates the distinguishing function) is trained on (a) the *raw* aggregates of the groups to be released; or (b) *noisy* aggregates of the groups to be released using the defense mechanism under examination.

In both cases, testing is done on the aggregates of the released groups, perturbed with the defense mechanism (using *fresh* noise). The first scenario represents a *passive* adversary that attempts to infer user membership on the noisy aggregates, exploiting only the raw aggregate information from her prior knowledge. The second one, represents a strategic *active* adversary that tries to mimic the behavior of the defender during training, knowing the parameters of the defense mechanism (i.e., the differential privacy parameter ϵ and the sensitivity of the aggregation function), but not knowing the concrete noise values used in the defender's perturbation. We follow the same procedure as in Section 7.2, i.e., we extract features from the aggregate location time-series of the user groups, and use RFE to reduce the number of features to the number of samples.

Settings. We run membership inference attacks against the same 150 users sampled from each dataset (see Section 7.2). We take as observation and inference period the first week in each dataset, i.e., $|T_O| = |T_I| = 168$. Aiming to examine a favorable setting for the utility of DP-based mechanisms, we construct large user groups: we set $m = 9,500$ for TFL, and $m = 500$ for SFC. Then, for each user victim, we generate the dataset D by randomly sampling 200 and 400 elements for TFL and SFC, respectively, half including her and half not. We pick a different number of groups for practical reasons as the TFL dataset has six times more ROIs than the SFC one, and this makes feature extraction significantly more expensive. Regarding the classifier, we employ MLP, which overall, performed well in the previous experiments.

We evaluate the perturbation mechanisms described in Section 2.2.2, namely, the generic Laplace (LPM) and Gaussian mechanisms (GSM), as well as the tailored to time-series settings Fourier perturbation algorithm (FPA) and its extended version with Gaussian noise (EFPAG). As a baseline for our evaluation, we employ the Simple Counting Mechanism (SCM) that guarantees event-level privacy. To configure the perturbation mechanisms, we calculate the sensitivity of the users in each dataset (i.e., the maximum number of events reported by an oyster or cab during the inference week), obtaining, respectively, 207 for TFL and 2,685 for SFC. We consider ϵ values of DP in the range $\{0.01, 0.1, 1.0, 10.0\}$, and set $\delta = 0.1$ for GSM and EFPAG. For FPA, we empirically find the best value for κ in terms of utility, setting $\kappa = 25$ for TFL, and $\kappa = 20$ for SFC.

Metrics. Our evaluation uses the following privacy and utility metrics to capture the amount of privacy gained compared to a setting where the DG game is played on raw aggregates, and the utility lost due to the noise addition.

Privacy Gain (PG): We define PG as the relative decrease in a classifier's AUC score when tested on perturbed aggregates ($\text{AUC}_{A'}$) versus raw aggregates (AUC_A):

$$\text{PG} = \begin{cases} \frac{\text{AUC}_A - \text{AUC}_{A'}}{\text{AUC}_A - 0.5} & \text{if } \text{AUC}_A > \text{AUC}_{A'} \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{7.4}$$

PG is a value between 0 and 1, which captures the decrease in adversarial performance, i.e.,

ϵ	**0.01**	**0.1**	**1.0**	**10.0**
LPM	3056.1	812.6	81.7	8.2
GSM	753.2	75.8	7.4	0.75
FPA	67.2	6.1	0.7	0.03
EFPAG	36.8	3.6	0.4	0.03
SCM	38.5	3.7	0.3	0.002

TABLE 7.2: MRE (utility loss) of aggregate location time-series with various differentially private mechanisms and ϵ parameter on the TFL dataset.

ϵ	**0.01**	**0.1**	**1.0**	**10.0**
LPM	131.9	129.3	114.4	41.9
GSM	129.6	94.7	14.1	1.4
FPA	85.9	11.3	1.1	0.11
EFPAG	57.9	6.1	0.6	0.04
SCM	24.7	2.5	0.2	0.001

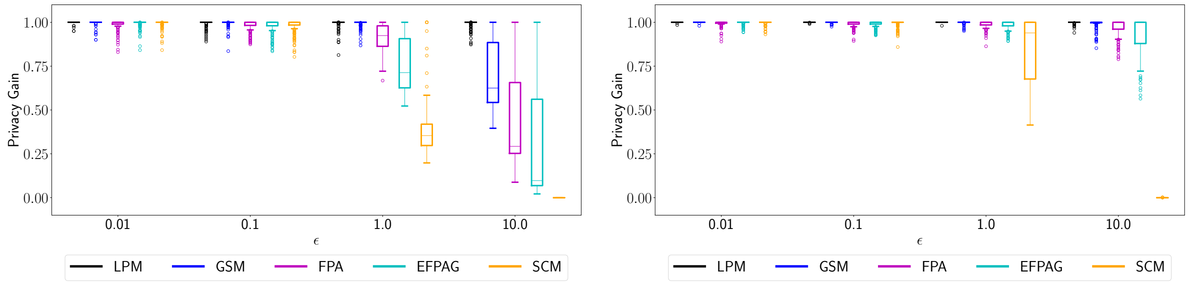
TABLE 7.3: MRE (utility loss) of aggregate location time-series with various differentially private mechanisms and ϵ parameter on the SFC dataset.

how much the adversary's inference power deteriorates towards the random guess baseline, when a defense mechanism is implemented.

Mean Relative Error (MRE): We measure the utility loss as a result of using a DP-based defense mechanism by means of the standard MRE metric (Section 2.3.1), computed between the raw aggregate location time-series and their perturbed versions. We compute the MRE over the aggregate time-series of each ROI in our datasets and report the mean value.

7.4.2 Results

Utility loss. We first report the utility loss measured as per the Mean Relative Error (MRE) of the TFL and SFC perturbed aggregates, for each mechanism and different values of ϵ, in Tables 7.2 and 7.3. Naturally, we observe that as ϵ increases, i.e., as the DP mechanisms provide lower privacy guarantees, the utility loss (MRE) decreases. Among the defense mechanisms, we note that LPM incurs the highest MRE, with the noisy aggregate values being 8 (resp., 41) times less accurate than the raw aggregates on TFL (resp., SFC) data, in the most relaxed privacy setting ($\epsilon = 10.0$). Accordingly, with GSM, utility does not improve much as the perturbed aggregates still have very large error, e.g., 7.4 (14.1) for TFL (SFC) with ϵ set to 1.0. On the other hand, the specialized for time-series mechanisms, namely, FPA and EFPAG, achieve better results, e.g., MRE is under 1.1 for values of $\epsilon \geq 1.0$. Finally, we observe that



(a) TFL ($m = 9,500$, $|T_I| = 168$)

(b) SFC ($m = 500$, $|T_I| = 168$)

FIGURE 7.12: Privacy Gain (PG) achieved by differentially private mechanisms with different values of ϵ, against a MLP classifier trained on *raw* aggregates and tested on *noisy* aggregates.

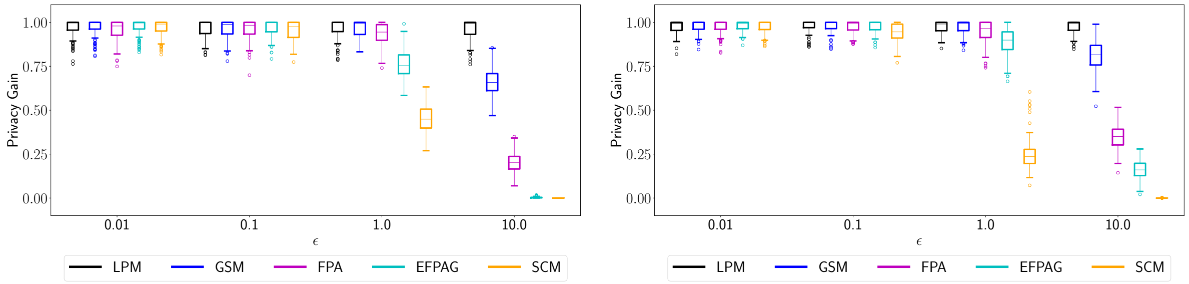
SCM achieves the best utility, but it provides poor privacy protection against membership inference attacks, as we will show below.

Privacy gain. We now evaluate the privacy gain (PG) provided by the different DP-based mechanisms, distinguishing between the two settings introduced in Section 7.4.1.

Train on raw and test on noisy aggregates: Figure 7.12 plots the PG achieved by various mechanisms against a MLP classifier trained on raw aggregates and tested on perturbed ones. For TFL (Fig. 7.12a), we observe that for low ϵ values (up to 0.1) all mechanisms provide excellent privacy protection, achieving a gain in privacy close to 1. However, this protection comes with poor utility, as shown in Table 7.2. As ϵ increases to 1.0, LPM and GSM still provide good protection, while we observe a small drop in PG for the mechanisms that achieve MRE < 1.0. In particular, FPA now yields a mean PG of 0.9, while EFPAG and SCM 0.75 and 0.38, resp. When $\epsilon = 10.0$ and the utility loss of FPA and EFPAG is very small, the decrease in PG is bigger (0.45 and 0.3, resp.), as expected.

With SFC data (Fig. 7.12b), we find that PG for all the mechanisms stays high for values of ϵ up to 1.0. This is reasonable, since the sensitivity, and thus, the magnitude of noise required, is much larger on SFC compared to TFL (2,685 vs. 207). However, as seen from Table 7.3, utility is quite poor in these settings. With $\epsilon = 10.0$, mean PG is almost 1 for LPM and GSM, i.e., users are very well protected against membership inference attacks. Meanwhile, PG slightly drops for FPA and EFPAG (0.96 and 0.92 on average) while their utility loss is smaller. Unsurprisingly, SCM achieves negligible privacy gain in this setting.

Train on noisy and test on noisy aggregates: Figure 7.13 reports the PG results when the



(a) TFL ($m = 9,500$, $|T_I| = 168$) (b) SFC ($m = 500$, $|T_I| = 168$)

FIGURE 7.13: Privacy Gain (PG) achieved by differentially private mechanisms with different values of ϵ, against a MLP classifier trained and tested on *noisy* aggregates.

MLP classifier is trained on noisy aggregates. Interestingly, the protection of the mechanisms decreases much faster for increasing values of ϵ. For TFL (Fig. 7.13a), we observe that for values of $\epsilon \leq 1.0$, the PG decreases only slightly compared to the previous setting, where training was done on raw aggregates (Fig. 7.12a). That is, the DP-based mechanisms still provide very good protection against membership inference. However, when $\epsilon = 10.0$, we notice a notable decrease in PG, with FPA and EFPAG. More precisely, FPA now achieves 0.2 mean PG (vs. 0.45 in the previous setting) and EFPAG provides negligible protection against membership inference (compared to 0.3 in Fig. 7.12a, $\epsilon = 10.0$).

Similarly, with SFC (Fig. 7.13b), mean PG remains high for $\epsilon \leq 1.0$ for all mechanisms, except for SCM, as expected. For $\epsilon = 10.0$, there is a significant decline in PG with GSM, FPA, and EFPAG. In particular, GSM now yields 0.8 mean PG, while FPA and EFPAG 0.32 and 0.15, respectively. This corresponds to a significant drop in privacy protection (20%, 66%, and 83%, for GSM, FPA, and EFPAG, resp.) compared to the setting where training was done on raw aggregates (cf. Fig. 7.12b).

7.4.3 Discussion

The experiments presented in this section demonstrate how our methodology can be used to evaluate the performance of differentially private mechanisms against membership inference on aggregate location time-series, both in terms of privacy and utility. Considering an advantageous setting for utility, but worst-case for privacy, we find that differentially private mechanisms are effective at preventing inferences, but with some important caveats.

In particular, our results show that a *passive* adversary who trains a classifier on raw aggregate location data is not very successful at inferring membership on noisy aggregates. However, when we consider a *strategic* adversary that mimics the behavior of the defender, and trains a classifier on noisy aggregates, we find that the actual privacy gain offered from the DP-based mechanisms is significantly reduced, and also decreases much faster with increasing ϵ values. This should draw the attention of the research community as advances in deep learning, might lead to novel attacks against defenses based on perturbation (e.g., by achieving noise filtering).

Among the defense mechanisms considered, we observe that the straightforward application of LPM and GSM yields very poor utility. This is not surprising, as previous work highlights the difficulty of releasing private statistics under continual observation [Dwo+10; CSS11; Kel+14]. Mechanisms specifically proposed for time-series settings (i.e., FPA and EFPAG) yield much better utility, at the cost of reduced privacy. This shows that achieving an optimal trade-off between privacy and utility in the settings we consider is still a challenging task.

Finally, our analysis also shows how dataset characteristics affect the performance of differentially private mechanisms too. Specifically, the privacy gain on a sparser dataset (TFL) decreases faster with growing ϵ, compared to a denser one (SFC). This is not surprising, taking into account the scale difference between the sensitivity of the aggregation in each case (recall that the sensitivity is 207 on the TFL dataset and 2,685 on the SFC one).

Chapter 8

Understanding Membership Inference On Aggregate Location Data

In Chapter 7, we showed that given a piece of aggregate location data, a knowledgeable adversary can launch a membership inference attack, i.e., she can infer whether or not a target user contributed her data to those aggregates. In particular, we formalized the problem as a distinguishability game and used supervised machine learning to instantiate the adversarial task. While we focused on the feasibility of the attack as well as the effectiveness of differential privacy to thwart it, we did not investigate what spatio-temporal factors contribute to the attack's success and in what context, nor which users are most vulnerable to it. Such lack of understanding not only leaves research questions around membership inference on aggregate locations unanswered but ultimately hampers the design of defenses providing acceptable privacy/utility tradeoffs.

In this chapter, we set to gain a deeper understanding of membership inference in the setting of aggregate location data. Using the real-world mobility datasets introduced in Chapter 4, we perform membership inference attacks and study the reasons behind its success. We follow a dimensionality reduction approach, based on Principal Component Analysis (PCA), which allows us to gain insights about locations and times that are important for the inference, by examining the correlation coefficients in the components. Furthermore, we investigate which users are more affected by the attack than others, studying the feature importance of a classifier trained on their mobility characteristics (e.g., number of events, number of unique locations visited, etc.) and identify those factors that help the inference. Subsequently, we explore whether defenses commonly used in the location privacy literature, such as generalization and hiding, can be used to mitigate the attack, building on the insights obtained from our analysis to inform their configuration. For these techniques, as well as perturbation ones (guaranteeing differential privacy), we evaluate the privacy/utility tradeoffs that they achieve with respect to membership inference and various analytics tasks on aggregate location data including traffic forecasting, anomaly detection, hotspot discovery, and location labeling.

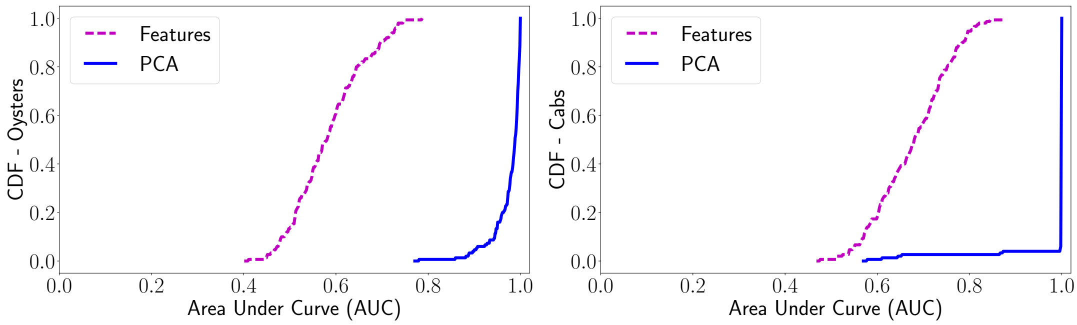
8.1 Understanding Membership Inference on Aggregate Locations

We first set to understand what makes the presence of a user's location data points inferable from aggregates, and how this might vary depending on the adversarial prior knowledge. In this section, we present our experimental setup and provide a summary of insights gained for the three different types of adversarial prior knowledge which were discussed in Section 7.1.3.

8.1.1 Experiment Setup

We use the notation introduced in Chapter 7 (see Table 7.1). Moreover, following the methodology and setup proposed in Section 7.2, we split the users of the TFL and SFC datasets into three mobility groups (highly, somewhat, and barely, mobile) and run membership inference attacks against 150 users, 50 from each group (sampled at random). To target a user, we create a *balanced* dataset containing aggregate location time-series that include and exclude her location data to train a classifier which is used as a distinguisher by the adversary (i.e., during testing).

PCA optimization. The classifiers trained for the evaluations of Chapter 7 use as features simple statistics calculated on the time-series of each location (mean, median, maximum, minimum, variance, standard deviation, and sum). However, upon performing a preliminary feature analysis, we find that in both TFL and SFC datasets, the *variance* of the location counts over time is among the most important features. Therefore, in our analysis, we skip



(a) TFL ($\alpha = 0.11$, $m = 1,000$, $|T_I| = 168$) (b) SFC ($\alpha = 0.2$, $m = 100$, $|T_I| = 168$)

FIGURE 8.1: Membership inference performance with the Subset of Locations prior: Features vs. PCA.

the extraction of features, since most of them do not seem to add much value to the classifier. Instead, we use Principal Components Analysis (PCA, see Section 2.1.3) to reduce the dimensionality of the problem and extract the valuable information. We then feed the resulting components to a Logistic Regression (LR) classifier (we choose LR as it yields the best performance).

This optimization not only helps us understand the effects of mobility patterns on membership inference, e.g., by investigating the correlation coefficients in the principal components, but also boosts the attack's performance. To illustrate this improvement, we plot, in Figure 8.1, the CDF of the classifier's AUC scores, computed over the 150 target users, for both the feature extraction approach (whose results were discussed in Section 7.2) and the dimensionality reduction one when using the Subset of Locations prior. The increase on the classifier's mean AUC score amounts to 65% for TFL, and 46% for the SFC dataset. We observe the same trend, though somewhat less prominently, for the other priors: with the Same Groups as Released one, the improvement is 22% for TFL and 16% for SFC; with the Different Groups than Released, the adversarial performance increases by 26% and 17%, respectively.

Studied characteristics. We first examine the correlation coefficients within the components of PCA. This provides us with insights about which spatial and temporal points contribute to the inference. Then, we train a machine learning classifier on the *mobility characteristics* of the victims that are most and least prone to the attack, and we investigate its feature importance in order to identify which factors make the attack more powerful. Specifically, we compute the following statistics of the users' trajectories: total events (i.e., (location, time) tuples), unique locations visited, active time slots, mean locations per time slot, mean events and active time slots during week days and weekends, spatial and temporal entropy, and unicity. The latter captures how unique is a user's $u \in U$ travel pattern and is computed as:

$$\text{unicity}_u = \frac{\sum_{t \in T} \mathbb{1}^t(u)}{|T|} \tag{8.1}$$

where $\mathbb{1}^t(u)$ indicates whether the ordered sequence of locations visited by user u at time t is unique or not.

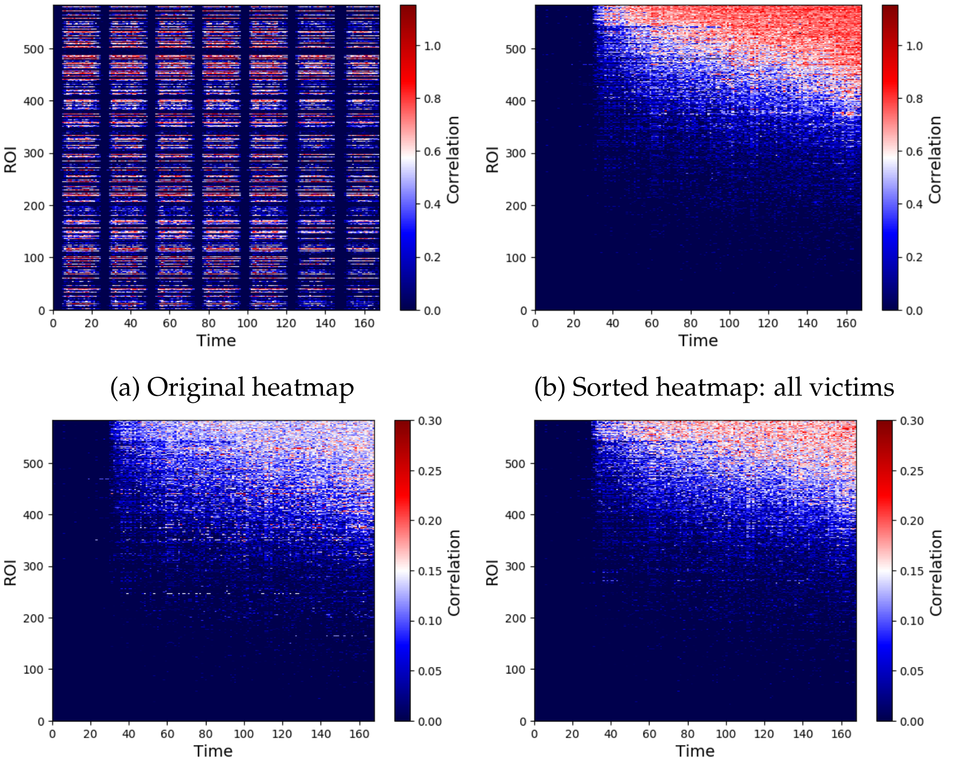
We also aim to understand whether or not intuitive hypotheses about the success of membership inference on aggregate locations hold. For instance, does the volume of data a user contributes to the aggregation affect her susceptibility to the attack? Do users' movements in less popular locations and/or times give membership away? Do very unique or very regular mobility patterns increase the attack's performance?

8.1.2 Subset of Locations

We first study the case where Adv knows the actual locations for a subset of users during the inference period, including her target.

Parameters. We set the percentage of users for which locations are known as $\alpha = 0.11$ and $\alpha = 0.2$ for, resp., TFL and SFC, and consider the maximum user group size that the adversary can attack: $m = 1,000$ for TFL and $m = 100$ for SFC. In both cases, we set the first week of data to be the observation and inference period, i.e., $|T_I| = 168$ hourly time slots, and we create datasets of 2K samples to train and test the classifier (following a $80\% - 20\%$ train/test split).

TFL. In Figure 8.2a, we plot the aggregate correlation coefficients for spatio-temporal points, calculated over the 2 most important principal components of each victim. We see that events in various locations and times yield high correlation values (dark red), indicating that diverse events *do* contribute to membership inference. Looking at the ROIs, we find that movements in tube stations or in overground stations have significant correlation towards distinguishing commuters. As for time, we clearly see differences in the patterns of week days (ids 1–120)



(a) Original heatmap

(b) Sorted heatmap: all victims

(c) Sorted heatmap: top 10% victims

(d) Sorted heatmap: bottom 10% victims

FIGURE 8.2: TFL, Subset of Locations prior ($\alpha = 0.11, m = 1{,}000, |T_I| = 168$): Aggregate spatio-temporal correlation over the top 2 components per victim: (a) original heatmap, ascending-order sorted heatmap by location and time slot popularity computed on (b) all victims, (c) top 10%, and (d) bottom 10% of distinguishable victims.

and of weekends (ids 121–168), with commuting hours having high correlation values. Interestingly, for some ROIs, mid-day hours also yield high correlation, possibly because less commuters use the transportation system. The same happens with weekend events (right side of the heatmap): the presence of users at stations during these times might reveal her membership in the aggregates.

We then study the importance of location and time popularity on membership inference. To this end, we plot the aggregate correlation heatmap sorting both locations and time according to their frequency of appearance in the dataset (see Fig. 8.2b). As expected, the more popular locations and time slots (upper-right corner) yield the highest correlation, since most of the events are generated in such locations and times. Nonetheless, data points in popular locations but reported in less popular time slots (upper-left corner) are also important, indicating that commuters can be distinguished in the aggregates by visiting such locations if

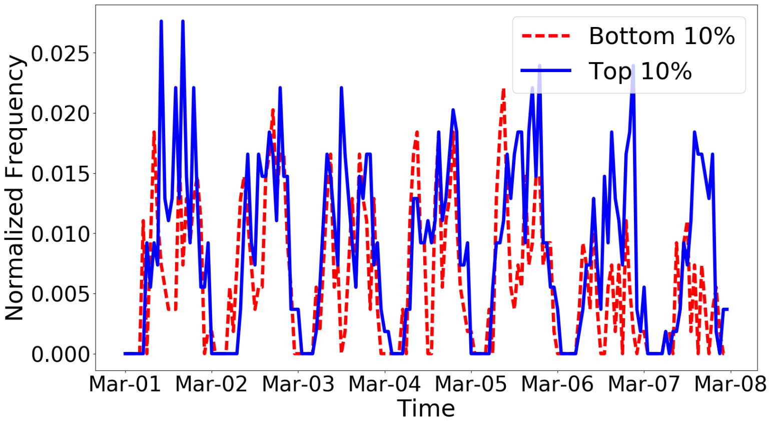


FIGURE 8.3: TFL, Subset of Locations prior ($\alpha = 0.11, m = 1,000$, $|T_I| = 168$): Aggregate normalized frequency of time slots over the inference week.

this happens at rare times. Finally, we note that there are few points in less popular locations (and various times) that obtain high correlation values, i.e., movements in sparser locations can give away a commuter's presence in the aggregates.

Next, we study the most and least distinguishable victims, according to their AUC, to gain insights on what makes commuters more or less susceptible to membership inference (see Figs. 8.2c–8.2d). For the top distinguishable victims, we observe very high coefficients for relatively unpopular locations and times (middle part of the heatmap), i.e., people visiting rare locations at rare times are easy to attack, and while the most popular ones (top right) do yield high correlation, they do not seem to be as crucial. On the other hand, for the less distinguishable commuters, popular locations and times (top right part of the heatmap) are the most important, and no other locations seem to help the attack. We conjecture that these are commuters that mostly travel at busy stations/times and their data hides better along with those of the crowd. To confirm this hypothesis, we study the aggregate (normalized) frequency of the time slots in the inference week over the two groups in Figure 8.3. We see that the more distinguishable commuters (blue solid line) have higher frequency in the middle and late evening hours of the week days (i.e., when the transportation system is less crowded) as well as during the weekends. This indicates that sporadic movements at these hours make it easier for the adversary to infer membership. Whereas, the less distinguishable oysters (dotted red line) mostly move during commuting hours and their privacy is less harmed by the release of aggregates.

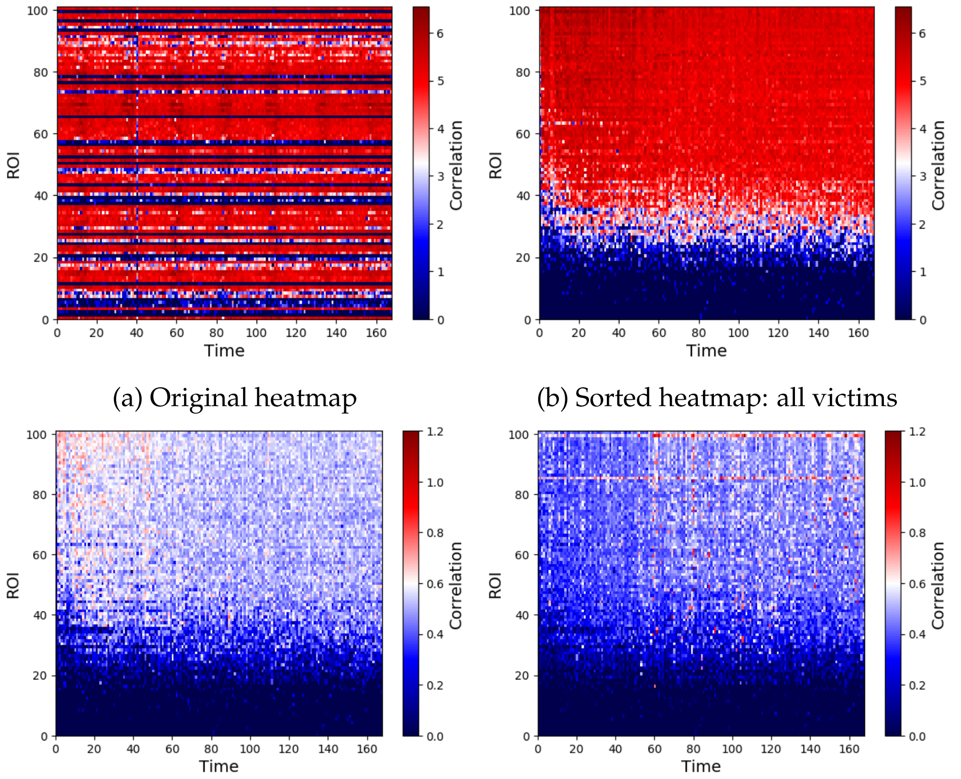
Feature	**TFL**	**SFC**
Total events	0.03	0.17
Unique locations	0.39	0.01
Active time slots	0.06	0.23
Locations per time slot	0.05	0.30
Active time slots / week day	0.01	0.01
Active time slots / weekend	0.11	0.01
Events / week day	0.01	0.07
Events / weekend	0.13	0.03
Spatial entropy	0.01	0.03
Temporal entropy	0.06	0.01
Unicity	0.16	0.17

TABLE 8.1: Subset of Locations prior: Feature importance of a Random Forest classifier for the top and bottom 10% distinguishable victims: TFL ($\alpha = 0.11, m = 1,000, |T_I| = 168$) and SFC ($\alpha = 0.2, m = 100, |T_I| = 168$).

Finally, to better understand what are the mobility characteristics that make the top and bottom 10% distinguishable oysters different, we feed them to a Random Forest Classifier, and examine which features can separate the two groups. The results are reported in Table 8.1. We see that the most important feature is the number of unique locations visited by an oyster, suggesting that visiting more (unique) locations increases the attack's surface and subsequently its success. The second most important one is the uniqueness of a travel pattern, highlighting a relationship between membership inference and the unicity of mobility patterns. We examine the unicity statistic and we find that the top victims have unique mobility pattern for 14 ± 5 time slots of T_I while the bottom ones for 4 ± 1. We also confirm that activity during the weekends differentiates the two groups of victims (cf. Fig. 8.3).

SFC. As in the previous case, we calculate the aggregate spatio-temporal correlation coefficients (see Fig. 8.4a). These are calculated over the cabs' top 5 components. As opposed to TFL, here a large number of locations yield high coefficients, highlighting that GPS movements offer a larger attack surface than the tap-in and out events in London stations. We see a similar effect for time, with areas of particularly high correlation concentrated around mid-day hours, but also during weekends when less drivers work shifts.

When studying the influence of popularity (Fig. 8.4b), unsurprisingly, we find that the most popular locations have high correlation values. However, contrary to TFL where only a small subset of stations and times are relevant (recall Fig. 8.2b), in SFC, also mid-popular



(a) Original heatmap

(b) Sorted heatmap: all victims

(c) Sorted heatmap: top 10% victims

(d) Sorted heatmap: bottom 10% victims

FIGURE 8.4: SFC, Subset of Locations prior ($\alpha = 0.2, m = 100$, $|T_I| = 168$): Aggregate spatio-temporal correlation over the top 5 components per victim: (a) original heatmap, ascending-order sorted heatmap by location and time slot popularity computed on (b) all victims, (c) top 10%, and (d) bottom 10% of distinguishable victims.

locations (i.e., ids 40-60), as well as certain hours in less popular ROIs (ids 20-40) obtain significant correlation. This suggests that there are many more regions and times that help membership inference on this dataset.

Next, we compare the coefficients of the top and bottom distinguishable cabs (Figs. 8.4c–8.4d). The former obtain slightly higher coefficients in the most popular locations but during the least busy times. This confirms that movements in less popular times enhance membership inference. At the same time, for these cabs the counts of popular locations and time slots also yield high values of correlation, suggesting that they contribute a large portion of data points during the inference week. On the contrary, the heatmap of the least distinguishable cabs is much sparser: most of them contribute a few amount of data and the attack has little information to build on. Similarly to TFL, the most popular locations and time slots get the highest correlation. We conclude that for these cabs these regions are not very revealing, as

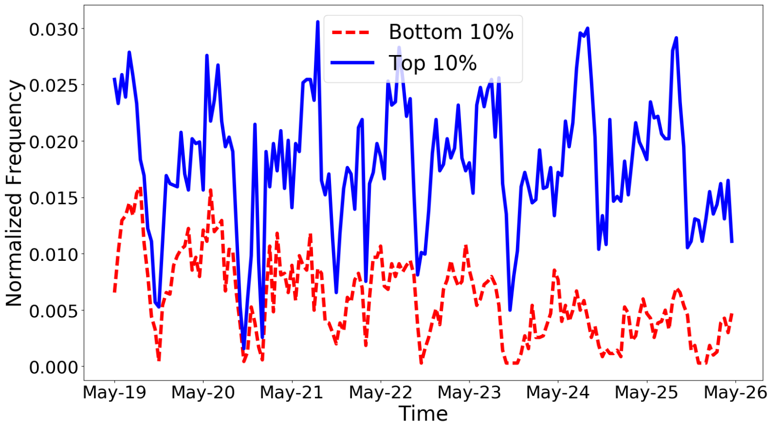
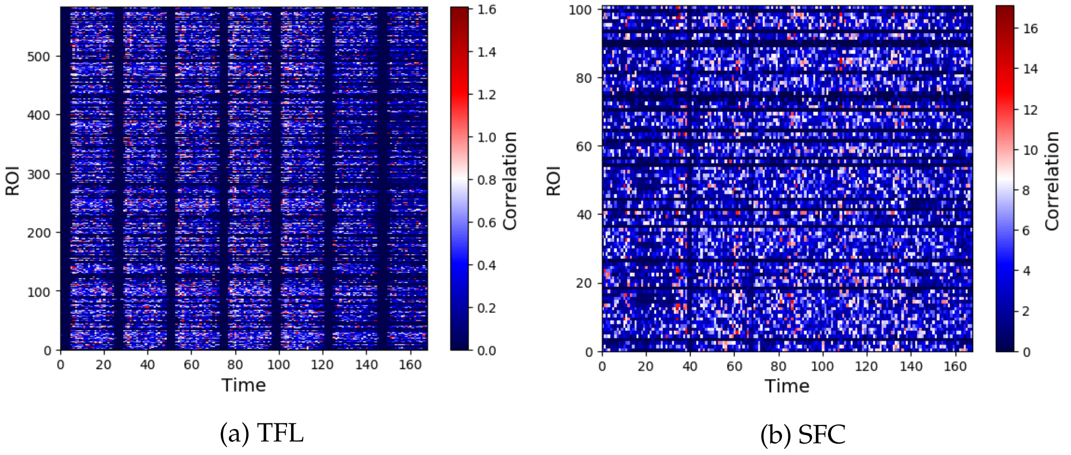


FIGURE 8.5: SFC, Subset of Locations prior ($\alpha = 0.2, m = 100, |T_I| = 168$): Aggregate normalized frequency of time slots over the inference week.

most cabs contribute these data points as well.

In terms of frequency, Figure 8.5 shows that the top distinguishable cabs (solid blue line) have higher frequencies during late night hours of weekdays and during the weekends, highlighting that movements in specific hours (e.g., due to the shift hours of a cab driver) can facilitate membership inference attacks. Whereas, the less distinguishable cabs (red line) contribute some data points in the beginning of the week, but less afterwards. This confirms the intuition that the most distinguishable cabs are those which contribute larger volume of spatio-temporal points, i.e., bigger data contribution enhances the performance of the attack.

Finally, the rightmost column of Table 8.1 shows the classifier's feature importance for the mobility characteristics of the top and bottom 10% distinguishable cabs. The top feature is the mean number of locations per time slot, followed by the number of active time slots, total number of events, and mean locations per week day. Thus, unlike TFL where most commuters report similar volumes of data, in the SFC dataset vehicles with more data points are overall more susceptible to membership inference attacks. Similar to TFL, we also find that the attack's performance is strongly linked to uniqueness of the cabs' mobility trajectories. The top distinguishable cabs exhibit larger unicity (i.e., their patterns are unique for 124 ± 6 out of the 168 time slots of T_I) than the bottom ones (unique pattern for 38 ± 30 time slots).



(a) TFL

(b) SFC

FIGURE 8.6: Same Groups as Released prior: Aggregate spatio-temporal correlation over the top components per victim for (a) TFL ($\beta = 500, m = 9,500$, $|T_I| = 168$) and (b) SFC ($\beta = 800, m = 500, |T_I| = 168$) datasets.

8.1.3 Participation in Past Groups

Next, we perform our analysis for the case where Adv knows the target victim's participation in aggregate location data released during an observation period T_O.

Parameters. We set the size of the user groups as $m = 9,500$ for TFL and $m = 500$ for SFC. We consider T_O to be the first few weeks of each dataset (3 for TFL and 2 for SFC), and use them to construct the prior knowledge that the adversary relies on to train her classifier. The adversary runs the attack on the last week of data (i.e., $|T_I| = 168$). We configure the number of known groups as $\beta = 500$ for TFL and $\beta = 800$, for SFC, i.e., we create large enough training and testing datasets (of 2K samples for TFL and 2.4K for SFC).

Same Groups as Released

First, we consider the case where the adversary performs membership inference on the aggregates of the same groups as those on which she trained her classifier. We plot the aggregate correlation coefficients for the most important principal components of the victims in TFL (top 1 component/victim) and SFC (top 5 components/victim) in Figure 8.6. For TFL (left plot), the most correlated data points now occur during the morning commuting hours of weekdays, highlighting that regularity in mobility patterns, e.g., the daily commute to work, helps membership inference. This explains why this attack is very effective on TFL with this type of prior (cf. Section 7.3.2). Interestingly, when we consider location and time popularity,

Feature	**TFL**	**SFC**
Total events	0.20	-0.36
Unique locations	0.78	1.29
Active time slots	-0.17	0.05
Locations per time slot	0.01	-0.33
Active time slots / week day	-0.48	0.28
Active time slots / weekend	0.42	-0.46
Events / week day	-0.47	-0.38
Events / weekend	0.64	-0.17
Spatial entropy	0.52	-0.18
Temporal entropy	0.17	-0.06
Unicity	-1.55	-0.68

TABLE 8.2: Different Groups than Released prior: Feature importance of a Logistic Regression classifier for the top and bottom 10% distinguishable victims: TFL ($\beta = 500, m = 9,500$, $|T_I| = 168$) and SFC ($\beta = 800, m = 500$, $|T_I| = 168$).

we find that correlated events are distributed across the board; popular locations or times as well as less popular locations on popular times (and vice versa) contribute to the success of the inference. This shows that commuters exhibit different regular patterns that are equally useful for the attack.

For SFC (Fig. 8.6b), movements in some weekdays' slots yield high correlation, i.e., there exist some regular cabs that are more susceptible to membership inference than others. Looking at the location and time slot popularity, we find high correlations scattered in the spatio-temporal space. Nonetheless, we observe that movements during less popular time slots obtain slightly higher correlation values in the components, i.e., cabs regular at such times are prone to the attack. This is consistent with some of the observations we made in Section 7.3.2, where we showed that the attack does not work very well with this prior as most cabs do not exhibit the same mobility patterns over the weeks.

Remark. We do not analyze the mobility characteristics of the most and least distinguishable victims with this prior, since: 1) for TFL, all commuters are harmed equally (AUC score of 1.0), and 2) for SFC, the insights are similar as with the Different Groups than Released prior, discussed next.

Different Groups than Released

Finally, we investigate the setting where Adv performs membership inference over different groups than those on which she trained her classifier. Overall, the aggregate correlation over the principal components for the victims in both datasets yield similar insights as for the Same Groups as Released prior (thus, we omit plotting the corresponding heatmaps): regular mobility patterns contribute to the success of membership inference. Nevertheless, it is not clear what locations or times are more important, i.e., various types of regular patterns make membership inference attacks successful.

We then compare the mobility characteristics of the top and bottom 10% distinguishable victims using a Logistic Regression classifier. Table 8.2 shows the model's coefficients for the computed features. Negative and positive coefficients indicate the more and less distinguishable victims, respectively.

For TFL, the strongest feature for the distinguishable oysters is the uniqueness of a commuting pattern, i.e., the more unique movements are, the easier it is to infer membership on dynamic groups. In particular, we find that the top victims have unique pattern for 47 ± 13 out of the 672 hourly time slots of the TFL dataset, whereas the bottom ones exhibit unicity for 32 ± 7 time slots. Moreover, features related to *time* patterns play an important role in separating the two groups; e.g., the top distinguishable users are mostly contributing events during days of the week, while the bottom ones report more events in the weekends. Looking at the individual trajectories, we find that the top users are mostly regular week day commuters in less popular ROIs (thus, more unique), while the less distinguishable users travel during weekends, to locations that are not on their regular week day pattern. This is confirmed by other features with high coefficients, such as the number of unique locations, or the spatial entropy. Overall, this means that irregular patterns which do not hold over time reduce the performance of membership inference when the adversary trains a classifier on aggregate data of past groups.

For SFC, we again find that the attack's performance is linked to uniqueness. Top victims exhibit unique mobility pattern for 357 ± 45 out of the 504 hourly time slots of the dataset, and the bottom ones are unique for 287 ± 85 time slots. Nonetheless, features related to the amount of data contributed by the cabs are stronger for the more distinguishable cabs. This suggests that cabs which are regular in their shifts (e.g., frequently working on weekends) and report larger volumes of data, are more identifiable. Finally, we find that the number of unique locations is stronger for the least distinguishable cabs, supporting our previous intuition that showing up in lots of locations, but without repeating patterns, reduces the adversarial performance.

8.1.4 Discussion

Overall, our analysis yields a few interesting insights. First, we show that the performance of membership inference on aggregate location time-series can be significantly boosted using dimensionality reduction techniques such as PCA; up to 65% mean AUC increase for TFL and 46% for SFC. This is because the aggregate location data retains strong spatio-temporal correlations with the data provided by the individual users.

Moreover, we find the spatio-temporal correlations within the principal components to be aligned with the mobility patterns in the data. For instance, commuting patterns emerge quite clearly in the components of the TFL dataset, while dense GPS trajectories create a large attack surface to be exploited by membership inference on the SFC dataset. In both cases, there are various spatio-temporal points and trends that contribute towards the success of the inference. Thus, designing a robust defense against this type of privacy attack for aggregate location time-series is an extremely challenging task.

By comparing the mobility characteristics of the most and least distinguishable victims, we gain understanding of which factors affect the success of the inference. In particular, we find that: (a) users who contribute more data points to the aggregation are more susceptible to membership inference; (b) movements in sparse locations and/or time slots can give away one's presence in the aggregates; and (c) unique and regular mobility patterns are more identifiable in the aggregates.

Finally, we also identify factors that negatively affect the performance of the attack. For instance, presence in popular locations and times contributes less to the attack's success, and irregular movements that do not hold over time can decrease the attack's power when the adversary trains her classifier using past information.

8.2 Evaluation of Defenses

The analysis presented in Section 8.1, shows that the supervised learning based approach for membership inference behaves in different ways for various adversarial settings, and for diverse mobility patterns. That is, it is not possible to identify a fundamental set of features to be protected, and as a result, there is no straightforward approach to distill a defense that directly tackles the core of membership inference on aggregate location data. In theory, there is an established framework to define private functions that are free from inferences, namely, Differential Privacy (DP, see Section 2.2). However, in Section 7.4, we showed that when DP protects against membership inference, the resulting noisy aggregates have poor utility, and are hardly useful for analytics tasks.

Therefore, in this section, we set to explore whether common techniques from the location privacy literature focused on non-aggregate settings (see Section 3.4) can be adapted to protect aggregate location time-series against membership inference for certain settings or tasks (i.e., specific mobility analytics which we describe in Section 8.2.1). We experiment with generalization and hiding techniques like suppression or sampling, using the insights obtained from our analysis to select the variants that are potentially suitable to hinder membership inference. More specifically, since uniqueness is linked to the success of the attack, we experiment with generalization; also, as contributing events in less popular locations and times makes users more distinguishable, we evaluate suppression, while sampling might be useful as it reduces the amount of data that users contribute to the aggregation. We briefly describe the defense strategies that we will evaluate in the remaining of this section.

Generalization. This type of defense consists in reducing the precision with which spatio-temporal events are reported. For instance, Gruteser et al. [GG03] propose the use of spatio-temporal obfuscation when querying location based services (LBS) to achieve *k-anonymity* while Bamba et al. [Bam+08] present a framework achieving additional properties like *l-diversity* in the same setting. Spatio-temporal generalization techniques have been used to mitigate various inferences from mobility trajectories [Kru07] as well as reduce their uniqueness [DM+13]. Additionally, bucketing techniques can also provide data generalization (e.g., rather than releasing exact statistics, report them in ranges). These have been used to protect privacy in other domains as well, e.g., obfuscating the length of network packets to prevent website fingerprinting [Cai+14] or providing inexact statistics to advertisers in online social networks [Ven+18].

Hiding. Another approach is to *not* include some of the users' spatio-temporal data points in the aggregates, by either suppressing or sampling them [Hoh+07; Sho+11a; Che+13]. For instance, Hoh et al. [Hoh+07] suppress some points in a location dataset containing single users' traces before releasing them. Overall, with hiding techniques, the released locations are not perturbed with any kind of noise, but *sensitive* points are suppressed.

Perturbation. As already mentioned, the state-of-the-art technique for the release of noisy aggregate statistics that mitigate inference attacks is Differential Privacy (DP) [Dwo08]. DP's indistinguishability-based definition ensures that the output of a data release is not significantly affected by the presence (or absence) of the data of any particular individual. As a representative of perturbation techniques and for comparison purposes with the previous techniques, we evaluate the Fourier Perturbation Algorithm (FPA, see Section 2.2.2) which achieved a good privacy and utility tradeoff in the evaluations of Chapter 7.

Remark. We do not consider strategies based on adding dummies [KYS05; SGI09; MRC09], as generating plausible dummy locations has been shown to be hard [CG09], and thus, they are likely to be easily filtered by an adversary. We also do not analyze synthetic data generation techniques [BS16; Mac+08], since, either they are not straightforward to adapt to the aggregate setting [BS16], or they only work for one specific task (e.g., computing origin-destination commute distances [Mac+08]).

8.2.1 Experiment Design

Adversarial Settings

We focus on the settings where the performance of membership inference is high, aiming to evaluate the defenses against a strong adversary. For TFL, we consider the Same Groups as Released prior knowledge, with $\beta = 500$, groups of $m = 9,500$ users, and T_O being the first *3 weeks* of the dataset while T_I the last one (i.e., $|T_I| = 168$). For SFC, we choose the Subset of Locations prior, with $\alpha = 0.5$, $m = 250$, and both T_O and T_I being the *first week* of data. The above decisions allow us to quantify the performance of various defense mechanisms against adversaries with different type of prior knowledge and on two datasets with different characteristics. Finally, we assume a *strategic* adversary that mirrors the mechanism employed by the defender during training (i.e., she applies the same defense on the aggregate location time-series before using them for training her classifier).

Privacy and Utility Metrics

Aiming to understand which defense is more suitable for which settings of aggregate location time-series, we evaluate both the extent to which they prevent membership inference, as well as their impact on the utility of the aggregates for different tasks or applications.

Privacy Gain. Following the evaluation of defenses presented in Section 7.4, we measure the effectiveness of a defense as the normalized decrease in the adversarial performance (i.e., by calculating the classifier's AUC score before and after a defense is applied). In particular, we employ the Privacy Gain (PG) metric described in Section 7.4.1 (recall Eq. 7.4), which captures how much the inference's power drops towards the random guess baseline (AUC score of 0.5) where users enjoy *perfect* privacy.

Utility. We reason about the impact of a defense strategy on the utility of the data using a few metrics that indicate the quality of the aggregate time-series for various analytics tasks. In general, as described in Section 3.1, location time-series are used for a wide range of mobility analytics, e.g., forecasting traffic volumes in Regions of Interest (ROIs) or detecting mobility anomalies [CZH12; HZS16; ZZQ17]. Hence, rather than only relying on the Mean Relative Error (MRE), as done in Chapters 6 and 7, we also consider other metrics such as different error definitions, correlation coefficients for anomaly detection, accuracy of prediction tasks, and distribution similarity, which we describe next.

Error metrics. These quantify the effect of a defense on the precision of the data release and indicate how useful are the location time-series for forecasting traffic analytics (similar to those performed in Chapter 5). Besides the Mean Relative Error (MRE), we also compute the Mean Absolute Error (MAE, see Section 2.3.1) over all the time-series or a percentage thereof, and report their average values.

Correlation coefficients. Such coefficients indicate whether a linear relationship between the two time-series, i.e., before and after applying a defense, is preserved. This is important, e.g.,

Dataset	F1	τ (10%)	τ	JS	r
TFL	0.097	0.003	0.001	0.733	-0.002
SFC	0.094	-0.006	0.001	0.472	0.007

TABLE 8.3: Utility metrics corresponding to a random guess.

when using aggregate locations to detect mobility anomalies or improve traffic prediction in the presence of an anomaly (i.e., similar to the enhanced analytics performed in Chapter 5). To this end, we use the Pearson's correlation coefficient, r, described in Section 2.3.2.

Prediction accuracy. A task that analysts are often interested in, is predicting *location hotspots* over time. This is particularly useful for journey planning and resource allocation purposes, e.g., transportation authorities need to learn which stations are the busiest in certain hours of a day and allocate staff accordingly, or suggest alternative routes to commuters. Moreover, hotspot detection is crucial for identifying the optimal location and time to place advertisements or open new shops [Tel18]. We measure utility for this task by using the aggregate time-series after a defense has been applied to predict the busiest 10% ROIs at each time slot of the inference week, and calculate the F1 score (see Section 2.1.2) to capture the performance of the predictions.

Note that, while the F1 score quantifies how successful hotspot detection is in each time slot, it does not capture if the ordering of the hotspots is preserved. This might be important for resource planning, e.g., a taxi company assigning vehicles to locations sorted by client demand. Therefore, we also calculate the Kendall rank correlation coefficient, τ (see Section 2.3.2), over the top 10% or all hotspots of each time slot.

Distribution similarity. Finally, tasks like location labeling and map inference rely on the fact that certain locations are more frequently visited than others [BS16]. Thus, we use the Jensen-Shannon (JS) divergence (see Section 2.3.3), which estimates the similarity between two probability distributions, to capture whether the distribution of location visits is preserved (for each time slot) after employing a defense.

Random Guess. To ease comparisons with respect to the results presented in the next section, Table 8.3 shows for each of the considered metrics, the utility corresponding to a random guess.

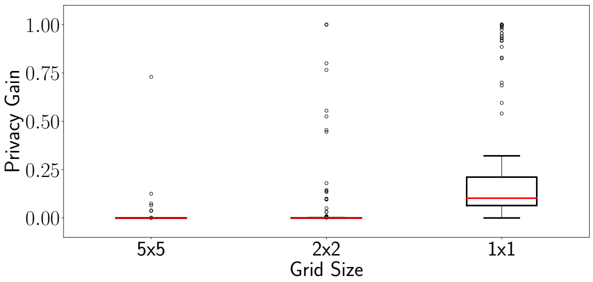


FIGURE 8.7: SFC, Spatial generalization: Privacy Gain (PG).

Grid Size	MRE	MAE	MRE 10%	MAE 10%	F1	τ 10%	τ	JS	r
5x5	0.840	18.628	0.019	20.964	0.534	0.036	0.112	0.215	0.684
2x2	71.270	82.246	0.055	57.852	0.344	0.016	0.158	0.402	0.507
1x1	114.199	149.011	0.086	91.902	0.049	0.017	-0.183	0.434	0.367

TABLE 8.4: SFC, Utility metrics for spatial generalization.

8.2.2 Results

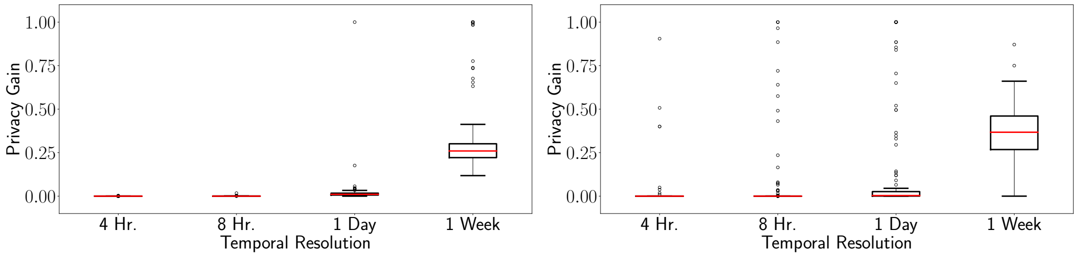
We now present our experimental results for the defense strategies discussed above.

Generalization

Our first set of experiments investigates how *generalization* techniques impact the performance of membership inference attacks. In the context of aggregate location time-series, generalization can be implemented either in the spatial domain or the temporal one, as well as in the data (counts) of the time-series.

Spatial generalization. We focus on the SFC dataset, since it contains the GPS coordinates visited by cabs, and thus, is more amenable to this defense than tube stations (TFL). We keep the temporal resolution of the dataset to one hour (see Chapter 4), and use grids of different spatial resolution to divide the 30.3mi^2 area of downtown San Francisco, ranging from a baseline 10 × 10 grid resulting in 100 ROIs of 0.3mi^2 each, to one single ROI of 30.3mi^2.

In Figure 8.7, we report a box plot with the Privacy Gain (PG), computed over the 150 target cabs, for different spatial resolutions with respect to the baseline, i.e., the privacy level on the 10 × 10 grid. Overall, we observe that spatial generalization does *not* provide significant privacy protection against membership inference: when increasing the size of the ROIs



(a) TFL

(b) SFC

FIGURE 8.8: Temporal generalization: Privacy Gain (PG).

Time slot	MRE	MAE	MRE 10%	MAE 10%	F1	τ 10%	τ	JS	r
4 Hours	0.146	14.768	0.152	102.586	0.776	0.076	0.332	0.272	0.596
8 Hours	0.308	28.974	0.308	197.312	0.753	0.049	0.315	0.426	0.474
1 Day	0.777	63.913	0.741	418.643	0.738	0.020	0.296	0.605	0.225
1 Week	2.945	171.575	2.221	1046.286	0.651	0.003	0.284	0.658	0.000

TABLE 8.5: TFL, Utility metrics for temporal generalization.

to a 5 × 5 or even a 2 × 2 grid, the mean PG is negligible, except for some outliers. Only when we consider a single ROI (1 × 1 grid), the PG increases slightly (0.25 on average). This means that an attacker can perform membership inference on this dataset even without any spatial information. In other words, the temporal dimension of the location contains enough information for the attack to succeed, when the adversary has the Subset of Locations prior.

In Table 8.4, we report the various utility metrics as per different grid sizes. We compute these by projecting the aggregates of the generalized grid to the corresponding cells of the 10 × 10 one. Unexpectedly, all utility metrics deteriorate towards the random guess baseline (cf. Table 8.3) with larger grid sizes, except for Pearson's correlation (r) of the time-series for grids with 25 or 4 ROIs. That is because aggregating (correlated) neighboring cells of the grid retains the trends of their time patterns.

Temporal generalization. We then experiment with the temporal resolution of the aggregates, varying the length of each time slot from 1 hour (the baseline) to 1 week[1]. Figure 8.8 shows the PG computed over the 150 victims under attack, for both datasets, using different temporal resolutions. Note that we set the attack's performance for aggregates of 1 hour temporal resolution, as our baseline for privacy and utility. For TFL, the attack's performance

[1] Note that for SFC, we keep the spatial resolution at the original 10 × 10 grid.

Time slot	MRE	MAE	MRE 10%	MAE 10%	F1	τ 10%	τ	JS	r
4 Hours	0.134	30.791	0.066	72.239	0.823	0.056	0.150	0.139	0.632
8 Hours	0.280	56.539	0.104	113.568	0.781	0.074	0.145	0.186	0.413
1 Day	0.703	99.678	0.135	146.583	0.709	0.040	0.141	0.246	0.086
1 Week	2.493	148.642	0.149	161.280	0.455	-0.057	0.135	0.324	0.000

TABLE 8.6: SFC, Utility metrics for temporal generalization.

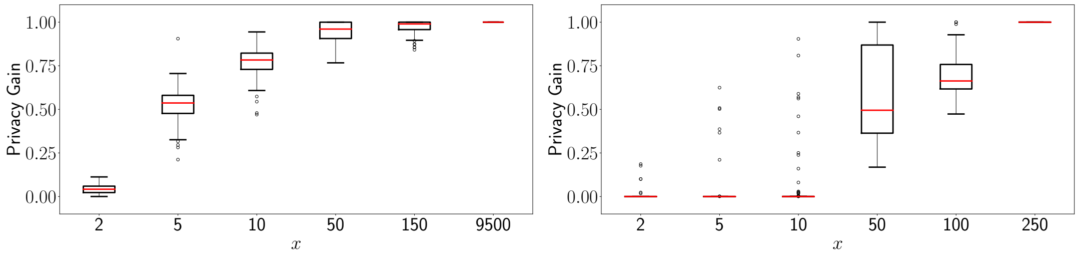
remains almost unaffected for time slots up to 1 day. This means that regular commuting patterns remain distinguishable in the aggregates, even for relatively long periods of time. The privacy gain only increases significantly for 1-week granularity, 0.31 on average. A similar trend can be observed with SFC, although the attack's performance starts degrading earlier (the mean PG is 0.15 for 1 day resolution), reaching a mean PG of 0.35 with a 1-week time slot. This highlights that just 1 time point may be sufficient for the membership inference attack, confirming that, as the temporal dimension shrinks, the spatial domain still contains enough information to perform inference.

In Tables 8.5 and 8.6, we report the utility metrics for both datasets with different temporal resolutions (the counts of the generalized aggregates are copied onto the corresponding time epochs for 1-hour granularity). While all utility metrics degrade with higher generalization, the F1 score is overall reasonably high, indicating that the busiest ROIs are relatively stable over time. This hints that temporal generalization could work relatively well for the task of discovering hotspot ROIs.

Remark. Applying generalization in the spatial and temporal domain, simultaneously, will increase the privacy gain. In fact, we have experimentally verified that on the SFC dataset, the mean PG reaches 0.96 when performing the attack on a 1×1 grid with 1-week temporal resolution. However, in such a setting, the resulting aggregates are hardly useful for any task.

Data generalization. Next, we experiment with reporting *ranges*, instead of exact spatio-temporal aggregates. For example, rather than reporting that there were 124 users in a given ROI during a 1-hour time slot, we report the range 120–130.

Fixed ranges. First, we fix the interval size (denoted as x) depending on the group size parameter m, i.e., we vary the interval based on the minimum and maximum possible values of the overall aggregates (as indicated by m). Then, we assign the location counts to the median



(a) TFL

(b) SFC

FIGURE 8.9: Data generalization: Privacy Gain (PG) with fixed intervals of size x.

x	MRE	MAE	MRE 10%	MAE 10%	F1	τ 10%	τ	JS	r
2	0.888	0.773	0.002	0.592	0.955	0.356	0.467	0.108	0.641
5	1.787	1.627	0.004	1.350	0.918	0.224	0.419	0.153	0.393
10	4.481	4.083	0.008	2.948	0.853	0.183	0.406	0.238	0.278
50	22.733	22.159	0.037	13.606	0.432	0.146	0.425	0.453	0.040
150	67.527	70.951	0.153	51.779	0.400	0.150	0.426	0.578	0.002
9500	4,327.790	4,743.204	13.264	4,699.148	0.399	0.149	0.429	0.724	0.000

TABLE 8.7: TFL, Utility metrics for data generalization with fixed intervals of size x.

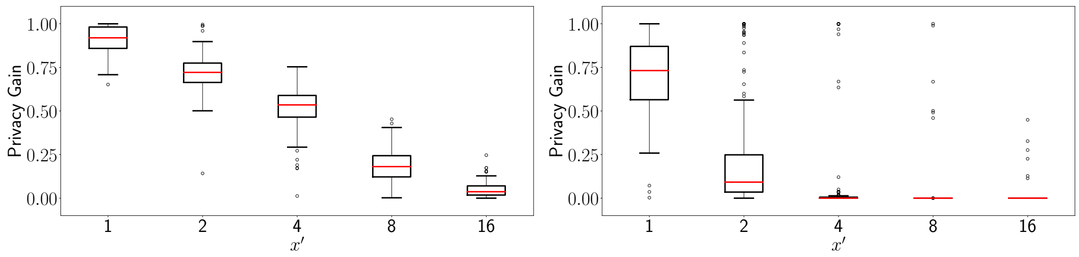
of a range interval. Figure 8.9 reports the PG obtained with this approach, on both datasets.

For TFL, when $x = 2$, there is no significant gain for commuters; however, for $x = 5$ and 10, respectively, mean PG increases to 0.52 and 0.77, showing that a small generalization suffices to provide good levels of privacy on sparse datasets. With larger range intervals, e.g., 50 or 150, the mean PG increases further to 0.94 and 0.97, respectively. When setting x to 9,500, i.e., the maximum possible count a station could have, PG reaches 1 for all users. This is because all the location counts over time are the same (i.e., there is no variance in the data), thus, membership inference is actually impossible. For SFC, data generalization requires larger intervals to have an effect on PG, due to its reduced sparseness as compared to TFL. For $2 \leq x \leq 10$, PG is almost negligible except for a few outliers. However, when $x = 50$, the mean PG reaches 0.58, and 0.68 for $x = 100$. Also in this case, using the maximum possible interval (i.e., $x = 250$), results to the maximum PG for all users.

As shown in Tables 8.7 and 8.8, even though the utility metrics deteriorate with larger intervals towards the random guess for metrics like F1, JS, and Pearson's r, several analytics tasks can still be performed while balancing the privacy/utility tradeoff. For instance, on

x	MRE	MAE	MRE	MAE	F1	τ	τ	JS	r
			10%	10%		10%			
2	0.711	0.599	0.000	0.509	0.984	0.792	0.343	0.070	0.813
5	1.381	1.344	0.001	1.217	0.952	0.509	0.212	0.102	0.705
10	3.079	3.097	0.002	2.519	0.908	0.330	0.160	0.165	0.596
50	15.670	16.401	0.011	12.032	0.663	0.063	-0.062	0.354	0.239
100	28.046	36.373	0.021	23.040	0.210	0.041	-0.154	0.417	0.055
250	70.068	106.591	0.054	58.060	0.103	0.022	-0.161	0.432	0.000

TABLE 8.8: SFC, Utility metrics for data generalization with fixed intervals of size x.



(a) TFL

(b) SFC

FIGURE 8.10: Data generalization: Privacy Gain (PG) with adaptive ranges (i.e., using x' buckets).

the sparse TFL dataset, small perturbations (e.g., $x = 5$) provide good PG (0.52, on average), while enabling tasks like forecasting traffic volumes on top stations (MRE $= 4 \cdot 10^{-3}$), detecting hotspots (F1 $= 0.91$), and location labeling (JS $= 0.15$). On the SFC dataset, where cabs contribute lots of data points, larger intervals are needed for privacy. Yet, with $x = 50$, we obtain a mean PG of 0.58, and forecasting traffic volumes on the top locations is still possible due to relatively low MRE.

Adaptive ranges. We also evaluate an adaptive approach to select the range interval for each location based on its minimum and maximum value over time. By tailoring the range to the locations, we expect to obtain better utility results. The number of buckets released *per location* is indicated by the parameter x'. We report, in Figure 8.10, the PG for both datasets, while Tables 8.9 and 8.10 display the corresponding utility metrics. Unexpectedly, as we increase the number of buckets (x'), i.e., as we reveal more information about each location's counts over time, PG decreases and utility improves.

For TFL, publishing 1 or 2 buckets per location results in a mean privacy gain of 0.91 and 0.71, respectively (Fig. 8.10a). When increasing x', PG decreases: 0.5 for 4 buckets, and 0.18

x'	MRE	MAE	MRE	MAE	F1	τ	τ	JS	r
			10%	10%		10%			
1	0.140	5.240	0.041	32.698	0.711	0.051	0.315	0.175	0.440
2	0.022	2.671	0.024	17.604	0.715	0.042	0.338	0.117	0.811
4	0.005	1.175	0.012	8.062	0.785	0.070	0.350	0.068	0.888
8	0.002	0.568	0.006	4.337	0.821	0.090	0.362	0.039	0.901
16	0.000	0.230	0.003	2.079	0.851	0.151	0.457	0.020	0.902

TABLE 8.9: TFL, Utility metrics for data generalization with adaptive ranges (i.e., using x' buckets).

x'	MRE	MAE	MRE	MAE	F1	τ	τ	JS	r
			10%	10%		10%			
1	0.052	7.382	0.021	24.036	0.744	0.081	0.162	0.152	0.413
2	0.015	4.559	0.013	14.178	0.782	0.117	0.150	0.114	0.846
4	0.006	2.339	0.006	7.099	0.874	0.162	0.207	0.064	0.942
8	0.002	1.065	0.003	3.469	0.926	0.347	0.327	0.029	0.977
16	0.001	0.459	0.001	1.637	0.962	0.540	0.495	0.014	0.989

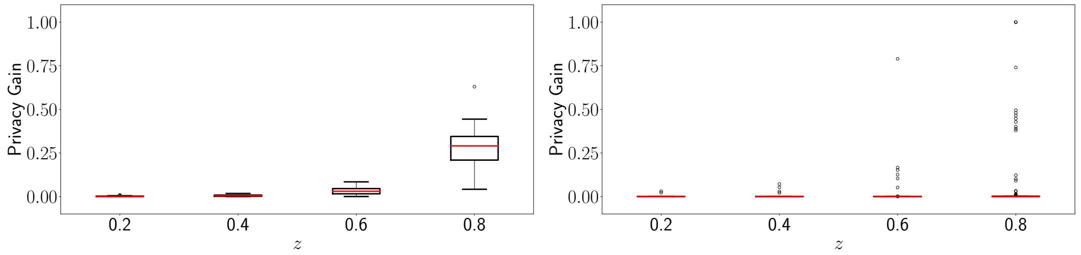
TABLE 8.10: SFC, Utility metrics for data generalization with adaptive ranges (i.e., using x' buckets).

for 8 buckets, and it becomes almost negligible for 16 buckets (0.05). Similarly, for the SFC dataset, publishing 1 bucket per location yields a mean privacy gain of 0.7. Interestingly, already with $x' = 2$, the mean PG drops to 0.25. This means that, as soon as information about the evolution of the aggregates over time is revealed, cabs' patterns are exposed to the attack. In terms of utility, the aggregates with this adaptive approach do retain better levels than that with fixed ranges. For instance, in both TFL and SFC datasets, if one reveals more than 2 buckets per location, most of the analytics tasks can be performed without a significant impact on utility.

Hiding

We now evaluate two defense strategies consisting in *hiding* certain data points, i.e., suppression and sampling.

Suppression. Our analysis presented in Section 8.1 shows that some commuters or cabs are more *distinguishable* in the aggregates because they contribute events in less popular locations or times. Therefore, we investigate whether suppressing such data points reduces the effectiveness of membership inference. More specifically, we consider a setting where the data



(a) TFL

(b) SFC

FIGURE 8.11: Suppression: Privacy Gain (PG).

z	MRE	MAE	MRE 10%	MAE 10%	F1	τ 10%	τ	JS	r
0.2	0.000	0.000	0.0	0.0	1.000	1.000	1.000	0.000	0.903
0.4	0.001	0.001	0.0	0.0	0.998	0.971	0.916	0.002	0.892
0.6	0.006	0.042	0.0	0.0	0.992	0.899	0.752	0.018	0.821
0.8	0.021	0.480	0.0	0.0	0.968	0.569	0.541	0.076	0.639

TABLE 8.11: TFL, Utility metrics for suppression.

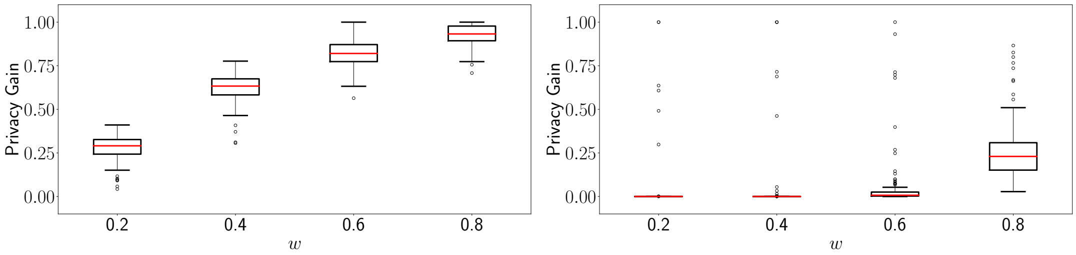
publisher suppresses a percentage z of locations and time slots based on their popularity, i.e., she does not include the least popular locations and time slots in the aggregates as such potential outliers might boost the power of the inference.

Figure 8.11 reports the privacy gain for increasing values of z. Interestingly, in both datasets, suppression does *not* provide significant PGs for the users that we attack. For TFL, values of z up to 0.6 yield negligible mean PG. Only when the provider releases aggregates for just the top 20% locations and times (i.e., $z = 0.8$), the mean PG reaches 0.27. This highlights that the counts of the busiest locations and times still contain information that can distinguish users in the aggregates, mirroring our analysis presented in Section 8.1. Similarly, for SFC, we get some privacy gain only for a small amount of victims, even when 80% of the data is suppressed. Thus, we conclude that suppression techniques do not provide good protection against membership inference in high-dimensional settings.

Finally, we report the utility metrics for various values of z in Tables 8.11 and 8.12 (note that we assign zero counts to the suppressed locations and times in order to calculate the various statistics). Suppressing outliers does not seem to generally affect the utility of the tasks we consider, e.g., forecasting the counts of each location, discovering hotspots, ranking, or labeling locations can be done with minimal error; however, as mentioned, this comes at a

z	**MRE**	**MAE**	**MRE 10%**	**MAE 10%**	**F1**	τ **10%**	τ	**JS**	r
0.2	0.001	0.002	0.0	0.0	1.000	1.000	0.960	0.003	0.960
0.4	0.006	0.155	0.0	0.0	0.999	0.985	0.761	0.039	0.892
0.6	0.016	1.561	0.0	0.0	0.989	0.918	0.488	0.149	0.721
0.8	0.031	5.744	0.0	0.0	0.953	0.799	0.208	0.335	0.524

TABLE 8.12: SFC, Utility metrics for suppression.



(a) TFL

(b) SFC

FIGURE 8.12: Sampling: Privacy Gain (PG).

big cost for privacy.

Sampling. In Section 8.1, we showed that another factor playing a part in the success of membership inference is the amount of data that users contribute to the aggregation. Therefore, we consider sampling the time-series, as a means to reduce the amount of contributed data. We remove a random percentage w of each user's data points, and release the aggregates computed on the sampled trajectories. We set w in the range $\{0.2, 0.4, 0.6, 0.8\}$ and report the resulting privacy gain values in Figure 8.12.

For TFL, we observe that this defense offers some privacy protection. For $w = 0.2$, the mean PG is 0.27, and increases up to 0.8 when removing 80% of the points. Thus, sampling can be a promising defense strategy against membership inference on sparse datasets. On the other hand, sampling does not work nearly as well on the dense SFC dataset. Here, the privacy gain is negligible even when 60% of the events are randomly removed, and only reaches 0.25 when 20% of the data points are left.

Tables 8.13 and 8.14 show that sampling yields acceptable utility for analytics tasks like discovering hotspots (relatively high F1 score), anomaly detection (Pearson's r), as well as preserving the locations' frequency distribution (JS).

w	MRE	MAE	MRE	MAE	F1	τ	τ	JS	r
			10%	10%		10%			
0.2	0.009	3.563	0.012	32.284	0.940	0.217	0.476	0.029	0.843
0.4	0.019	7.088	0.024	64.213	0.904	0.198	0.456	0.047	0.756
0.6	0.029	10.612	0.036	96.149	0.864	0.177	0.438	0.068	0.656
0.8	0.038	14.137	0.048	128.072	0.793	0.160	0.410	0.105	0.505

TABLE 8.13: TFL, Utility metrics for sampling.

w	MRE	MAE	MRE	MAE	F1	τ	τ	JS	r
			10%	10%		10%			
0.2	0.010	3.724	0.012	13.638	0.928	0.317	0.238	0.040	0.921
0.4	0.021	7.491	0.024	27.415	0.892	0.207	0.187	0.066	0.841
0.6	0.032	11.227	0.036	41.113	0.845	0.155	0.168	0.098	0.752
0.8	0.042	14.824	0.048	54.354	0.764	0.101	0.148	0.161	0.591

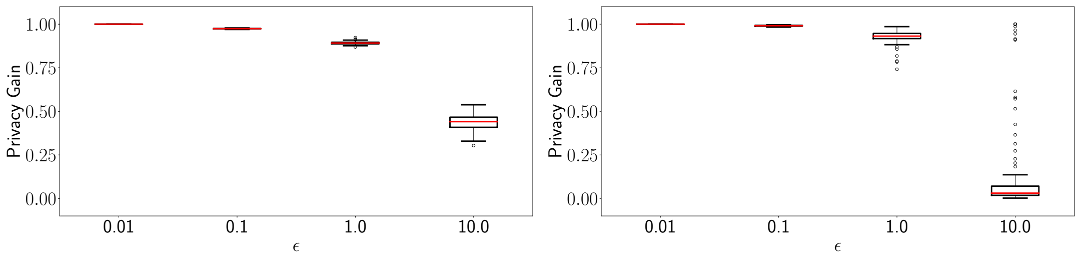
TABLE 8.14: SFC, Utility metrics for sampling.

Perturbation

We also report our experimental results for one of the defense strategies evaluated in Section 7.4, i.e., adding noise to the aggregate counts in such a way that Differential Privacy (DP) is guaranteed. As discussed earlier, we select the FPA algorithm [RN10] whose details are described in Section 2.2.2.

Figure 8.13 shows the privacy gain obtained with FPA, for values of ϵ in the range $\{0.01, 0.1, 1.0, 10.0\}$. As expected, PG is higher for smaller values of ϵ (i.e., when the mechanism achieves stronger DP guarantees). For TFL, the mean PG reaches 1.0 for $\epsilon = 0.01$, and only slightly decreases to 0.97 and 0.89, respectively, when ϵ is set to 0.1 and 1.0. For larger values of ϵ, e.g., 10.0, PG drops to 0.43. Similarly, on the SFC dataset, the mean privacy gain is very high for ϵ values up to 1.0 (e.g., 0.92 for $\epsilon = 1.0$) but drops to 0.11 when $\epsilon = 10.0$.

The various utility metrics for both datasets are reported in Tables 8.15 and 8.16. These confirm the results of Section 7.4, i.e., when ϵ is set to 0.01 or 0.1 and the privacy gain is very large, there are huge errors in the aggregates and most metrics are not performing better than a random guess (cf. Table 8.3). Yet, when $\epsilon = 1.0$, the perturbed aggregates can realistically be used for hotspot detection, as the F1 score amounts to 0.72 and 0.78, on resp., the TFL and SFC datasets, as well as forecasting tasks on the top 10% ROIs (e.g., MRE $\leq 2 \cdot 10^{-2}$ for TFL and SFC).



(a) TFL (b) SFC

FIGURE 8.13: FPA perturbation: Privacy Gain (PG).

ϵ	MRE	MAE	MRE 10%	MAE 10%	F1	τ 10%	τ	JS	r
0.01	111.748	127.660	0.383	143.771	0.124	0.033	0.006	0.666	0.006
0.1	11.603	15.111	0.058	29.218	0.250	0.029	0.035	0.413	0.042
1.0	1.163	3.469	0.022	16.431	0.721	0.044	0.204	0.176	0.232
10.0	0.085	2.474	0.021	16.082	0.758	0.050	0.336	0.116	0.449

TABLE 8.15: TFL, Utility metrics for FPA perturbation.

Combining Defenses: Hiding and Perturbation

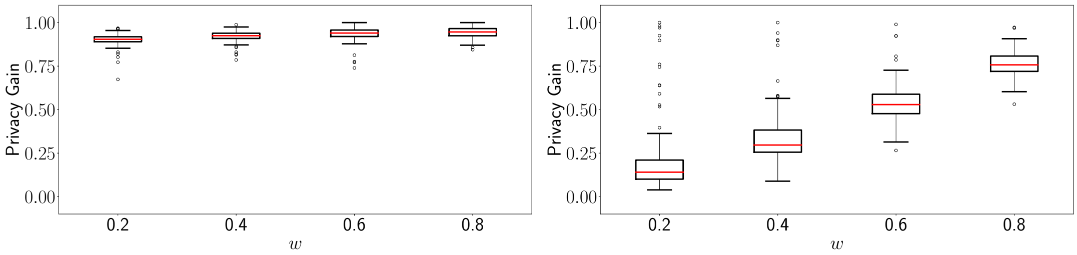
Recall that sampling yields reasonable privacy/utility tradeoffs in the TFL setting, but little privacy for SFC. We now investigate whether augmenting it with perturbation – concretely, with the FPA differentially private mechanism [RN10] – can provide DP's high privacy gain levels, without destroying utility. The intuition is that, as sampling reduces the users' sensitivity, less noise would suffice to gain good levels of privacy. Note that we focus on the SFC dataset for this experiment.

Figure 8.14 shows a box plot of the Privacy Gain (PG) for $\epsilon = 1.0$ and 10.0, for increasing values of the sampling rate w. We find that PG is significantly boosted for $\epsilon = 1.0$, i.e., it grows from very small values (less than 0.25, cf. Fig. 8.12b) to larger than 0.9, on average. However, the sampling rate w has almost no effect on the privacy gain, i.e., the improvement is mostly due to the noise addition by the FPA mechanism. In fact, Table 8.17 shows that the loss in utility is very similar to FPA and overall worse than sampling alone (cf., respectively, Tables 8.16 and 8.14). Therefore, in this case, there is no significant advantage in combining the methods.

On the other hand, for $\epsilon = 10.0$, the PG increases compared to both FPA and sampling individually (e.g., for $w = 0.6$, it is 5× larger than FPA, and 12× higher than sampling). As

ϵ	**MRE**	**MAE**	**MRE 10%**	**MAE 10%**	**F1**	τ 10%	τ	**JS**	r
0.01	73.020	111.273	0.104	115.982	0.140	0.063	-0.002	0.578	0.000
0.1	24.485	42.796	0.051	57.371	0.218	0.007	0.040	0.521	0.033
1.0	2.016	7.365	0.013	14.303	0.781	0.086	0.125	0.216	0.276
10.0	0.251	4.331	0.011	12.623	0.843	0.134	0.152	0.108	0.540

TABLE 8.16: SFC, Utility metrics for FPA perturbation.



(a) $\epsilon = 1.0$

(b) $\epsilon = 10.0$

FIGURE 8.14: SFC, Sampling in combination with FPA: Privacy Gain (PG).

before, the utility metrics (reported in Table 8.18) degrade compared to sampling, and are similar to FPA. Hence, we conclude that, for $\epsilon = 10.0$, combining the two defenses might be suitable for settings where some decrease in privacy gain (e.g., compared to the case of $\epsilon = 1.0$) can increase utility significantly.

8.2.3 Discussion

Overall, our experimental evaluation of defense strategies provides some very interesting findings. On the one hand, we observe that *spatio-temporal generalization*, a technique commonly used to protect privacy in the setting of mobility trajectories [Kru07; GG03; DM+13], does not really provide meaningful protection against membership inference, even when the utility of the aggregates is destroyed. Similarly, *suppression* techniques, which retain utility for various analytics under consideration (e.g., ranking of hotspots or location labeling), also fail to provide significant privacy gains, showing that the membership inference attack exploits multiple data points in the high-dimensional setting of aggregate location time-series.

On the other hand, *data generalization* approaches like discretizing the aggregate location counts can provide an acceptable balance on the privacy/utility tradeoff and allow analysts to perform a number of tasks, like forecasting location traffic, hotspot detection, and location

w	**MRE**	**MAE**	**MRE**	**MAE**	**F1**	τ	τ	**JS**	r
			10%	**10%**		**10%**			
0.2	1.719	7.821	0.017	19.375	0.773	0.094	0.123	0.229	0.258
0.4	1.629	9.930	0.026	29.656	0.757	0.073	0.112	0.250	0.231
0.6	1.274	12.220	0.036	41.101	0.714	0.088	0.115	0.279	0.185
0.8	1.240	15.171	0.048	53.919	0.620	0.074	0.105	0.330	0.140

TABLE 8.17: SFC, Utility metrics for sampling in combination with FPA ($\epsilon = 1.0$).

w	**MRE**	**MAE**	**MRE**	**MAE**	**F1**	τ	τ	**JS**	r
			10%	**10%**		**10%**			
0.2	0.223	5.520	0.016	18.415	0.841	0.140	0.149	0.110	0.523
0.4	0.199	8.016	0.025	28.524	0.836	0.142	0.151	0.114	0.500
0.6	0.185	11.175	0.036	40.366	0.830	0.129	0.153	0.120	0.472
0.8	0.114	14.899	0.048	54.392	0.818	0.115	0.159	0.135	0.408

TABLE 8.18: SFC, Utility metrics for sampling in combination with FPA ($\epsilon = 10.0$).

labeling. However, configuring the granularity of the generalization is challenging as this is highly dependent on the data under examination and requires extensive privacy vs. utility analysis, similar to that performed in this work.

Furthermore, our experiments show that *sampling* techniques can provide good privacy protection when the input signal is sparse (e.g., as in the case of the TFL dataset), and allow anomaly detection tasks or data analysis based on the distribution over the locations. Whereas, *perturbation* mechanisms providing DP guarantees only achieve reasonable utility for applications such as hotspot detection and forecasting traffic at the most popular locations, when they are configured to *relax* their privacy guarantees. Finally, we find that combining defenses, e.g., sampling and perturbation, can help privacy and, under circumstances, retain utility for some tasks (e.g., discovering hotspots).

Overall, our evaluation shows that there is no silver bullet against membership inference on aggregate locations. It is clear from our experiments that the adversary exploits several dimensions of the data, which makes the design of generic, robust defenses very challenging. Yet, for particular tasks, it may be possible to select a defense strategy that provides acceptable utility to analysts without being detrimental to privacy. However, in these cases, it is unlikely that the dataset will be useful for other tasks, let alone for general-purpose analytics.

Chapter 9

Conclusion

Collecting or releasing aggregate location information is often considered as a privacy-friendly strategy to support mobility analytics applications in the context of smart-cities [Shi+10; Pop+11; Waz18; Ube18; Tel18]. With this motivation in mind, in this thesis, we presented an end-to-end evaluation of crowdsourced privacy-friendly location aggregation aiming to understand its usefulness for analytics tasks as well as its privacy implications towards the users who contribute their data to the aggregation process. Our measurements focused on *aggregate location time-series*, indicating the number of people transiting in certain locations over time, using two real-world datasets capturing different mobility patterns in modern cities. Overall, the contributions of this dissertation can be summarized as follows:

- A time-series methodology, which, along with privacy-friendly crowdsourcing of aggregate locations, supports predictive mobility analytics such as forecasting and anomaly detection, in the context of modern cities.

- Quantification frameworks that allow reasoning about privacy loss for individual users stemming from the collection or release of aggregate location data against various adversarial goals such as profiling, localization, and membership inference.

- A comprehensive evaluation of defenses, ranging from generalization and hiding, to differential privacy, which can limit the information leakage from aggregate location time-series in terms of privacy protection and utility loss towards various analytics tasks.

9.1 Discussion

The results presented in this thesis have several real-world implications for location data practices and applications. First, we demonstrated that a range of mobility analytics that aim to improve the quality of life in modern cities can be supported via crowdsourced privacy-preserving aggregation introducing minimal computation overhead on users' devices (Chapter 5). This in turn implies that privacy-by-design solutions can be adopted by third-parties who are interested in deploying data collection services for the benefit of the society and while minimizing the exposure of users' location data.

Then, we highlighted that the end result of aggregation, i.e., the raw aggregate location time-series, leak information about users who contribute their data to the process. More specifically, in Chapters 6, 7, and 8, we showed that adversaries who have some prior information about users can exploit the aggregate information to infer users' profiles, locations, as well as their membership in the aggregates. This indicates that aggregation itself is a *weak* privacy protection mechanism and that additional defenses are required to protect users' end-to-end privacy in aggregate location crowdsourcing settings.

Our experiments with defenses that can be employed to thwart inferences on aggregate location time-series illustrated that there is a trade-off to be configured: the privacy of the users versus the utility of the aggregates towards the analytics tasks under consideration. Overall, we found that the setting of aggregate location statistics is very challenging as there is not a single defense that enables high quality analytics and simultaneously high privacy protection. Nonetheless, we are hopeful that our frameworks will be useful for researchers who are interested in designing and evaluating novel defenses for the setting, as well as providers who desire to assess the quality of privacy protection before releasing aggregate location data for analytics.

Furthermore, we believe that this thesis raises wider awareness about the privacy implications of aggregate location statistics. In particular, our results should be drawn to the attention of users who voluntarily share their location data with third-parties under the promise that it will only be used in anonymized and aggregate form. Similarly, entities whose business or services rely on aggregate location statistics should be careful about the privacy expectations they create to their users or customers, while legal groups should adjust their regulations regarding the appropriate ways of collecting and sharing aggregate location data.

Finally, while the privacy concerns highlighted in this thesis were centered around aggregate locations, they should be taken into account by researchers working on other types of aggregate time-series data. That is due to the fact that the evolution of a data value over time involves correlations, which can be potentially harmful for the data subjects' privacy. To this end, we are optimistic that the techniques and methods of this work will inspire the research community to pursue the privacy evaluation of aggregate statistics over time, in other data domains and applications, e.g., medical data, web search, network traffic, or smart metering.

9.2 Limitations

Like any research study, the work presented in this thesis is not without limitations. The mobility analytics presented in Chapter 5 are based on a relatively simple time-series methodology that takes into account seasonal patterns in the data. While this method allows us to prove that privacy-friendly crowdsourced aggregate location data is useful for mobility analytics in the context of modern cities, we are certain that more advanced models, e.g., based on deep learning, could be employed to improve the predictive analytics. In particular, as previous work has shown, the combination of aggregate location data with that capturing external factors such as weather conditions [ZZQ17] or social media trends [Pan+13], can further enhance forecasting and anomaly detection tasks.

A second limitation relates to the datasets used for the experimental evaluations. While the TFL and SFC datasets (presented in Chapter 4) represent different types of mobility patterns in modern cities, they are limited in size (number of users and time frame) and they only capture a specific case of location time-series, namely, transport data. Hence, it remains an open question if the privacy implications discovered in our evaluations hold for other types of location data, such as those emerging from mobile network events, social network check-ins, smartphone fitness applications, or combinations of these that can be used to build detailed user mobility profiles.

Next, the privacy quantification frameworks described in Chapters 6 and 7 assume adversaries with some prior knowledge about users' whereabouts, however, it would be appealing to examine weaker adversarial settings as well. Moreover, the attacks deployed within the frameworks are not *optimal*, thus, it would be interesting to investigate different ways to perform them as well as establish bounds on their effectiveness. To this end, we are hopeful that our framework designs are generic enough to be adapted and we encourage the research community to improve them towards stronger or different types of inferences with less restrictive prior knowledge assumptions.

Finally, the range of defenses that we evaluated in terms of privacy protection against inferences on aggregate location time-series and utility towards analytics tasks is somewhat limited. Although we quantified the privacy/utility tradeoffs achieved by generalization, hiding, and differential privacy, there exist other defenses that we did not investigate. For instance, we did not experiment with highly tuned differential privacy mechanisms based on spatial decompositions and clustering techniques, such as those proposed in [FXS13; Che+16]. Similarly, we did not analyze synthetic data generation techniques as they were not straightforward to adapt to the aggregate setting [He+15; BS16] or work only for specific analytics tasks (e.g., computing origin-destination commute distances [Mac+08] or modeling CDRs [Mir+13]). Further research in this context remains an interesting open problem.

9.3 Future Work

With the above limitations in mind, there are some interesting research directions that could be explored in the future. A straightforward extension of this thesis's work is to investigate the feasibility of privacy-friendly analytics tasks on aggregate location datasets of different nature, as well as considering other types of statistics. For instance, aggregate mobile network events could be used to generate origin-destination matrices and perform mobility analytics in the scale of a whole country, while aggregate location and activity statistics from fitness trackers could be employed to detect workout hotspots and promote health exercise challenges within smart-cities. Evaluating the privacy implications of aggregate location statistics in such applications would also be a worthwhile effort.

Another line of research could be the improvement of our privacy frameworks towards novel and stronger inferences on aggregate location statistics. For example, the adaptation of new kinds of adversarial prior knowledge, e.g., information about user co-locations [Olt+14], as well as the deployment of advanced statistical techniques such as Bayesian networks [Olt+17], could lead to more effective profiling or localization attacks (cf. Chapter 6). Similarly, using unsupervised learning techniques to perform membership inference attacks (cf. Chapters 7 and 8), or adapting the attacker's goal to infer membership about *groups* of users would also be valid research paths.

Next, an interesting research direction is related to the design of novel defenses that can thwart inferences on aggregate location statistics while achieving better utility for the analytics. To this end, a promising approach is the adaptation of highly specialized differential privacy frameworks that take into account data correlations [SWC17; Cao+17] to the setting of aggregate location time-series. Accordingly, adversarial training techniques [ZOP17; Gup+18] could be employed to generate aggregate location statistics the achieve optimal utility, while preserving privacy against specific adversaries [NSH18]. Evaluating the privacy/utility tradeoffs achieved by these techniques for the case of mobility analytics remains an open question.

Finally, another exciting research avenue would be the privacy evaluation of aggregate statistics that enable predictive analytics in other data domains and applications, e.g., medical data [Zha+14; Rai+17], web search [CMR15; Kor+09], network traffic [Bur+10; NMG17; ACG18], or smart metering [Bue+17; Lyu+18]. Similar to this dissertation's research, an end-to-end evaluation that demonstrates the usefulness of the aggregate statistics for machine learning and analytics tasks while quantifying privacy loss/gain from raw/protected aggregate data would be of paramount importance for our data-driven society.

Bibliography

[AC14] Gergely Acs and Claude Castelluccia. "A case study: Privacy preserving release of spatio-temporal density in paris". In: *Proceedings of the 20th International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. ACM. 2014, pp. 1679–1688.

[ACG18] Mohammad Alaggan, Mathieu Cunche, and Sébastien Gambs. "Privacy-preserving Wi-Fi analytics". In: *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS)*. De Gruyter Open, 2018, pp. 4–26.

[And+13] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. "Geo-indistinguishability: Differential privacy for location-based systems". In: *Proceedings of the 20th Conference on Computer and Communications Security (CCS)*. ACM. 2013, pp. 901–914.

[AP03] Andrew Ang and Monika Piazzesi. "A no-arbitrage vector autoregression of term structure dynamics with macroeconomic and latent variables". In: *Journal of Monetary Economics* 50.4 (2003), pp. 745–787.

[Bac+16] Michael Backes, Pascal Berrang, Mathias Humbert, and Praveen Manoharan. "Membership privacy in microRNA-based studies". In: *Proceedings of the 23rd Conference on Computer and Communications Security (CCS)*. ACM. 2016, pp. 319–330.

[Bam+08] Bhuvan Bamba, Ling Liu, Peter Pesti, and Ting Wang. "Supporting anonymous location queries in mobile environments with privacygrid". In: *Proceedings of the 17th International Conference on World Wide Web (WWW)*. ACM. 2008, pp. 237–246.

[Ben+09] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. "Pearson correlation coefficient". In: *Noise Reduction in Speech Processing*. Springer, 2009, pp. 1–4.

[Ber+12] Daniel J Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. "High-speed high-security signatures". In: *Journal of Cryptographic Engineering* 2.2 (2012), pp. 77–89.

[Ber15] Anna Berlee. *Using NYC taxi data to identify Muslim taxi drivers*. http://www.theiii.org/index.php/997/using-nyc-taxi-data-to-identify-muslim-taxi-drivers/. 2015.

[Bil+14] Igor Bilogrevic, Julien Freudiger, Emiliano De Cristofaro, and Ersin Uzun. "What's the gist? Privacy-preserving aggregation of user profiles". In: *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*. Springer. 2014, pp. 128–145.

[Boj+16] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. "End to end learning for self-driving cars". In: *arXiv preprint arXiv:1604.07316* (2016).

[BOT13] Joshua WS Brown, Olga Ohrimenko, and Roberto Tamassia. "Haze: Privacy-preserving real-time traffic statistics". In: *Proceedings of the 21st International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. ACM. 2013, pp. 540–543.

[Box+15] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[Bre01] Leo Breiman. "Random forests". In: *Journal of Machine Learning* 45.1 (2001), pp. 5–32.

[BS03] Alastair R Beresford and Frank Stajano. "Location privacy in pervasive computing". In: *Journal of Pervasive Computing* 1 (2003), pp. 46–55.

[BS15] Raef Bassily and Adam Smith. "Local, private, efficient protocols for succinct histograms". In: *Proceedings of the 47th Annual Symposium on Theory of Computing (STOC)*. ACM. 2015, pp. 127–135.

[BS16] Vincent Bindschaedler and Reza Shokri. "Synthesizing plausible privacy-preserving location traces". In: *Proceedings of the Symposium on Security and Privacy (S&P)*. IEEE. 2016, pp. 546–563.

[BT11] Javier A Barria and Suttipong Thajchayapong. "Detection and classification of traffic anomalies using microscopic traffic variables". In: *Journal of Transactions on Intelligent Transportation Systems* 12.3 (2011), pp. 695–704.

[Bue+17] Niklas Buescher, Spyros Boukoros, Stefan Bauregger, and Stefan Katzenbeisser. "Two s not enough: Privacy assessment of aggregation schemes in smart metering". In: *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS)*. De Gruyter Open, 2017, pp. 198–214.

[Bur+10] Martin Burkhart, Mario Strasser, Dilip Many, and Xenofontas A. Dimitropoulos. "SEPIA: Privacy-preserving aggregation of multi-fomain network events and statistics". In: *Proceedings of the 19th USENIX Security Symposium*. 2010, pp. 223–240.

[Cai+14] Xiang Cai, Rishab Nithyanand, Tao Wang, Rob Johnson, and Ian Goldberg. "A systematic approach to developing and evaluating website fingerprinting defenses". In: *Proceedings of the 21st Conference on Computer and Communications Security (CCS)*. ACM. 2014, pp. 227–238.

[Cao+16] Wei Cao, Zhengwei Wu, Dong Wang, Jian Li, and Haishan Wu. "Automatic user identification method across heterogeneous mobility data sources". In: *Proceedings of the 32nd International Conference on Data Engineering (ICDE)*. IEEE. 2016, pp. 978–989.

[Cao+17] Yang Cao, Masatoshi Yoshikawa, Yonghui Xiao, and Li Xiong. "Quantifying differential privacy under temporal correlations". In: *Proceedings of the 33rd International Conference on Data Engineering (ICDE)*. IEEE. 2017, pp. 821–832.

[CBK09] Varun Chandola, Arindam Banerjee, and Vipin Kumar. “Anomaly detection: A survey”. In: *Journal of Computing Surveys (CSUR)* 41.3 (2009), p. 15.

[CG09] Richard Chow and Philippe Golle. “Faking contextual data for fun, profit, and privacy”. In: *Proceedings of the 8th Annual Workshop on Privacy in the Electronic Society (WPES)*. ACM. 2009, pp. 105–108.

[CH67] Thomas Cover and Peter Hart. “Nearest neighbor pattern classification”. In: *Journal of Transactions on Information Theory* 13.1 (1967), pp. 21–27.

[Che+13] Rui Chen, Benjamin CM Fung, Noman Mohammed, Bipin C Desai, and Ke Wang. “Privacy-preserving trajectory data publishing by local suppression”. In: *Journal of Information Sciences* 231 (2013), pp. 83–97.

[Che+16] Rui Chen, Haoran Li, AK Qin, Shiva Prasad Kasiviswanathan, and Hongxia Jin. “Private spatial data aggregation in the local setting”. In: *Proceedings of the 32nd International Conference on Data Engineering (ICDE)*. IEEE. 2016, pp. 289–300.

[Cle+11] Maarten Clements, Pavel Serdyukov, Arjen P de Vries, and Marcel JT Reinders. “Personalised travel recommendation based on location co-occurrence”. In: *arXiv preprint arXiv:1106.5213* (2011).

[CM05] Graham Cormode and Shan Muthukrishnan. “An improved data stream summary: the count-min sketch and its applications”. In: *Journal of Algorithms* 55.1 (2005), pp. 58–75.

[CMR15] Aleksandr Chuklin, Ilya Markov, and Maarten de Rijke. “Click models for web search”. In: *Journal of Synthesis Lectures on Information Concepts, Retrieval, and Services* 7.3 (2015), pp. 1–115.

[Cor+12] Graham Cormode, Cecilia Procopiuc, Divesh Srivastava, Entong Shen, and Ting Yu. “Differentially private spatial decompositions”. In: *Proceedings of the 28th International Conference on Data Engineering (ICDE)*. IEEE. 2012, pp. 20–31.

[Cox58] David R Cox. “The regression analysis of binary sequences”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1958), pp. 215–242.

[CPS14] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. "A predictive differentially-private mechanism for mobility traces". In: *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS)*. Springer. 2014, pp. 21–41.

[CPS15] Konstantinos Chatzikokolakis, Catuscia Palamidessi, and Marco Stronati. "Constructing elastic distinguishability metrics for location privacy". In: *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS)*. De Gruyter Open, 2015, pp. 156–170.

[CSC12] Irina Ceapa, Chris Smith, and Licia Capra. "Avoiding the crowds: understanding tube station congestion patterns from trip data". In: *Proceedings of the International Workshop on Urban Computing (SIGKDD)*. ACM. 2012, pp. 134–141.

[CSS11] T-H Hubert Chan, Elaine Shi, and Dawn Song. "Private and continual release of statistics". In: *Journal of Transactions on Information and System Security (TISSEC)* 14.3 (2011), p. 26.

[CZH12] Sanjay Chawla, Yu Zheng, and Jiafeng Hu. "Inferring the root cause in road traffic anomalies". In: *Proceedings of the 12th International Conference on Data Mining (ICDM)*. IEEE. 2012, pp. 141–150.

[DF79] David A Dickey and Wayne A Fuller. "Distribution of the estimators for autoregressive time series with a unit root". In: *Journal of the American Statistical Association* 74.366a (1979), pp. 427–431.

[DFN05] Wenliang Du, Lei Fang, and P Ningi. "LAD: Localization anomaly detection for wireless sensor networks". In: *Proceedings of the 19th International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE. 2005, pp. 15–30.

[DM+08] Yoni De Mulder, George Danezis, Lejla Batina, and Bart Preneel. "Identification via location-profiling in GSM networks". In: *Proceedings of the 7th Annual Workshop on Privacy in the Electronic Society (WPES)*. ACM. 2008, pp. 23–32.

[DM+13] Yves-Alexandre De Montjoye, César A Hidalgo, Michel Verleysen, and Vincent D Blondel. "Unique in the crowd: The privacy bounds of human mobility". In: *Nature Scientific Reports* 3 (2013), p. 1376.

[DR+14] Cynthia Dwork, Aaron Roth, et al. “The algorithmic foundations of differential privacy”. In: *Journal of Foundations and Trends in Theoretical Computer Science* 9.3–4 (2014), pp. 211–407.

[Dwo+10] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N Rothblum. “Differential privacy under continual observation”. In: *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*. ACM. 2010, pp. 715–724.

[Dwo08] Cynthia Dwork. “Differential privacy: A survey of results”. In: *Proceedings of the International Conference on Theory and Applications of Models of Computation*. Springer. 2008, pp. 1–19.

[EPK14] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. “Rappor: Randomized aggregatable privacy-preserving ordinal response”. In: *Proceedings of the 21st Conference on Computer and Communications Security (CCS)*. ACM. 2014, pp. 1054–1067.

[ES03] Dominik Maria Endres and Johannes E Schindelin. “A new metric for probability distributions”. In: *Journal of Transactions on Information Theory* 49.7 (2003), pp. 1858–1860.

[Fan+08] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. “LIBLINEAR: A library for large linear classification”. In: *Journal of Machine Learning Research* 9 (2008), pp. 1871–1874.

[Fan+15] Zipei Fan, Xuan Song, Ryosuke Shibasaki, and Ryutaro Adachi. “CityMomentum: an online approach for crowd behavior prediction at a citywide level”. In: *Proceedings of the 2015 International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2015, pp. 559–569.

[FX12] Liyue Fan and Li Xiong. “Real-time aggregate monitoring with differential privacy”. In: *Proceedings of the 21st International Conference on Information and Knowledge Management (CIKM)*. ACM. 2012, pp. 2169–2173.

[FX13] Liyue Fan and Li Xiong. “Differentially private anomaly detection with a case study on epidemic outbreak detection”. In: *Proceedings of the 13th International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2013, pp. 833–840.

[FXS13] Liyue Fan, Li Xiong, and Vaidy Sunderam. "Differentially private multi-dimensional time series release for traffic monitoring". In: *Proceedings of the Annual IFIP Conference on Data and Applications Security and Privacy (DBSec)*. Springer. 2013, pp. 33–48.

[Gar+13] András Garzó, András A. Benczúr, Csaba István Sidló, Daniel Tahara, and Erik Francis Wyatt. "Real-time streaming mobility analytics". In: *Proceedings of the International Conference on Big Data*. IEEE, 2013, pp. 697–702.

[GC09] Thiago S Guzella and Walmir M Caminhas. "A review of machine learning approaches to spam filtering". In: *Journal of Expert Systems with Applications* 36.7 (2009), pp. 10206–10222.

[Geh+12] Johannes Gehrke, Michael Hay, Edward Lui, and Rafael Pass. "Crowd-blending privacy". In: *Advances in Cryptology (CRYPTO), 2012*. Springer, 2012, pp. 479–496.

[GF15] Marco Gramaglia and Marco Fiore. "Hiding mobile traffic fingerprints with glove". In: *Proceedings of the 11th Conference on Emerging Networking Experiments and Technologies (CoNEXT)*. ACM. 2015, pp. 26–39.

[GG03] Marco Gruteser and Dirk Grunwald. "Anonymous usage of location-based services through spatial and temporal cloaking". In: *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM. 2003, pp. 31–42.

[Ghi+08] Gabriel Ghinita, Panos Kalnis, Ali Khoshgozaran, Cyrus Shahabi, and Kian-Lee Tan. "Private queries in location based services: anonymizers are not necessary". In: *Proceedings of the International Conference on Management of Data (SIGMOD)*. ACM. 2008, pp. 121–132.

[GKPC14] Sébastien Gambs, Marc-Olivier Killijian, and Miguel Núñez del Prado Cortez. "De-anonymization attack on geolocated data". In: *Journal of Computer and System Sciences* 80.8 (2014), pp. 1597–1614.

[GL05] Bugra Gedik and Ling Liu. "Location privacy in mobile systems: A personalized anonymization model". In: *Proceedings of the 25th International Conference on Distributed Computing Systems (ICDS)*. IEEE. 2005, pp. 620–629.

[Gol07] Ian Goldberg. “Privacy-enhancing technologies for the internet III: ten years later”. In: *Digital Privacy*. Auerbach Publications, 2007, pp. 25–40.

[GP09] Philippe Golle and Kurt Partridge. “On the anonymity of home/work location pairs”. In: *Proceedings of the International Conference on Pervasive Computing*. Springer. 2009, pp. 390–397.

[Gra+17] Marco Gramaglia, Marco Fiore, Alberto Tarable, and Albert Banchs. “Preserving mobile subscriber privacy in open datasets of spatiotemporal trajectories”. In: *Proceedings of the International Conference on Computer Communications (INFOCOM)*. IEEE. 2017, pp. 1–9.

[Gre18] Andy Greenberg. *Apple’s differential privacy is about collecting your data - but not your data*. https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/. 2018.

[Gup+18] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. “Social GAN: Socially acceptable trajectories with generative adversarial networks”. In: *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. 2018.

[HAPC17] Briland Hitaj, Giuseppe Ateniese, and Fernando Perez-Cruz. “Deep models under the GAN: information leakage from collaborative deep learning”. In: *Proceedings of the 24th Conference on Computer and Communications Security (CCS)*. ACM. 2017, pp. 603–618.

[Hay+17] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. “LOGAN: evaluating privacy leakage of generative models using generative adversarial networks”. In: *arXiv preprint arXiv:1705.07663* (2017).

[He+15] Xi He, Graham Cormode, Ashwin Machanavajjhala, Cecilia M Procopiuc, and Divesh Srivastava. “DPT: differentially private trajectory synthesis using hierarchical reference systems”. In: *Proceedings of the VLDB Endowment* 8.11 (2015), pp. 1154–1165.

[Her+14] Michael Herrmann, Alfredo Rial, Claudia Diaz, and Bart Preneel. "Practical privacy-preserving location-sharing based services with aggregate statistics". In: *Proceedings of the 7th Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*. ACM, 2014, pp. 87–98.

[HHC14] Ren-Hung Hwang, Yu-Ling Hsueh, and Hao-Wei Chung. "A novel time-obfuscated algorithm for trajectory privacy protection". In: *Journal of Transactions on Services Computing (TSC)* 7.2 (2014), pp. 126–139.

[Hoh+07] Baik Hoh, Marco Gruteser, Hui Xiong, and Ansaf Alrabady. "Preserving privacy in GPS traces via uncertainty-aware path cloaking". In: *Proceedings of the 14th Conference on Computer and Communications Security (CCS)*. ACM. 2007, pp. 161–171.

[Hom+08] Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays". In: *PLoS Genetics* 4.8 (2008), e1000167.

[Hor+12] Eric J Horvitz, Johnson Apacible, Raman Sarin, and Lin Liao. "Prediction, expectation, and surprise: Methods, designs, and study of a deployed traffic forecasting service". In: *arXiv preprint arXiv:1207.1352* (2012).

[HR11] Shen-Shyang Ho and Shuhua Ruan. "Differential privacy for location pattern mining". In: *Proceedings of the 4th International Workshop on Security and Privacy in GIS and LBS (SIGSPATIAL)*. ACM. 2011, pp. 17–24.

[Hyl14] Svend Hylleberg. *Seasonality in regression*. Academic Press, 2014.

[HZS16] Minh X. Hoang, Yu Zheng, and Ambuj K. Singh. "FCCF: forecasting citywide crowd flows based on big data". In: *Proceedings of the 24th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. ACM. 2016, pp. 60–70.

[JK12] Marek Jawurek and Florian Kerschbaum. "Fault-tolerant privacy-preserving statistics". In: *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS)*. Springer. 2012, pp. 221–238.

[Jol02] Ian Jolliffe. *Principal Component Analysis*. Wiley & Sons, 2002.

[JW08] Iris A Junglas and Richard T Watson. "Location-based services". In: *Journal of Communications of the ACM* 51.3 (2008), pp. 65–69.

[Kat+12] Ryo Kato, Mayu Iwata, Takahiro Hara, Akiyoshi Suzuki, Xing Xie, Yuki Arase, and Shojiro Nishio. "A dummy-based anonymization method based on user trajectory with pauses". In: *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. ACM. 2012, pp. 249–258.

[KDK11] Klaus Kursawe, George Danezis, and Markulf Kohlweiss. "Privacy-friendly aggregation for the smart-grid". In: *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS)*. Springer. 2011, pp. 175–191.

[Kel+14] Georgios Kellaris, Stavros Papadopoulos, Xiaokui Xiao, and Dimitris Papadias. "Differentially private event sequences over infinite streams". In: *Proceedings of the VLDB Endowment* 7.12 (2014), pp. 1155–1166.

[Ken45] Maurice G Kendall. "The treatment of ties in ranking problems". In: *Journal of Biometrika* 33.3 (1945), pp. 239–251.

[KL51] Solomon Kullback and Richard A Leibler. "On information and sufficiency". In: *Journal of the Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86.

[KMM12] Christine Kopp, Michael Mock, and Michael May. "Privacy-preserving distributed monitoring of visit quantities". In: *Proceedings of the 20th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. ACM. 2012, pp. 438–441.

[Kor+09] Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. "Releasing search queries and clicks privately". In: *Proceedings of the 18th International Conference on World Wide Web (WWW)*. ACM. 2009, pp. 171–180.

[Kou+04] Yufeng Kou, Chang-Tien Lu, Sirirat Sirwongwattana, and Yo-Ping Huang. "Survey of fraud detection techniques". In: *International Conference on Networking, Sensing, and Control (ICNSC)*. Vol. 2. IEEE. 2004, pp. 749–754.

[Kru07] John Krumm. "Inference attacks on location tracks". In: *Proceedings of the 5th International Conference on Pervasive Computing*. Springer. 2007, pp. 127–143.

[Kru09] John Krumm. "Realistic driving trips for location privacy". In: *Proceedings of the 7th International Conference on Pervasive Computing*. Springer. 2009, pp. 25–41.

[KS09] Ali Khoshgozaran and Cyrus Shahabi. "Private information retrieval techniques for enabling location privacy in location-based services". In: *Privacy in Location-based Applications*. Springer, 2009, pp. 59–83.

[Kul14] Stefan Kulk. *Which celebrity is taking a taxi where? And what gentlemen's club are you visiting?* `http://www.theiii.org/index.php/316/which-celebrity-is-taking-a-taxi-where/`. 2014.

[KYS05] Hidetoshi Kido, Yutaka Yanagisawa, and Tetsuji Satoh. "Protection of location privacy using dummies for location-based services". In: *Proceedings of the 21st International Conference on Data Engineering Workshops*. IEEE. 2005, pp. 1248–1253.

[Lam+05] William HK Lam, KS Chan, Mei Lam Tam, and John WZ Shi. "Short-term travel time forecasts for transport information system in hong kong". In: *Journal of Advanced Transportation* 39.3 (2005), pp. 289–306.

[Laz+03] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. "A comparative study of anomaly detection schemes in network intrusion detection". In: *Proceedings of the International Conference on Data Mining (SDM)*. SIAM. 2003, pp. 25–36.

[Lyu+18] Lingjuan Lyu, Karthik Nandakumar, Benjamin Rubinstein, Jiong Jin, Justin Bedo, and Marimuthu Palaniswami. "PPFA: Privacy preserving fog-enabled aggregation in smart grid". In: *Journal of Transactions on Industrial Informatics* 14.8 (2018), pp. 3733–3744.

[Mac+06] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. "l-Diversity: Privacy beyond k-anonymity". In: *Proceedings of the 22nd International Conference on Data Engineering (ICDE)*. IEEE. 2006, pp. 24–36.

[Mac+08] Ashwin Machanavajjhala, Daniel Kifer, John Abowd, Johannes Gehrke, and Lars Vilhuber. "Privacy: Theory meets practice on the map". In: *Proceedings of the 24th International Conference on Data Engineering (ICDE)*. IEEE. 2008, pp. 277–286.

[Man+18] Dionysis Manousakas, Cecilia Mascolo, Alastair R Beresford, Dennis Chan, and Nikhil Sharma. "Quantifying privacy loss of human mobility graph topology". In: *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS)*. De Gruyter Open, 2018, pp. 5–21.

[Mar+05] Giannis F Marias, Constantinos Delakouridis, Leonidas Kazatzopoulos, and Panagiotis Georgiadis. "Location privacy through secret sharing techniques". In: *Proceedings of the International Symposium on a World of Wireless Mobile and Multimedia Networks*. IEEE. 2005, pp. 614–620.

[Mar16] Bernard Marr. *21 scary things big data knows about you*. https://www.forbes.com/sites/bernardmarr/2016/03/08/21-scary-things-big-data-knows-about-you/. 2016.

[MDD16] Luca Melis, George Danezis, and Emiliano De Cristofaro. "Efficient private statistics with succinct sketches". In: *Proceedings of the 23rd Annual Network & Distributed System Security Symposium (NDSS)*. Internet Society. 2016.

[Mel+18] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. "Inference Attacks Against Collaborative Learning". In: *arXiv preprint arXiv:1805.04049* (2018).

[Mir+13] Darakhshan J Mir, Sibren Isaacman, Ramón Cáceres, Margaret Martonosi, and Rebecca N Wright. "Dp-where: Differentially private modeling of human mobility". In: *Proceedings of the International Conference on Big Data*. IEEE. 2013, pp. 580–588.

[MRC09] Joseph Meyerowitz and Romit Roy Choudhury. "Hiding stars with fireworks: location privacy through camouflage". In: *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM. 2009, pp. 345–356.

[Nai+16] Farid Movahedi Naini, Jayakrishnan Unnikrishnan, Patrick Thiran, and Martin Vetterli. "Where you are is who you are: User identification by matching statistics." In: *Journal of Transactions on Information Forensics and Security* 11.2 (2016), pp. 358–372.

[NMG17] Oana-Georgiana Niculaescu, Mihai Maruseac, and Gabriel Ghinita. "Differentially-private big data analytics for high-speed research network traffic measurement". In: *Proceedings of the 7th Conference on Data and Application Security and Privacy (CODASPY)*. ACM. 2017, pp. 151–153.

[NSH18] Milad Nasr, Reza Shokri, and Amir Houmansadr. "Machine Learning with Membership Privacy using Adversarial Regularization". In: *Proceedings of the 25th Conference on Computer and Communications Security (CCS)*. ACM. 2018.

[Olt+14] Alexandra-Mihaela Olteanu, Kévin Huguenin, Reza Shokri, and Jean-Pierre Hubaux. "Quantifying the effect of co-location information on location privacy". In: *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS)*. Springer. 2014, pp. 184–203.

[Olt+17] Alexandra-Mihaela Olteanu, Kévin Huguenin, Reza Shokri, Mathias Humbert, and Jean-Pierre Hubaux. "Quantifying interdependent privacy risks with location data". In: *Journal of Transactions on Mobile Computing* 16.3 (2017), pp. 829–842.

[Olu+10] Femi Olumofin, Piotr K Tysowski, Ian Goldberg, and Urs Hengartner. "Achieving efficient query privacy for location based services". In: *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS)*. Springer. 2010, pp. 93–110.

[OTPG17] Simon Oya, Carmela Troncoso, and Fernando Pérez-González. "Is geo-indistinguishability what you are looking for?" In: *Proceedings of the 16th Annual Workshop on Privacy in the Electronic Society (WPES)*. ACM. 2017, pp. 137–140.

[Pan+13] Bei Pan, Yu Zheng, David Wilkie, and Cyrus Shahabi. "Crowd sensing of traffic anomalies based on human mobility and social media". In: *Proceedings of the 21st International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. ACM. 2013, pp. 344–353.

[Pan17] DJ Pangburn. *Even this data guru is creeped out by what anonymous location data reveals about us*. `https://www.fastcompany.com/3068846/how-your-location-data-identifies-you-gilad-lotan-privacy`. 2017.

[PBP10] Stavros Papadopoulos, Spiridon Bakiras, and Dimitris Papadias. "Nearest neighbor search with strong location privacy". In: *Proceedings of the VLDB Endowment* 3.1-2 (2010), pp. 619–629.

[PDR16] Apostolos Pyrgelis, Emiliano De Cristofaro, and Gordon J Ross. "Privacy-friendly mobility analytics using aggregate location data". In: *Proceedings of the 24th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. ACM. 2016, pp. 34–44.

[Pop+11] Raluca Ada Popa, Andrew J Blumberg, Hari Balakrishnan, and Frank H Li. "Privacy and accountability for location-based aggregate statistics". In: *Proceedings of the 18th Conference on Computer and Communications Security (CCS)*. ACM. 2011, pp. 653–666.

[Pri+14] Vincent Primault, Sonia Ben Mokhtar, Cédric Lauradoux, and Lionel Brunie. "Differentially private location privacy in practice". In: *arXiv preprint arXiv:1410.7744* (2014).

[Pri+15] Vincent Primault, Sonia Ben Mokhtar, Cédric Lauradoux, and Lionel Brunie. "Time distortion anonymization for the publication of mobility data with high utility". In: *Proceedings of the International Conference on Big Data Science and Engineering*. Vol. 1. IEEE. 2015, pp. 539–546.

[PSDG09] Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. *CRAWDAD dataset epfl/mobility (v. 2009-02-24)*. Downloaded from `https://crawdad.org/epfl/mobility/20090224`. Feb. 2009. DOI: `10.15783/C7J010`.

[PTD17] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. "What does the crowd say about you? Evaluating aggregation-based location privacy". In: *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS)*. De Gruyter Open. 2017, pp. 156–176.

[PTD18] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. "Knock knock, who's there? Membership inference on aggregate location data". In: *Proceedings of the 25th Annual Network & Distributed System Security Symposium (NDSS)*. Internet Society. 2018.

[Que+11] Daniele Quercia, Ilias Leontiadis, Liam McNamara, Cecilia Mascolo, and Jon Crowcroft. "Spotme if you can: Randomized responses for location obfuscation on mobile phones". In: *Proceedings of the 31st International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2011, pp. 363–372.

[QYL13] Wahbeh Qardaji, Weining Yang, and Ninghui Li. "Differentially private grids for geospatial data". In: *Proceedings of the 29th International Conference on Data Engineering (ICDE)*. IEEE. 2013, pp. 757–768.

[Rai+17] Jean Louis Raisaro, Juan Ramon Troncoso-Pastoriza, Mickael Misbach, E Sousa Gomes de Sa, Joao Andre, Sylvain Pradervand, Edoardo Missiaglia, Olivier Michielin, Bryan Alexander Ford, and Jean-Pierre Hubaux. "MedCo: Enabling privacy-conscious exploration of distributed clinical and genomic data". In: *4th Internation Workshop on Genome Privacy and Security (GenoPri)*. 2017.

[RHW85] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. *Learning internal representations by error propagation*. Tech. rep. California University San Diego La Jolla Institute for Cognitive Science, 1985.

[RM14] Luca Rossi and Mirco Musolesi. "It's the way you check-in: identifying users in location-based social networks". In: *Proceedings of the 2nd Conference on Online Social Networks*. ACM. 2014, pp. 215–226.

[RN10] Vibhor Rastogi and Suman Nath. "Differentially private aggregation of distributed time-series with transformation and encryption". In: *Proceedings of the International Conference on Management of Data (SIGMOD)*. ACM. 2010, pp. 735–746.

[RWM15] Luca Rossi, James Walker, and Mirco Musolesi. "Spatio-temporal techniques for user identification by means of GPS mobility data". In: *EPJ Data Science* 4.1 (2015), p. 11.

[Sab+15] Mohammad Sabokrou, Mahmood Fathy, Mojtaba Hoseini, and Reinhard Klette. "Real-time anomaly detection and localization in crowded scenes". In: *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops*. IEEE. 2015, pp. 56–62.

[Sal+18] Ahmed Salem, Yang Zhang, Mathias Humbert, Mario Fritz, and Michael Backes. "ML-leaks: Model and data independent membership inference attacks and defenses on machine learning models". In: *arXiv preprint arXiv:1806.01246* (2018).

[Sen+18] Hansi Senaratne, Manuel Mueller, Michael Behrisch, Felipe Lalanne, Javier Bustos-Jiménez, Jörn Schneidewind, Daniel Keim, and Tobias Schreck. "Urban mobility analysis with mobile network data: A visual analytics approach". In: *Journal of Transactions on Intelligent Transportation Systems* 19.5 (2018), pp. 1537–1546.

[SGI09] Pravin Shankar, Vinod Ganapathy, and Liviu Iftode. "Privately querying location-based services with sybilquery". In: *Proceedings of the 11th International Conference on Ubiquitous Computing*. ACM. 2009, pp. 31–40.

[SH12] Mudhakar Srivatsa and Mike Hicks. "Deanonymizing mobility traces: Using social network as a side-channel". In: *Proceedings of the 19th Conference on Computer and Communications Security (CCS)*. ACM. 2012, pp. 628–637.

[Shi+10] Jing Shi, Rui Zhang, Yunzhong Liu, and Yanchao Zhang. "Prisense: privacy-preserving data aggregation in people-centric urban systems". In: *Proceedings of the International Conference on Computer Communications (INFOCOM)*. IEEE. 2010, pp. 1–9.

[Shi+11] Elaine Shi, HTH Chan, Eleanor Rieffel, Richard Chow, and Dawn Song. "Privacy-preserving aggregation of time-series data". In: *Proceedings of the 18th Network & Distributed System Security Symposium (NDSS)*. Internet Society. 2011.

[Sho+10] Reza Shokri, Carmela Troncoso, Claudia Diaz, Julien Freudiger, and Jean-Pierre Hubaux. "Unraveling an old cloak: k-anonymity for location privacy". In: *Proceedings of the 9th Annual Workshop on Privacy in the Electronic Society (WPES)*. ACM. 2010, pp. 115–118.

[Sho+11a] Reza Shokri, George Theodorakopoulos, Jean-Yves Le Boudec, and Jean-Pierre Hubaux. "Quantifying location privacy". In: *Proceedings of the Symposium on Security and Privacy (S&P)*. IEEE. 2011, pp. 247–262.

[Sho+11b] Reza Shokri, George Theodorakopoulos, George Danezis, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. "Quantifying location privacy: the case of sporadic location exposure". In: *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS)*. Springer. 2011, pp. 57–76.

[Sho+17] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. "Membership inference attacks against machine learning models". In: *Proceedings of the Symposium on Security and Privacy (S&P)*. IEEE. 2017, pp. 3–18.

[SK03] Anthony Stathopoulos and Matthew G Karlaftis. "A multivariate state space approach for urban traffic flow modeling and prediction". In: *Journal of Transportation Research Part C: Emerging Technologies* 11.2 (2003), pp. 121–135.

[SKA15] Ricardo Silva, Soong Moon Kang, and Edoardo M Airoldi. "Predicting traffic volumes and estimating the effects of shocks in massive transportation systems". In: *Proceedings of the National Academy of Sciences* 112.18 (2015), pp. 5643–5648.

[SLB12] Günther Sagl, Martin Loidl, and Euro Beinat. "A visual analytics approach for extracting spatio-temporal urban mobility information from mobile network traffic". In: *ISPRS International Journal of Geo-Information* 1.3 (2012), pp. 256–271.

[Sun+04] Bo Sun, Fei Yu, Kui Wu, and Victor Leung. "Mobility-based anomaly detection in cellular mobile networks". In: *Proceedings of the 3rd Workshop on Wireless Security*. ACM. 2004, pp. 61–69.

[SWC17] Shuang Song, Yizhen Wang, and Kamalika Chaudhuri. "Pufferfish privacy mechanisms for correlated data". In: *Proceedings of the International Conference on Management of Data (SIGMOD)*. ACM. 2017, pp. 1291–1306.

[Swe02] Latanya Sweeney. "k-anonymity: A model for protecting privacy". In: *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems* 10.05 (2002), pp. 557–570.

[Tan+17] Jun Tang, Aleksandra Korolova, Xiaolong Bai, Xueqiang Wang, and Xiaofeng Wang. "Privacy loss in Apple's implementation of differential privacy on macOS 10.12". In: *arXiv preprint arXiv:1709.02753* (2017).

[Tho+12] Dennis Thom, Harald Bosch, Steffen Koch, Michael Wörner, and Thomas Ertl. "Spatiotemporal anomaly detection through visual analysis of geolocated twitter messages". In: *Proceedings of the Pacific Visualization Symposium (PacificVis)*. IEEE. 2012, pp. 41–48.

[TM08] Manolis Terrovitis and Nikos Mamoulis. "Privacy preservation in the publication of trajectories". In: *Proceedings of the 9th International Conference on Mobile Data Management (MDM)*. IEEE. 2008, pp. 65–72.

[TNS16] Hien To, Kien Nguyen, and Cyrus Shahabi. "Differentially private publication of location entropy". In: *Proceedings of the 24th International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. ACM. 2016, pp. 35–45.

[Ven+18] Giridhari Venkatadri, Athanasios Andreou, Yabing Liu, Alan Mislove, Krishna P Gummadi, Patrick Loiseau, and Oana Goga. "Privacy risks with Facebook's PII-based targeting: Auditing a data broker's advertising interface". In: *Proceedings of the Symposium on Security and Privacy (S&P)*. IEEE. 2018.

[Wan+09] Rui Wang, Yong Fuga Li, XiaoFeng Wang, Haixu Tang, and Xiaoyong Zhou. "Learning your identity and disease from research papers: information leaks in genome wide association study". In: *Proceedings of the 16th Conference on Computer and Communications Security (CCS)*. ACM. 2009, pp. 534–544.

[Wan+16] Qian Wang, Yan Zhang, Xiao Lu, Zhibo Wang, Zhan Qin, and Kui Ren. "RescueDP: Real-time spatio-temporal crowd-sourced data publishing with differential privacy". In: *Proceedings of the International Conference on Computer Communications (INFOCOM)*. IEEE. 2016, pp. 1–9.

[Wan+18] Huandong Wang, Chen Gao, Yong Li, Gang Wang, Depeng Jin, and Jingbo Sun. "De-anonymization of mobility trajectories: Dissecting the gaps between theory and practice". In: *Proceedings of the 25th Annual Network & Distributed System Security Symposium (NDSS)*. Internet Society. 2018.

[War65] Stanley L Warner. "Randomized response: A survey technique for eliminating evasive answer bias". In: *Journal of the American Statistical Association* (1965), pp. 63–69.

[Waz18] Waze. https://www.waze.com. 2018.

[WDR12] Marius Wernke, Frank Dürr, and Kurt Rothermel. "PShare: Position sharing for location privacy based on multi-secret sharing". In: *Proceedings of the International Conference on Pervasive Computing and Communications (PerCom)*. IEEE. 2012, pp. 153–161.

[Wei+12] Jeremy C Weiss, Sriraam Natarajan, Peggy L Peissig, Catherine A McCarty, and David Page. "Machine learning for personalized medicine: Predicting primary myocardial infarction from electronic health records". In: *AI Magazine* 33.4 (2012), p. 33.

[Wer+14] Marius Wernke, Pavel Skvortsov, Frank Dürr, and Kurt Rothermel. "A classification of location privacy attacks and approaches". In: *Journal of Personal and Ubiquitous Computing* 18.1 (2014), pp. 163–175.

[WM03] Arnold D Well and Jerome L Myers. *Research design & statistical analysis*. Psychology Press, 2003.

[WN16] Atsushi Waseda and Ryo Nojima. "Analyzing randomized response mechanisms under differential privacy". In: *Proceedings of the International Conference on Information Security (ISC)*. Springer. 2016, pp. 271–282.

[WWY15] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. "Collaborative deep learning for recommender systems". In: *Proceedings of the 21st International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. ACM. 2015, pp. 1235–1244.

[XKP09] Mingqiang Xue, Panos Kalnis, and Hung Keng Pung. "Location diversity: Enhanced privacy protection in location based services". In: *Proceedings of the International Symposium on Location and Context Awareness (LoCA)*. Springer. 2009, pp. 70–87.

[Xu+17] Fengli Xu, Zhen Tu, Yong Li, Pengyu Zhang, Xiaoming Fu, and Depeng 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 (WWW)*. 2017, pp. 1241–1250.

[Yav+05] Gökhan Yavaş, Dimitrios Katsaros, Özgür Ulusoy, and Yannis Manolopoulos. "A data mining approach for location prediction in mobile environments". In: *Journal of Data & Knowledge Engineering* 54.2 (2005), pp. 121–146.

[YLP17] Lei Yu, Ling Liu, and Calton Pu. "Dynamic differential location privacy with personalized error bounds". In: *Proceedings of the 24th Network & Distributed System Security Symposium (NDSS)*. Internet Society. 2017.

[ZB11] Hui Zang and Jean Bolot. "Anonymization of location data does not work: A large-scale measurement study". In: *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom)*. ACM. 2011, pp. 145–156.

[Zha+14] Kuan Zhang, Xiaohui Liang, Mrinmoy Baura, Rongxing Lu, and Xuemin Sherman Shen. "PHDA: A priority based health data aggregation with privacy preservation for cloud assisted WBANs". In: *Journal of Information Sciences* 284 (2014), pp. 130–141.

[Zho+16] Chen Zhong, Michael Batty, Ed Manley, Jiaqiu Wang, Zijia Wang, Feng Chen, and Gerhard Schmitt. "Variability in regularity: Mining temporal mobility patterns in London, Singapore and Beijing using smart-card data". In: *PloS one* 11.2 (2016).

[ZN12] Jiangchuan Zheng and Lionel M Ni. "An unsupervised framework for sensing individual and cluster behavior patterns from human mobile data". In: *Proceedings of the 14th International Conference on Ubiquitous Computing (UbiComp)*. ACM. 2012, pp. 153–162.

[ZOP17] Chaoyun Zhang, Xi Ouyang, and Paul Patras. "Zipnet-gan: Inferring fine-grained mobile traffic patterns via a generative adversarial neural network". In: *Proceedings of the 13th International Conference on Emerging Networking Experiments and Technologies (CoNEXT)*. ACM. 2017, pp. 363–375.

[ZZQ17] Junbo Zhang, Yu Zheng, and Dekang Qi. "Deep spatio-temporal residual networks for citywide crowd flows prediction". In: *Proceedings of the 31st Conference on Artificial Intelligence (AAAI)*. 2017, pp. 1655–1661.

[ZZY15] Yu Zheng, Huichu Zhang, and Yong Yu. "Detecting collective anomalies from multiple spatio-temporal datasets across different domains". In: *Proceedings of the 23rd International Conference on Advances in Geographic Information Systems (SIGSPATIAL)*. ACM. 2015, pp. 2–12.

[Cit18] Citymapper. *Smart Ride*. https://citymapper.com/smartride. 2018.

[Eur18] European Union. *General Data Protection Regulation*. https://www.eugdpr.org/. 2018.

[Pow18] Power Tutor. *A power monitor for Android-based mobile platforms*. http://ziyang.eecs.umich.edu/projects/powertutor/. 2018.

[Tel18] Telefonica Smart Steps. https://www.business-solutions.telefonica.com/en/enterprise/solutions/smarter-selling/big-data-insights/. 2018.

[Tra16] Transport For London. *Budget report 2015-2016*. http://content.tfl.gov.uk/board-20150326-part-1-item08-tfl-budget-2015-16.pdf. 2016.

[Tra18] Transport For London. *Travel alerts*. https://twitter.com/tfltravelalerts?lang=el. 2018.

[Ube18] Uber Movement. https://movement.uber.com/. 2018.

Appendix A

Privacy-Preserving Data Aggregation

For clarity, we describe the privacy-preserving data aggregation protocol, proposed by Melis et al. [MDD16], which we employ in Chapter 5 to enable crowdsourced mobility analytics on aggregate locations in a privacy-friendly way.

The Privacy-Preserving Data Aggregation Protocol by Melis et al. [MDD16]

While there exist a number of privacy-preserving aggregation protocols in the literature (see Section 3.2), we decided to utilize that proposed by Melis et al. [MDD16] as it guarantees: scalability, independence from trusted third-parties or key distribution centers, and fault tolerance. Scalability is achieved by combining the private aggregation protocol of Kursawe et al. [KDK11] (secure under the Computational Diffie Hellman assumption in the presence of *honest but curious* adversaries) with data structures supporting succinct data representation, i.e., Count-Min Sketches [CM05]. The latter introduce a small, upper-bounded error in the aggregates, but reduce the computational and communication complexities of the cryptographic operations from linear to logarithmic in the size of the input. It also features a completely distributed key generation, which, unlike other protocols, e.g. [JK12; Pop+11], does not require any other authorities. Finally, its fault tolerance property addresses one of the main limitations of [KDK11], i.e., if one or more users fail to report their (encrypted) data, the aggregator cannot correctly decrypt the aggregates (since it relies on encryption keys summing up to zero).

More precisely, the protocol proposed by Melis et al. [MDD16] consists of the following four phases:

1. **Setup:** Assuming a cyclic group $\mathbb{G}$ of prime order q for which the Computational Diffie-Hellman problem is hard, and g a generator of this group, each user $u \in U$ generates a private key $x_u \in_r \mathbb{G}$ (i.e., sampled at random from G) and a public key $y_u = g^{x_u} \bmod q$. The public keys are published with the aggregator.

2. **Encryption:** Each user $u \in U$ holds an input vector $\mathcal{L}_u$ which contains $|S|$ elements, representing the locations she visited at a specific time slot. To participate in the privacy-preserving aggregation each user needs to generate *blinding factors* based on the public keys of the other users in such a way that they all sum up to zero. At time $t \in T$, for each location $s \in \{1, \cdots, |S|\}$, user u computes $b_{us} = \sum_{j=1, j \neq u}^{|U|} H(y_j^{x_u} \| s \| t) \cdot (-1)^{u>j} \bmod q$, where H is a cryptographic hash function, and $\|$ denotes the concatenation operator. Then, u encrypts each entry $\ell_{us} \in \mathcal{L}_u$, as $\ell'_{us} = \ell_{us} + b_{us} \bmod 2^{32}$, and sends the resulting ciphertexts to the aggregator.

3. **Aggregation:** The aggregator collects the ciphertexts from each user $u \in U$ and (obliviously) aggregates them. More precisely, for each location $s \in S$ it computes $A_s = \sum_{u=1}^{|U|} \ell'_{us} = \sum_{u=1}^{|U|} \ell_{us} + \sum_{u=1}^{|U|} b_{us} = \sum_{u=1}^{|U|} \ell_{us} \bmod 2^{32}$, where A_s denotes the s-th item of vector A, i.e., the number of users who were in location s, at time t.

4. **Fault recovery:** If, during the aggregation phase, only a subset of users U_{on} successfully submits its data, the aggregator communicates the subset U_{on} to its users. Subsequently, each $u' \in U_{on}$ computes, for each location $s \in S$, $b'_{u's} = \sum_{j=1, j \neq u', j \notin U_{on}}^{|U|} H(y_j^{x_{u'}} \| s \| t) \cdot (-1)^{u'>j} \bmod q$. Then each user $u' \in U_{on}$ sends these values back to the aggregator who can now obtain the aggregate counts of each location $s \in S$ by computing $A_s^{'} = (\sum_{u' \in U_{on}} \ell'_{u's} - \sum_{u' \in U_{on}} b'_{u's}) \bmod 2^{32}$.

Groups. Furthermore, another feature of [MDD16] is the ability to dynamically allocate users in groups, perform within-group aggregation, and then combine the aggregate statistics from multiple groups. This is crucial if one desires to cope with a dynamic and mobile setting, similar to that considered in Chapter 5 (i.e., crowdsourcing aggregate location data). This feature also allows to bound the complexity of the encryption phase, which depends on the number of users in the groups over which the aggregate statistics are computed.

Input compression. Finally, the authors of [MDD16] use Count-Min Sketches to guarantee scalability when the input vector ($\mathcal{L}$) is large. Specifically, the encryption phase is modified as follows: each user $u \in U$ initializes a Count-Min Sketch vector $C_u \in \mathbb{N}^{c \times d}$ with zero entries, then encodes her original input vector $\mathcal{L}_u$ using the update procedure of Count-Min Sketches [CM05] while employing the following pairwise hash function: $h(x) = ((a \cdot x + b) \bmod p') \bmod d$ for $a \neq 0, b$ random integers modulo a random prime p'. Then, each user encrypts C_u as in the previously described encryption phase. For more details regarding the configuration of the parameters of the Count-Min Sketch, a reader is advised to consult the paper of Melis et al. [MDD16].

Appendix B

Additional Experimental Results for Chapter 6

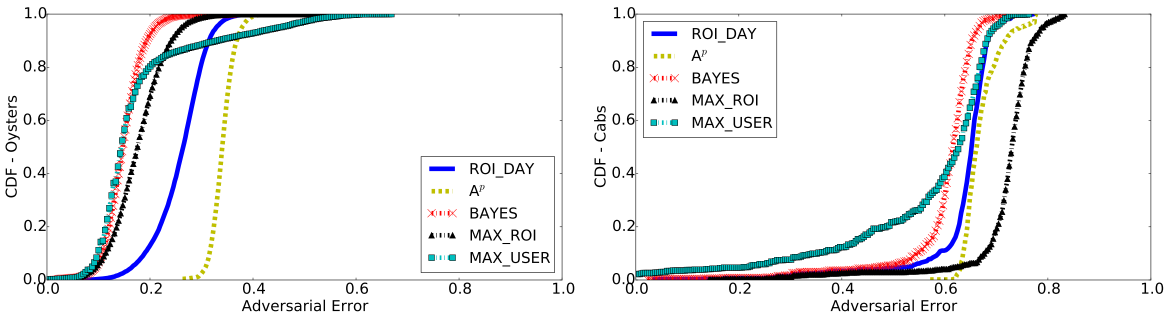
We report additional experimental results related to Section 6.2 of Chapter 6.

B.1 User Profiling

Probabilistic Priors

ROI_DAY. Recall that, with ROI_DAY, Adv knows for the users, a profile for each hour of any day (e.g., user's frequent locations at 4pm). For TFL (Fig. B.1a), we observe that this is a more instructive prior than commuters' frequent ROIs (FREQ_ROI), with an average prior error of 0.25. Moreover, we note that BAYES and MAX_ROI inferences remarkably improve Adv's profiling accomplishment for all users, yielding 0.41 and 0.31 average privacy loss, respectively. MAX_USER improves Adv's predictions for $\sim$ 80% of the users and achieves 0.37 average loss in privacy. Similarly for SFC (Fig. B.1b), ROI_DAY (0.63 avg. error) is a more revealing prior knowledge than cabs' frequent ROIs (FREQ_ROI – 0.65) for the adversary. BAYES and MAX_USER give advantage to Adv in profiling users (resulting in 0.06 and 0.13 privacy loss, resp.) while MAX_ROI does not, once again, indicating the bias of this strategy towards less active cabs.

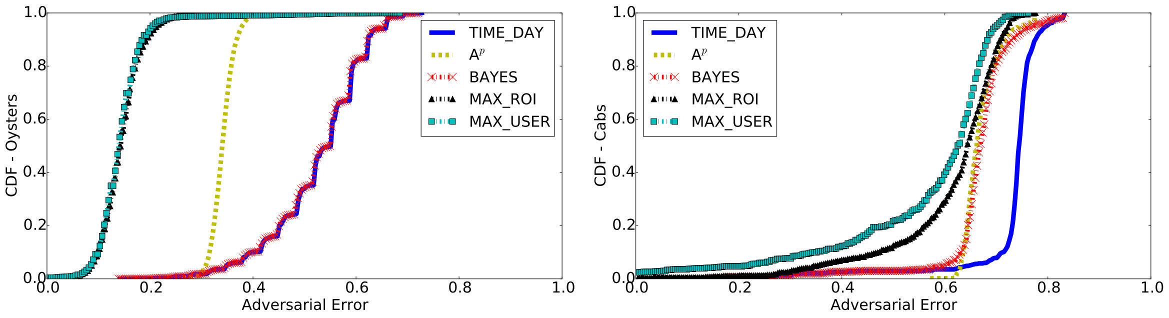
TIME_DAY. Figure B.2a plots the CDF of Adv's total error in profiling TFL commuters, with the TIME_DAY prior knowledge, i.e., a time profile indicating which hours of day a user is likely to report ROIs. We observe that Adv's performance is worse (0.52 mean



(a) TFL

(b) SFC

FIGURE B.1: ROI_DAY prior – Adv's profiling error.



(a) TFL

(b) SFC

FIGURE B.2: TIME_DAY prior – Adv's profiling error.

error) compared to priors containing location information (i.e., FREQ_ROI, ROI_DAY, and ROI_DAY_WEEK). This is expected, since the TIME_DAY prior contains only time information for the users, and it is a uniform distribution over the ROIs, for the time slots that they are likely to be in the transportation system. Indeed, Fig. B.2a shows that profiling only with the aggregate profile (A^p), Adv achieves smaller error (0.34). Among the inference strategies, we note that BAYES negligibly improves Adv's error in profiling users due to the very small prior probabilities. MAX_ROI and MAX_USER exhibit similar performance, as in both cases the users who are more likely to be inside the system, are selected to cover the aggregate values (in this case both strategies pick users based on their total number of ROIs). With these strategies, Adv's performance increases significantly and there is notable privacy loss for the users (0.72 on average).

On the SFC dataset (Fig. B.2b), we observe that when Adv knows the cabs' most frequent time slots of day (TIME_DAY), she obtains a worse prior (0.73 mean error) compared to cabs'

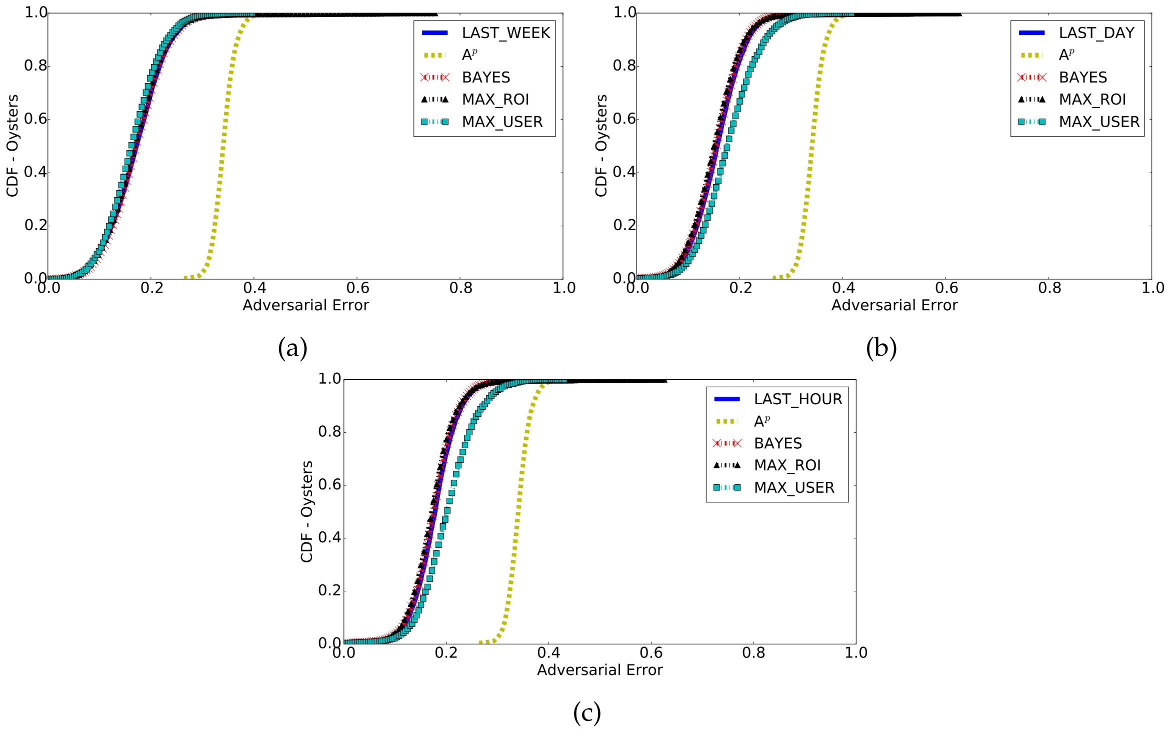
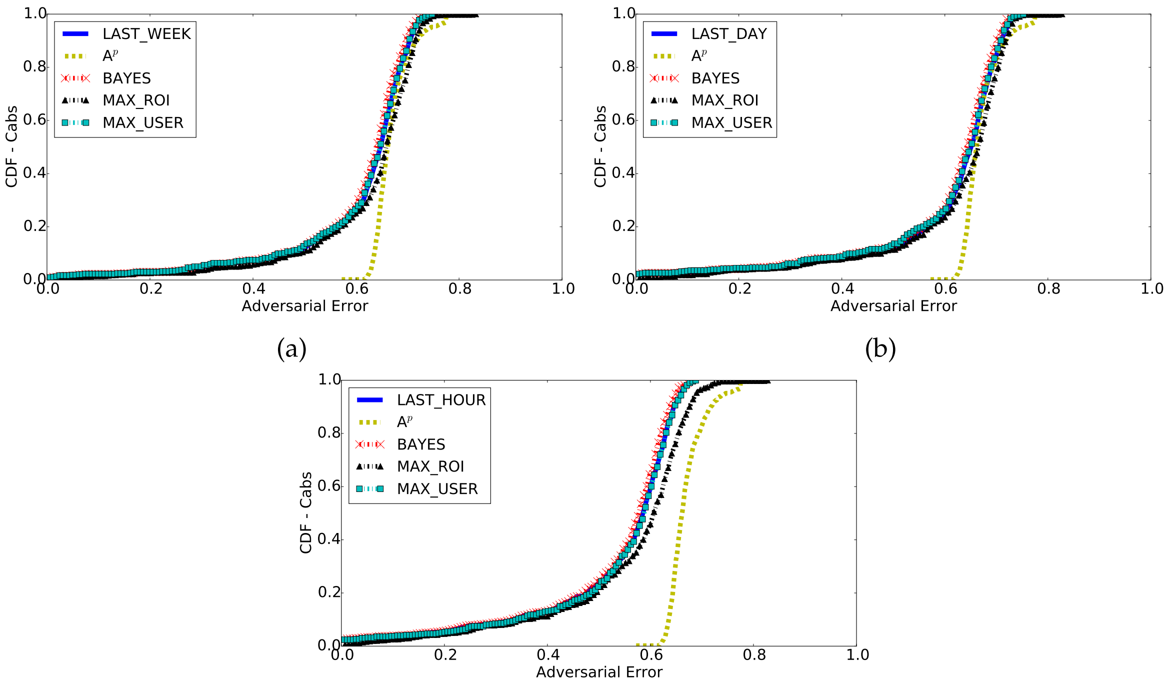


FIGURE B.3: LAST_WEEK (a), LAST_DAY (b), and LAST_HOUR (c), priors – Adv's profiling error on the TFL dataset.

most frequent ROIs (FREQ_ROI – 0.65) or cabs' most frequent ROIs with time and day semantics (ROI_DAY_WEEK – 0.61). Unlike TFL, Bayesian inference yields a 0.1 privacy loss, while the greedy strategies perform even better. More precisely, with MAX_ROI the mean privacy loss is 0.16 and with MAX_USER 0.22.

Assignment Priors

Figures B.3 and B.4 plot the results that are briefly discussed in Section 6.2.2. More precisely, with TFL, when Adv knows users' last week's whereabouts (LAST_WEEK), her baseline mean error is 0.17, indicating that commuters are fairly regular in their weekly patterns. BAYES, MAX_ROI, and MAX_USER, somewhat reduce Adv's error and achieve only little privacy loss (0.01, 0.03, and 0.05, resp.). When the users' last day's ROIs are available to Adv (LAST_DAY), her initial error is comparable to LAST_WEEK but smaller (0.15 on average). BAYES and MAX_ROI only slightly reduce the adversarial error, causing, resp., 0.02 and 0.05 privacy loss. On the contrary, MAX_USER does not harm commuters' privacy as it actually increases Adv's error, indicating that the most mobile users might not follow the patterns of



(a)

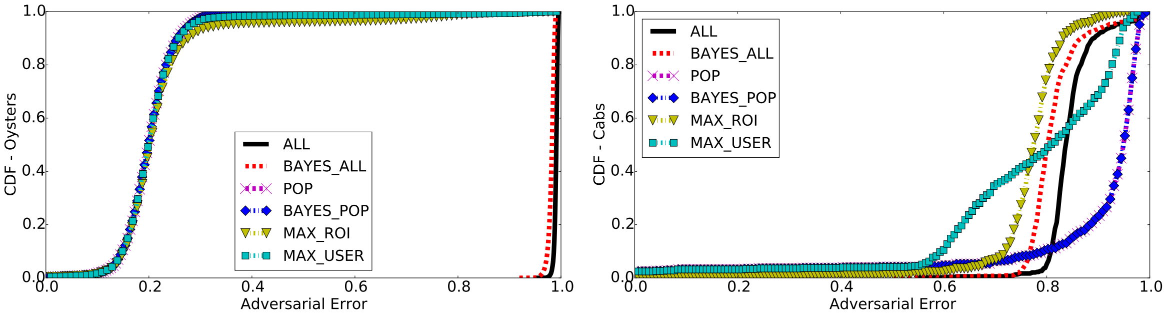
(b)

(c)

FIGURE B.4: LAST_WEEK (a), LAST_DAY (b), and LAST_HOUR (c), priors – Adv's profiling error on the SFC dataset.

their previous day. LAST_HOUR yields larger error compared to the previous ones (0.19), as passengers do not exhibit as strong hourly seasonality. Once again, all inferences yield negligible privacy loss. In general for TFL, we remark that seasonal historic profiles are more instructive priors than probabilistic ones (e.g., FREQ_ROI or TIME_DAY_WEEK), thus, the privacy loss for individuals from the aggregate time-series is actually small compared to that of probabilistic priors.

Our experiments on the SFC dataset show that, unlike TFL, LAST_HOUR is the most revealing among the assignment priors, with a mean error of 0.53 (vs. 0.63 for LAST_DAY and 0.67 for LAST_WEEK). Interestingly, Adv profiles cabs more efficiently knowing their last hour's ROIs than with probabilistic priors, e.g., their most frequently visited ROIs (FREQ_ROI) or their most frequent ROIs for the time slots of a week (ROI_DAY_WEEK). That is, cabs in San Francisco are more likely to appear in those ROIs they visited during the last hour, while their daily/weekly patterns are less regular. In all assignment prior cases, BAYES and MAX_USER reduce Adv's error by little, while MAX_ROI increases it, thus, the privacy loss from the aggregates is again quite low.



(a) TFL

(b) SFC

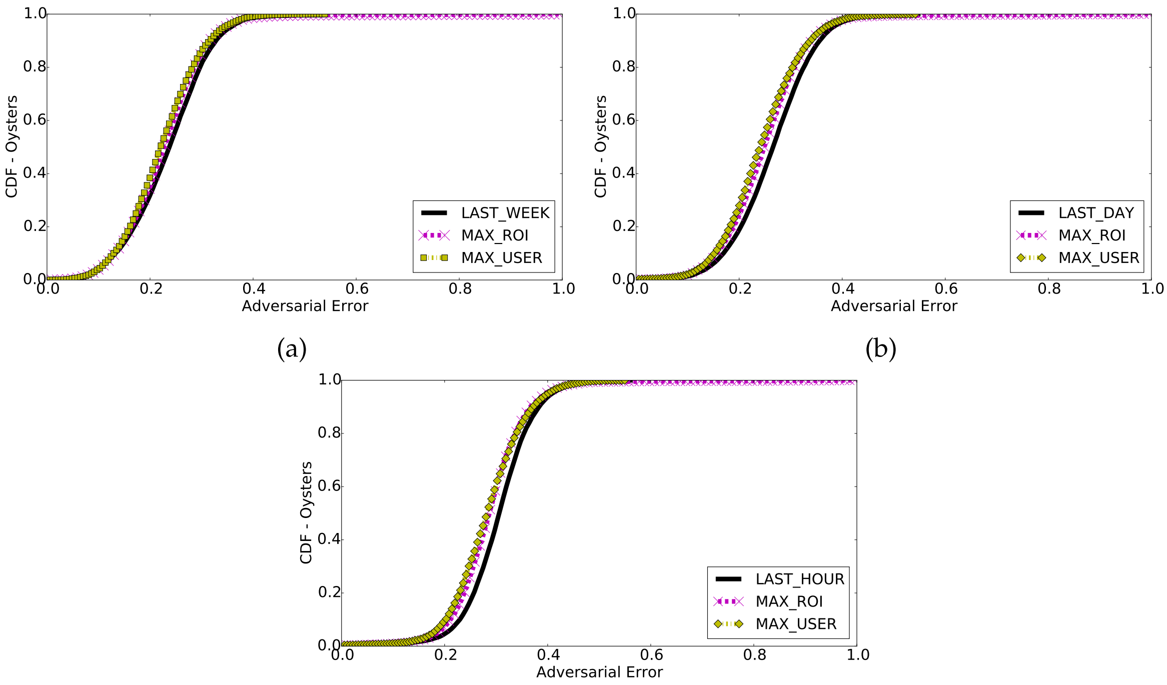
FIGURE B.5: TIME_DAY_WEEK prior – Adv's localization error.

B.2 User Localization

Probabilistic Priors

TIME_DAY_WEEK. Figure B.5 displays Adv's error when localizing users with the TIME_DAY_WEEK prior. For TFL (Fig. B.5a), we observe that ALL results in a very large error (0.99 on average). This is not surprising since TIME_DAY_WEEK is a uniform distribution over ROIs, for the time slots that users are likely to be in the transportation system. ALL after BAYES achieves negligible privacy loss (0.03), while we observe no adversarial advantage between POP and POP after BAYES due to the very small prior probabilities. Furthermore, both MAX_USER and MAX_ROI improve remarkably Adv's performance compared to ALL and they yield 0.79 and 0.77 average privacy loss, respectively. We note that MAX_ROI achieves error larger than 0.25 for 20% of the users, while MAX_USER yields error larger than 0.25 for only 5% of the users, i.e., those users that report the most locations and always get assigned to locations to consume the aggregates.

For the SFC data (Fig. B.5b), we observe that ALL yields 0.84 average error, while ALL after BAYES results in small privacy loss (0.04). Once again, POP is the worst inference strategy as the small probabilities of the prior do not exceed the threshold λ' and cabs are predicted to be outside the network. Moreover, unlike the case of ROI_DAY_WEEK, MAX_ROI now improves Adv's performance towards localizing the cabs, yielding 0.1 privacy loss. MAX_USER achieves a similar mean loss in privacy, however, Adv's knowledge is only improved for 60% of the cabs compared to the baseline ALL. Once again, we remark how localization strategies result to different amount of privacy leakage on sparse (TFL) and dense (SFC) datasets.



(a)

(b)

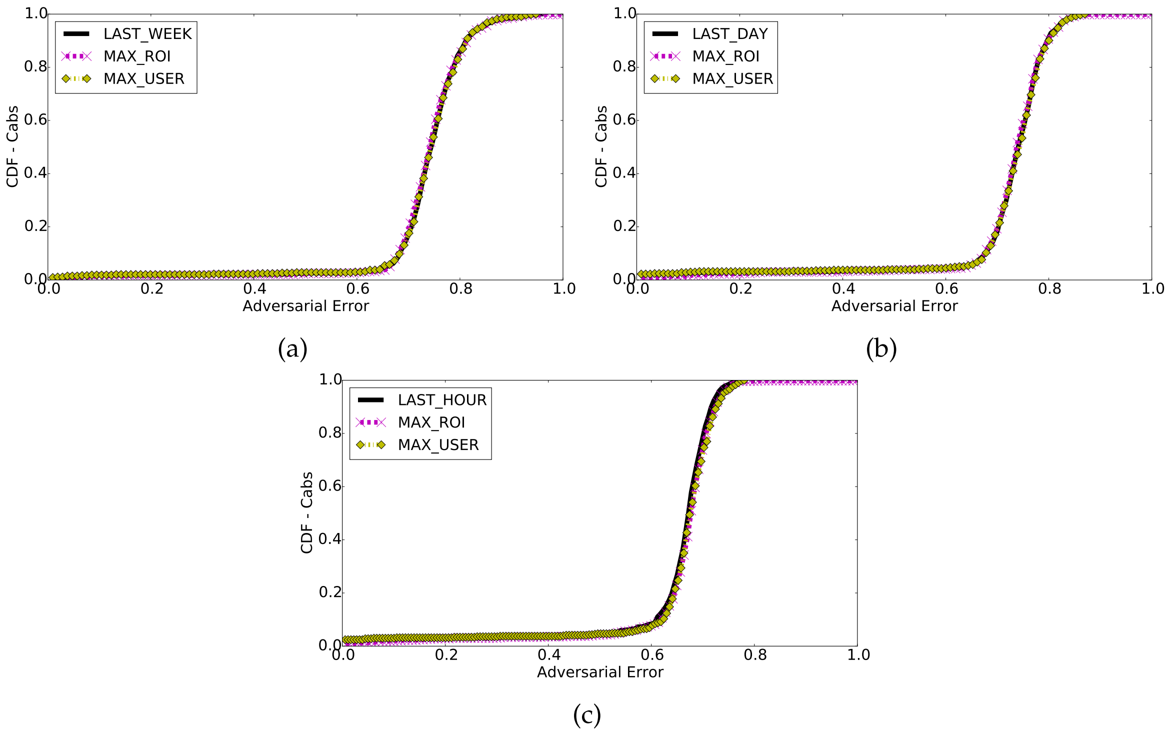
(c)

FIGURE B.6: LAST_WEEK (a), LAST_DAY (b), and LAST_HOUR (c), priors – Adv's localization error on the TFL dataset.

Assignment Priors

Figures B.6 and B.7 plot the results discussed in Section 6.2.3. In particular, we evaluate Adv's performance against user localization, i.e., predicting users' future locations with a seasonal part of their location matrix as prior knowledge. We experiment with LAST_WEEK, LAST_DAY, and LAST_HOUR, and focus on the MAX_ROI and MAX_USER inference strategies. Adv's baseline prediction is to replicate the prior, as described in Section 6.1.2.

LAST_WEEK. Adv's average error localizing tube passengers with LAST_WEEK is 0.24 (see Fig. B.6a). Both MAX_ROI and MAX_USER inference strategies vaguely improve her performance and yield small privacy loss (0.02). This indicates that users reporting lots of ROIs and ROIs themselves show regularity within weeks. Furthermore, MAX_USER is more consistent in improving Adv's localization success than MAX_ROI, which increases Adv's error (compared to the prior) for 5% of the users. For SFC cabs (Fig. B.7a), we observe that Adv's average localization error is 0.73, while both MAX_ROI and MAX_USER do not reduce it further. Unlike TFL, we observe that the aggregates do not give any advantage to Adv in localizing taxis and there is no privacy loss.



(a)

(b)

(c)

FIGURE B.7: LAST_WEEK (a), LAST_DAY (b), and LAST_HOUR (c), priors – Adv's localization error on the SFC dataset.

LAST_DAY. With LAST_DAY, Adv's mean error in predicting TFL passengers' locations during the inference week is 0.27 (see Fig. B.6b), thus, this prior is less revealing than LAST_WEEK (0.24, cf. B.6a). This indicates that commuters show stronger weekly seasonality in their journeys. MAX_ROI and MAX_USER achieve very small privacy loss (0.01 and 0.02, resp.), thus, aggregate time-series enhance insignificantly Adv's inference goal. MAX_USER constantly reduces Adv's error over the prior, while MAX_ROI increases it for a small percentage of users (5%). For SFC (Fig. B.7b), Adv's localization error with LAST_DAY is 0.71 indicating that cabs are a bit more likely to appear in the ROIs of last day rather than those of last week (0.73 error). Once again, the greedy inference strategies do not help Adv improve her predictions and there is negligible privacy loss.

LAST_HOUR. Finally, we plot Adv's error while localizing users with the LAST_HOUR prior. For TFL (Fig. B.6c), we observe that her error is now larger (0.31) compared to the two previous cases (LAST_WEEK, LAST_DAY) indicating that, in general, commuters do not show up in the ROIs of their last hour. The knowledge of the aggregate time-series enables Adv to improve her localization performance insignificantly and the greedy strategies MAX_ROI and MAX_USER yield negligible amount of privacy loss (0.01 and 0.02, resp.). When localizing SFC cabs with LAST_HOUR, Adv's mean error is 0.64 (see Fig. B.7c). Thus, as this assignment prior helps Adv localize cabs better than LAST_DAY (0.71) or LAST_WEEK (0.73) and unlike tube commuters, taxis are more likely to appear in the locations they have recently reported. Both inference strategies lead to very small privacy loss.