

Scuola Normale Superiore  
Scuola Superiore S.Anna  
Università di Pisa  
Consiglio Nazionale della Ricerca (CNR)  
Scuola IMT Alti Studi Lucca

*Individual Human Mobility Models  
for Sustainable Cities Applications*

Ph.D. Thesis

Supervisors:

**Prof. Mirco Nanni**

**Prof. Giovanni Comandè**

**Prof. Gennady Andrienko**

PhD Candidate:

**Agnese Bonavita**





## Abstract

Humans are inherently mobile creatures. The way we move around our environment has consequences for a wide range of problems, including the design of efficient transportation systems and the planning of urban areas. Having good prediction models able to abstract and infer human mobility behaviours within a city is of extreme importance to improve the urban life.

This thesis proposes to study human behavior and dynamics through a combination of techniques from network science and data mining. In the context of human mobility, we use GPS data from vehicles to define trajectories in order to understand the mobility patterns. We based our mobility models on the Individual Mobility Networks, a graph representation of users trips that will be presented and used in this thesis. Our work also aims to represent a step towards a reliable Mobility Analysis framework, capable to exploit the richness of the spatio-temporal data nowadays available. The work done is an exploration of meaningful open challenges, from an efficient Trajectory Segmentation of low sampling GPS data to the definition of a stable car crash prediction model. From simulation of electric vehicles to the ethics aspects of mobility data usage we have today many ways to make our cities more sustainable and smart. Another promising perspective is the use of such extracted knowledge to investigate more extensive topics such as Geographical Transfer Learning and Explainability.

Further experimentation has been performed in order to improve the characterization of the individual human movements having a more complete and richer picture of that.



## Acknowledgement

Rather than a job, or a course of study, a Ph.D. is a three year long, complex and not linear journey you make with your friends, colleagues, random encounters, loves, failures and successes. Only a Ph.D. can really understand what a Ph.D is.

First of all, I want to thank my supervisors Mirco Nanni and Giovanni Comandè that believed in me and supported me in these years. They guided me and shared with me exciting moments of research.

A particular "*Grazie*" to Mirco who sometimes was more than a advisor, putting up with my mood swings, my anxieties and my worries.

Many thanks also to the senior researchers Riccardo, Paolo and Omid that helped me to better understand my results and findings.

I want to thank also all my Ph.D. colleagues of the XXXIV cycle whom I shared the doctoral years with, between doubts about the future and amazing summer schools.

Many thanks to my friends at Scuola Normale Superiore and in Pisa that made these 3 years enjoyable and full of unforgettable moments. Thank you to all those who have been close to me during these years.

A special thanks to my friend Leonardo: if I see the end of this PhD it is also thanks to your help. You have always been there for all my problems, even if on the other side of the world.

Thanks to my best friends: Lucrezia, Matilde, Alessia, Beatrice, Marco and Pietro that supported me during all the steps of my life, including this hard period. You are my crutches for every time I can't stand up on my own.

The final, enormous "thank you" goes to all my family.

My mum and my dad are the ones I want to dedicate this thesis: they constantly make me a better person giving me the example with their lives. Even if I'm not good at showing you my gratitude day by day, I'll always consider me as the luckiest in the world for having so special parents. As I said once, (but it is always better to repeat it), you are the coaches of my life. I will never be completely ready without you.



# Contents

<b>Introduction</b>	<b>xv</b>
<b>1 Summary of Contributions</b>	<b>1</b>
1.1 Building and Representing Individual Mobility Models . . . . .	2
1.1.1 Graph Embedding . . . . .	3
1.2 Applications of IMNs . . . . .	4
1.2.1 Electric Mobility . . . . .	4
1.2.2 Crash event risk prediction . . . . .	5
1.3 Explore Open Questions . . . . .	6
1.3.1 Transfer Learning . . . . .	6
1.3.2 Explainability . . . . .	8
<b>2 Setting the Stage</b>	<b>9</b>
2.1 Data Analytics Landscape . . . . .	9
2.2 Background on Mobility Data Analysis . . . . .	10
2.2.1 Mobility Data Science . . . . .	10
2.2.2 Data Mining and User Profiling . . . . .	11
2.2.3 Human Mobility Data . . . . .	13
2.2.4 Mobility Metrics . . . . .	17
2.2.5 Individuality vs Collectivity . . . . .	19
2.2.6 Human Mobility Patterns . . . . .	22
2.2.7 Personal location detection . . . . .	23
2.3 Transfer Learning . . . . .	26
2.3.1 Basic Definitions . . . . .	27
2.3.2 Transfer Learning Scenarios . . . . .	30
2.3.3 Avoiding Negative Transfer . . . . .	36
2.3.4 Applications . . . . .	37
2.4 Ethical Aspects of Data Science . . . . .	39
2.4.1 General Data Protection Regulation (GDPR) . . . . .	40
2.4.2 Privacy Aspects of Human Mobility Analysis . . . . .	43
2.5 Explainability . . . . .	44
2.5.1 Lack of Transparency . . . . .	44

2.5.2	Need of interpretability . . . . .	46
2.5.3	Interpretable Models . . . . .	47
2.5.4	Interpretability techniques . . . . .	49
<b>3</b>	<b>Act I: Human Mobility as a Complex Network</b>	<b>51</b>
3.1	What is a Complex Network? . . . . .	51
3.1.1	Graph Representations . . . . .	53
3.2	Individual Mobility Networks . . . . .	56
3.2.1	Problem Formulation . . . . .	57
3.3	Segmentation . . . . .	61
3.3.1	Trajectory Reconstruction . . . . .	63
3.3.2	Self-Adaptive Trajectory Segmentation . . . . .	66
3.3.3	Individual and Collective Adaptive Trajectory Segmentation . . . . .	68
3.3.4	Experiments . . . . .	71
3.3.5	Conclusions . . . . .	84
3.4	On the Pursuit of Graph Embedding Strategies for Individual Mobility Networks . . . . .	86
3.4.1	Graph Embedding: State of Art . . . . .	88
3.4.2	Comparative study of IMN properties . . . . .	91
3.4.3	Features, algorithms and validation task . . . . .	93
3.4.4	Empirical evaluation . . . . .	96
3.4.5	Summary and Conclusions . . . . .	101
<b>4</b>	<b>Act II: Individual Mobility Models at Work</b>	<b>105</b>
4.1	Electrical Vehicles . . . . .	105
4.1.1	Related Works . . . . .	107
4.1.2	Problem definition . . . . .	111
4.1.3	Simulation framework . . . . .	114
4.1.4	Experiments . . . . .	122
4.1.5	Notes on privacy and ethical issues . . . . .	131
4.1.6	Conclusions . . . . .	132
4.2	Comparative cities studies through City Indicators . . . . .	135
4.2.1	Local City Indicators . . . . .	135
4.2.2	Flows in a Grid Network . . . . .	137
4.2.3	Individual Mobility . . . . .	139
4.2.4	Roads and Traffic . . . . .	139
4.2.5	Global City Indicators . . . . .	140
4.2.6	Complete Network of Cities . . . . .	140
4.2.7	Ego-Networks . . . . .	142
4.2.8	City Clustering . . . . .	143
4.2.9	Testing Model Transferability . . . . .	148
4.3	Car Crash Prediction . . . . .	151
4.3.1	IMN-based Crash Risk Prediction . . . . .	156
4.3.2	Experiments . . . . .	161
4.3.3	Geographically Transferred Crash Prediction Evaluation . . . . .	170

4.3.4	Notes on privacy and ethical issues . . . . .	175
4.3.5	Conclusion and Future Work . . . . .	177
4.4	Assessing privacy risks in human mobility . . . . .	178
4.4.1	Personal Data vs Sensitive Data . . . . .	178
4.4.2	What can I infer with mobility data? . . . . .	181
4.4.3	Hidden Risk behind Mobility Data . . . . .	185
4.4.4	Conclusions . . . . .	186
<b>5</b>	<b>Epilogue</b>	<b>189</b>
5.1	Conclusions . . . . .	189
5.2	Future Works . . . . .	191



# List of Figures

1.0.1 <i>My PhD research project in a nutshell</i> . . . . .	2
2.2.1 <i>The process of Knowledge Discovery in Databases.</i> . . . . .	13
2.2.2 <i>Some of the most used mobility data in data mining.</i> . . . . .	14
2.2.3 <i>Brownian motion (left) is described as a random walk in which all the steps give the same contribution. Levy-flight (right) occurs when the trip is dominated by a few very large steps</i> . . . . .	22
2.2.4 <i>Visualization of the three application classes: (a) user modeling where the object of analysis is a single user (represented by circles of the same color), (b) place modeling where the object is a geographic area (visited by different users; each is represented by a circle of its own color), (c) trajectory modeling where the object is a set of spatial-temporal points by the same user (represented by a yellow arrow). The map was generated using Google Maps</i> . . . . .	24
2.2.5 <i>Orange points represent the stop observations input of TOSCA. Blue dotted circles correspond to the X-Means clusters and the blue points to their medoids, which are then processed by Single Linkage. On the resulting dendrogram we highlight the differences among the element distances. The red interval represents a possible cut. The image comes from [128].</i> . . . . .	26
2.2.6 <i>Personal locations detected with X -means (left) and TOSCA (right). Different colors denote the different clusters (personal locations) to which each point is assigned [16].</i> . . . . .	26
2.3.1 <i>Traditional Learning vs Transfer Learning</i> . . . . .	29
2.3.2 <i>Some intuitive examples of transfer learning</i> . . . . .	29
2.3.3 <i>A schema of the four typical scenarios [96].</i> . . . . .	31
2.3.4 <i>Inductive learning can be viewed as a directed search through a specified hypothesis space [207]. Inductive transfer uses source-task knowledge to adjust the inductive bias, which could involve changing the hypothesis space or the search steps [304].</i> . . . . .	33

2.3.5	<i>Bayesian learning uses a prior distribution to smooth the estimates from training data. Bayesian transfer may provide a more informative prior from source-task knowledge. [304]. . . . .</i>	33
2.3.6	<i>An example of a concept hierarchy that could be used for hierarchical transfer, in which solutions from simple tasks are used to help learn a solution to a more complex task. Here the simple tasks involve recognizing lines and curves in images, and the more complex tasks involve recognizing surfaces, circles, and finally pipe shapes. [304]. . . . .</i>	34
2.3.7	<i>(a) One way to avoid negative transfer is to choose a good source task from which to transfer. In this example, Task 2 is selected as being the most related. (b) Another way to avoid negative transfer is to model the way source tasks are related to the target task and combine knowledge from them with those relationships in mind [304]. . . . .</i>	37
3.1.1	<i>The “Seven Bridges of Königsberg” problem representation . . . . .</i>	51
3.1.2	<i>Different ways to represent a graph. . . . .</i>	55
3.2.1	<i>The IMN extracted from the mobility of an individual. Edges represent the existence of a route between locations. The function <math>\omega(e)</math> indicates the number of trips performed on the edge <math>e</math>, while <math>\delta(x)</math> the total time spent in a location <math>x</math> . . . . .</i>	59
3.3.1	<i>Frequency distribution of pseudo-stop durations for a user trajectory (left), and the durations of the segments obtained using a specific threshold to cut the trajectory (right). The threshold used corresponds to the vertical line on the left image. . . . .</i>	67
3.3.2	<i>Time threshold distributions for trajectories obtained with ATS in Rome and London. The peaks show the ideal thresholds to be set to build the trajectories. . . . .</i>	72
3.3.3	<i>Boxplots for the <math>MF_{.25}</math> results. On the Rome data ATS yields better results than the FTS solutions, while in London all three produce almost the same results. The variability of ATS results is consistently smaller than the other methods, which is a sign of robustness. . . . .</i>	74
3.3.4	<i>Distributions of average number of points per segment obtained by ATS. . . . .</i>	75
3.3.5	<i>Distribution of the number of trajectory segments over Rome (top) and London (bottom) with each segmentation method (on the columns, grouped by family). . . . .</i>	76
3.3.6	<i>Distributions of the average length (top) and duration (bottom) for the trajectory segments returned by ATS (left) and FTS (right) for the area of Rome. . . . .</i>	76
3.3.7	<i>Trajectory segmentation returned by <math>FTS_{1200}</math> (left) and ATS (right). The user is traveling from South to North. Top: spatial representation showing the trajectory segments. Center: temporal segmentation showing the inter-leaving time between GPS points. Bottom: zoom on the service area highlighted in the top maps where the user probably stops for <math>\sim 15</math> minutes. Best viewed in color. . . . .</i>	77



3.3.8	<i>Time threshold distributions for the users obtained by ACTS<sub>LOC</sub> in Rome and London. Compared to ATS, the distributions are more concentrated on the two peaks.</i>	79
3.3.9	<i>Time threshold distributions for the users obtained by ACTS<sub>WOTC</sub> in Rome and London. The overall distributions are very similar to ACTS<sub>LOC</sub>.</i>	79
3.3.10	<i>Points distribution in Rome and London datasets over the geohash grid (<math>h = 6</math>).</i>	80
3.3.11	<i>Comparison of ACTS<sub>WOTC</sub> vs. ATS thresholds for all user-cell pairs, on Rome (left) and London (right). In both cases the difference appear significant and overall symmetric.</i>	80
3.3.12	<i>Comparison of ACTS<sub>WOTC</sub> vs. ATS thresholds for all user-cell pairs, lowering the precision value (<math>h = 5</math>), on Rome (left) and London (right). In both cases the difference appear significant and overall symmetric.</i>	81
3.3.13	<i>Comparison of ACTS<sub>WOTC</sub> vs. ATS thresholds for all user-cell pairs, increasing the precision value (<math>h = 7</math>), on Rome (left) and London (right). In both cases the difference appear significant and overall symmetric.</i>	82
3.3.14	<i>Boxplots for MF<sub>.25</sub>. On the Rome data ACTS<sub>WOTC</sub> yields results similar to ATS, while ACTS<sub>LOC</sub> significantly improves them. On London the differences are less pronounced.</i>	83
3.3.15	<i>Boxplots for MF<sub>.25</sub>. In this case it is possible to see the comparison in terms of performance between our approaches and the DBSCAN and OPTICS cluster methods. In both cases the performance of the cluster methods are visibly worse than those achievable with ATS and ACTS.</i>	83
3.3.16	<i>Distribution of the number of segments, points and trajectories (from left to right) over Rome (top) and London (bottom).</i>	84
3.3.17	<i>Run times experiments in function of the number of users and the data collection period. In both cases the time trend grows quite linearly.</i>	85
3.4.1	<i>Distribution of nodes properties in various graph datasets.</i>	92
3.4.2	<i>Sample IMNs.</i>	93
3.4.3	<i>Impact of features: network measures (NET), mobility (TRJ), location usage (USE). Top plot: without geospatial features; bottom plot: with geospatial features.</i>	99
3.4.4	<i>Query test 1: Top 10 most similar IMNs to a sample user, using Feather-N (left) and GCN-RoG (right) embeddings.</i>	102
3.4.5	<i>Query test 2: Top 10 most similar IMNs to a sample user, using Feather-N (left) and GCN-RoG (right) embeddings.</i>	102
4.1.1	<i>Maximum speed in Tuscany (left) using the attribute MaxSpeed and (right) inferring it from the Highway attribute</i>	116
4.1.2	<i>Nodes altitudes on a portion of Tuscany (blue=low, yellow=high)</i>	116
4.1.3	<i>Slopes of roads on a portion of Tuscany (blue=flat, red=steep)</i>	117

4.1.4	<i>Distribution of the number of trips involved in the experiments: (left) trips per user; (right) trips per day of each user.</i>	123
4.1.5	<i>Distribution of trip lengths (left) and duration (right).</i>	124
4.1.6	<i>(left) Number of stations by maximum recharge power provided (in KW); (right) Geographical distribution of stations (red=slow, orange=fast, green=rapid, blue=ultra-rapid).</i>	124
4.1.7	<i>Simulation runtime for different input sizes (seconds vs. n. of trips).</i>	125
4.1.8	<i>Average runtime for trips, divided by trip length, for the Home and Work scenario (top) and the Public-only one (bottom). The black line represents the number of trips in each group (log-scale).</i>	126
4.1.9	<i>Distribution of trip lengths and duration aggregated by user for the 4 scenarios.</i>	128
4.1.10	<i>Usage frequency of stations by power by increasing values of <math>k_{charge}</math> (weight of recharge time in the path selection algorithm).</i>	130
4.1.11	<b>Use case A:</b> <i>IMNs (top) and temporal graph of charge (bottom). Left: Home + Work scenario; right: Public-only. Size and width in IMNs represent frequency of stop/trip, darkness of red represents frequency of recharge. In the temporal graph, passive charges are green, those at stations are red.</i>	132
4.1.12	<b>Use case B:</b> <i>same layout as Figure 4.1.11.</i>	133
4.2.1	<i>The areas of study: 10×10km squares centered on each municipality in Tuscany.</i>	135
4.2.2	<i>Disconnected nodes vs. flow threshold.</i>	141
4.2.3	<i>Network of correlations for the first set of attributes (total graph).</i>	143
4.2.4	<i>Dendrogram and selected clusters.</i>	144
4.2.5	<i>Map of clustered municipalities</i>	145
4.2.6	<i>Selected cells for some municipalities.</i>	146
4.2.7	<i>XGBoost traffic forecasting on Florence (green) against real values (blue).</i>	147
4.2.8	<i>Transfer scores matrix with cluster separation.</i>	149
4.2.9	<i>NMRSE mean values for all train-test pairs.</i>	150
4.3.1	<i>Schema of the three geographical transfer learning approaches explored. The input city data is used either to extract individual city models (downward) or create a resampled dataset (upward). In the first case, Approach 1 selects the best model, while Approach 2 creates an ensemble. In the second case, a new model is built on the resampled data.</i>	159
4.3.2	<i>Geographical areas of experiments. Dataset 1 includes London in UK (left), Tuscany and Rome in Italy (center). Dataset 2 is a zoom on the Tuscany area (highlighted in the center) by also considering its 10 provinces, shown on the right.</i>	162
4.3.3	<i>ROC curve for different areas for D1.2 and D1.3.</i>	168
4.3.4	<i>F1 score and auc for the RFI and RFP approaches on the Tuscany dataset by varying the prediction span from 1 month to 4 months.</i>	169

4.3.5 Receiver Operating Characteristic (ROC) curve for geographically transferred crash prediction with target areas Pisa and Florence for D2. . .	173
4.3.6 Aggregated SHAP explanation of the five most important features for geographically transferred crash prediction with target areas Pisa and Florence for D2 using A1. . . . .	174
4.3.7 Aggregated SHAP explanation of the five most important features for geographically transferred crash prediction with target areas Pisa and Florence for D2 using A3. . . . .	175



# List of Tables

3.3.1	<i>Evaluation on Rome data. The first three columns show the measures illustrated in Section 10. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of segments.</i>	73
3.3.2	<i>Evaluation on London data. The first three columns show the measures illustrated in Section 10. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of trajectories.</i>	73
3.3.3	<i>Evaluation on Rome data. The first three columns show the measures illustrated in Section 10. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of segments.</i>	81
3.3.4	<i>Evaluation on London data. The first three columns show the measures illustrated in Section 10. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of segments.</i>	82
3.4.1	<i>Comparison of statistics of various graph datasets.</i>	92
3.4.2	<i>AUC performances of embedding methods on the province classification problem, based on a logistic classifier.</i>	97
3.4.3	<i>AUC of Fully Connected architecture</i>	100
4.1.1	<i>General Simulation Parameters</i>	120
4.1.2	<i>Overall impact of EVs on trips. We focus on the comparison between the average length and the average duration of real and simulated trips.</i>	127
4.1.3	<i>Temporal variations of EVs impact for the four scenarios. The two months period is split in four shorter periods of two weeks each (<math>t_1</math>, <math>t_2</math>, <math>t_3</math> and <math>t_4</math>) in order to see how the percentages change inside the selected time.</i>	128
4.1.4	<i>Geographical variations of EVs impact. Each province is associated to the trips that start from it.</i>	129
4.1.5	<i>Effects of varying <math>k_1</math> and <math>k_2</math> on EVs impact. <math>k_{charge}</math> is fixed to 0.2.</i>	130

4.2.1	<i>Cluster Population</i>	144
4.3.1	<i>Datasets summary as average values of some features.</i>	164
4.3.2	<i>Performance for the experimental setting D1. For D1.1 the metrics are aggregated in terms of means and standard deviation over different periods. The best performance results are those underlined.</i>	167
4.3.3	<i>Crash prediction performance for the various geographical units inside Tuscany in D2. Each model is trained and tested in the same area similarly to D1.2. The last line report the performance of a model trained and tested on the whole dataset similarly to D1.3.</i>	170
4.3.4	<i>Geographically transferred crash prediction auc for NN and RF. The best transfer are underlined, the transfer suggested by Approach 1 – Best City Transfer w.r.t the similarity of city indicators are in bold.</i>	171
4.3.5	<i>Geographically transferred crash prediction auc for NN and RF for the various approaches. Best results for each target area are highlighted in bold.</i>	171

# Introduction

One of the most fascinating challenges of our time is to understand the complexity of the global interconnected society and possibly to predict human behavior. A great part of human behavior is observable through individual movements, registered in many different layers: mobile phone network, GPS devices, social media applications, road sensors, credit card transactions. Movement is the “hardware” of our daily life. We move to perform any activity: we have to move to bring children at school, to buy a new electronic device, to meet with colleagues at work and so on [113]. If we understand the patterns of human movement, we can also comprehend how to improve the metropolis conditions, reduce urban traffic and create smarter cities. Modeling human mobility is a challenging task due to many reasons: first of all human trajectories are extremely high dimensional and second the trajectory of each individual is very unique in the geographical space [208]. Nevertheless human movements are similar in an underlying semantic space, which gives meaning to a trajectory [42]. For example, majority of people leave their respective home in the morning and go to their respective work places, spend time for lunch at their favorite cafes and visit their points of interest. This is not substantially different among individuals since the semantics and the geographic features of locations in a trajectory are all correlated with each other. There is also a correlation between the mobility features of different individuals in an area. Thus, it is extremely hard to characterize (geographically and semantically) meaningful location trajectories using sequential models, where a location is developed given only a short list of its preceding visited locations. The whole trajectory, instead, needs to be built in a consistent manner, using a model that captures all its features. Several questions are still open for investigation, from general ones such as: *How to capture the multi-dimensional and multi-scale of individual mobility in a single model?*, to specific ones as: *What are the frequent patterns of people’s travels? How do big data attractors and extraordinary events influence mobility? How to predict mobility related events such as traffic jams or car accidents in the near future?*

In the world of Big Data, mobility is one of the most interesting phenomena to study, both for technological and philosophical reasons. The ensemble of all the possible kind of data produced by billions of everywhere connected users represents the big mosaic of mobility data. Every single piece of such mosaic encloses a dimension of human mobility, every right combination of two or more pieces is a step towards the understanding of big picture. So far, main results in Mobility Data Analysis have been the comprehension of (almost) all the single dowels; the next challenge is

to compose the whole mosaic.

Starting with the Global Positioning System (GPS) tracks only, the goal of my research work is to find new answers to those questions focusing the attention on the study of the individual mobility. The objective is to model the mobility of the single individual as a whole, creating a unique, complete picture of it adding semantics to the raw data.

The main idea is to exploit innovative data mining algorithms trying to build a model able to give us an accurate prediction of individual movements at the city level. We want also to pay attention to the privacy aspect and understand how privacy limits interfere with the quality of result. Moreover, the human mobility context considered here presents two very important and interesting open challenges that we would like to take into consideration: the transfer learning method for mobility data models and the explainability aspects behind risk event recognition. Both were born and evolved in contexts far from the one developed in this thesis but we think that it will be challenging to try to use them in the mobility context. The first one aims to find a way to exploit a pre-trained model of trajectory prediction in another geographical setting. Building a universal algorithm able to predict the future tracks of users would be very useful in those context where data availability is low. The second aims to understand how the Black Box works: usually machine learning algorithms map user features into classes or scores without explaining why and how, because the decision model is either not comprehensible to stakeholders, or secret. This is worrying not only for the lack of transparency, but also due to the possible hidden biases.

In the field of mobility and risk prediction there is still lot of work to do and a wide range of open scenarios to explore. One of the most interesting challenges is trying to transform predictions into "prescriptive rules" to prevent risk phenomena (for example car accidents in the insurance industry or heart attacks in the medical field). Based on these observation, in this thesis we want to face some challenges related to the mentioned topics: from trajectory segmentation to electric vehicles simulations, from mobility graph embedding to geographical transfer learning.

## Motivation

Mobility is definitely a critical phenomenon in urban environments.

In fact, we can think about it as one of the most important mechanisms underlying the structure and dynamics of contemporary cities. Indeed, cities are places where intensive buying, selling or exchanging goods is taking place, where individuals commute to work or meet with other individuals. An obvious means to achieve all this is transportation. Here is where technology comes into play through the speed of the different transportation modes. This velocity has increased considerably as technology evolved and modified the spatial organization of cities. For example the possibility to reach a place for an individual depends on the transportation mode. For a pedestrian, the reachability horizon is typically isotropic and small, whereas the car



permits a wider yet anisotropic exploration of space due to the existing infrastructures. The described correlation between the spatial organisation of a city and the available technology at the time has been demonstrated by [13] for American cities. The authors of the study show how many big cities, such as Denver, grew around rail stations which unleashed the development of central business districts. Later automobile-era cities such as Dallas, on the other hand, display a spatial structure primarily conditioned by the highway system. In terms of mobility, the traditional city center can be considered as the location that minimises the average distance to all other locations in the city. As a natural consequence, it has thus historically attracted businesses and residences, leading to competition for the limited space among individuals or firms, which gave rise to the real-estate market.

In this wide and complex scenario we want to focus on how cities can be theoretically seen as a big complex network and how this allows to investigate them in some fascinating ways.

But why complex network?

The science of networks has been witnessing a rapid development in recent years: the metaphor of the network, with all the power of its mathematical devices, has been applied to complex, self-organized systems as diverse as social, biological, technological and economic, leading to the achievement of several unexpected results in the works of Barabási, Strogatz, and others [5] [285]. The understanding of spatial networks detectable in biological, technological and infrastructural systems has seen an unprecedented progress in the recent years. However, despite a significant amount of research on these kinds of networks, in disciplines covering among others mathematics, physics and geography, their structural and dynamical properties are not yet completely explored in the mobility context. Our goal is to add a small missing piece to the huge puzzle in order to show a few new applications within the set of possible solutions for which complex networks can be used.

## Contribution and Organization of the thesis

I decided to imagine my thesis work as a "theatre play". The thesis is organized in four parts. The first part "*Summary of Contributions*" has been thought of as a sort of short presentation of the work that will be presented in the thesis. Having dealt with very different topics which require detailed references to theoretical parts, we consider that a summary of the most significant aspects is necessary and helpful to the comprehension. Basically, we prepare ground for the subsequent body of work by motivating why studying human mobility is important, by introducing core concepts we are going to work with, and by providing an extensive review of literature in urban mobility, and baseline techniques and models we will build upon and develop. The second part, *Setting the Stage*, is devoted to the background needed for the introduction of the novelty parts and to the issue related to the understanding of human behavior through the analysis of Big Data. In this part there are no original contributions or personal interventions.

The next two parts, called "*Acts*" (to respect the idea of a play structure) are about my personal contribution and the challenges we tackle.

In *Act I* we introduce the core topic of my work, namely complex networks, and we describe in detail the methodology and process of building the urban mobility network. Section 3.1 is related to an introduction about complex system and individual mobility networks (Sec.3.2) . We present also (in Section 3.3) the first original work of the thesis: a new trajectory segmentation method originated from a multi-granularity perspective that aims to look for a better understanding of the problem and for an user-adaptive solution. In conclusion in Section3.4 we investigate the graph embedding challenge in the mobility framework. We discuss the existing approaches to graph embedding and the specificities of IMNs, trying to find the best matching solutions. Our goal is to exploring Embedding Strategies for Individual Mobility Networks in order to compare two of them even if very different a priori.

*Act II* is dedicated to three interesting problems in real life context. Section 4.4 it is related to the privacy aspect of mobility data and the risks that lie behind the improper use of such data. Section 4.1 is about the electrical vehicles impact in citizens life and in our cities organizations. Section 4.2 introduces the city indicators in details while Section 4.3 tackles the car crash prediction problem and the geographical transfer learning challenge behind that.

In conclusion, like at the end of a show, the *Epilogue* chapter concludes the thesis by presenting possible future research directions in the study of human mobility.

## Publications

This section lists all the publications that constitute the innovative contributions of this thesis. The single papers are inserted into the chapters of this thesis in order to make the context and overall objectives of the entire work uniform and easily understandable.

- A.Bonavita, R.Guidotti, M.Nanni, "*Self-Adapting Trajectory Segmentation*" 3rd International Workshop on Big Mobility Data Analytics (BMDA), see Sec. 3.3;
- A.Bonavita, R.Guidotti, M.Nanni, "*Individual and Collective Adaptive Trajectory Segmentation*" *GeoInformatica: International Journal on Advances of Computer Science for Geographic Information Systems*, see Sec. 3.3;
- A.Bonavita, R.Guidotti, M.Nanni, "*City Indicators for Mobility Data Mining*" 3rd International Workshop on Big Mobility Data Analytics (BMDA), see Sec. 4.2;
- A.Bonavita, G.Comandè, "*Mobility Data (knowledge discovery from)*" in G. Comandè (editor) *Encycopaedia of Law and Data Science*, Edward Elgar, see Sec. 4.4;

- A.Bonavita, R.Guidotti, M.Nanni, "*City Indicators for Geographical Transfer Learning: An Application to Crash Prediction*" Transactions on Intelligent Transportation Systems, see Sec. 4.3;
- O.I.Alamdari, A.Bonavita, P.Cintia, M.Nanni, "*From fossil fuel to electricity: studying the impact of EVs on the daily mobility life of users*" Transactions on Intelligent Transportation Systems, see Sec. 4.1;
- O.I.Alamdari, A.Bonavita, P.Cintia, M.Nanni, "*From fossil fuel to electricity: studying the impact of EVs on the daily mobility life of users*" Transactions on Intelligent Transportation Systems, see Sec. 3.4;



# Summary of Contributions

In this chapter the main contributions of this thesis are presented.

The common thread of this PhD research is the study of mobility at an individual level. So, all the works presented and published start from the study of the behavior of the individual users.

Figure 1.0.1 shows the starting idea of the research work: in the picture there are three main objectives (represented by the three colored circles) linked together that will be presented more in detail in this chapter.

The challenges that a mobility data scientist has to face are multilateral: they could be related to algorithm building and implementation or to the application of existing methods to very specific tasks. Theory and application are always connected and they are mutually dependent on each other. So the idea behind this research project is to bring some contributions to every level, from implementation of general methods to application for very specific tasks.

In addition a particular attention has been given on two data mining open question: transfer learning and explainability. Both of them have been studied in the mobility context inside a very specific problem.

## **Privacy Aspects of Mobility Data**

Since mobility data can convey personal and sensible information, our work paid a particular attention to the privacy and ethic aspects related to mobility data. Indeed it becomes fundamental to take into consideration the legal and ethical aspects of processing of personal data, especially given the entry into force of the General Data Protection Regulation (GDPR) in May 2018. This is why it aims at devising methodologies and tools that enable us to arrive at data science solutions that are demonstrably in accordance with shared societal and moral values. For this reason, it is important that legal requirements and constraints are complemented by a solid understanding of ethical and legal views and values such as privacy and data protection.

As a larger part of modern life is digitized, individuals generate an increasing volume and variety of digital traces, which reveal information about their everyday activity

and movements. In this context, we analyze from a technical and legal point of view the potentials to infer personal sensitive data from mobility information. By demonstrating how the mobility knowledge discovery process blurs the boundaries in the dichotomies anonymous/personal data and sensitive/non-sensitive personal data this work explores its implications and propose to leverage the requirement of a Data Protection Impact Assessment to expand the reach of personal data protection law without impairing the blooming use of mobility data in many public and private domains. For this purpose, some concrete examples based on real experiments will be presented in order to show how easy it is to obtain personal information from this type of data. After a chapter totally reserved for this theme (see 4.4), in each section of a specific work there will be a brief analysis about the potential risks of attack on privacy in that specific context. It's crucial to keep in mind the potential dangers of using personal data even if the scope of a project seems completely ethically safe.

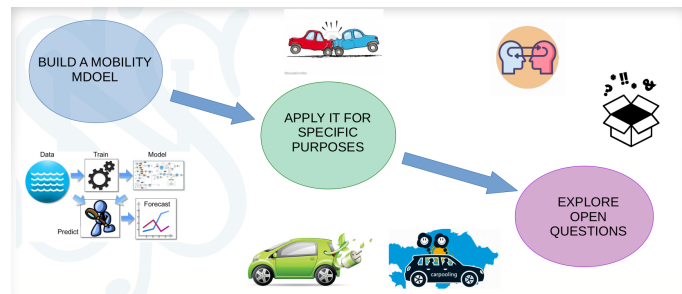


Figure 1.0.1: *My PhD research project in a nutshell*

## 1.1 Building and Representing Individual Mobility Models

Building sophisticated models to understand and predict the individual mobility is crucial to face and solve several issues related to urban planning, traffic congestions and trajectory pattern mining. The objective is to model the mobility of the single individual as a whole, creating a unique, complete picture of it and, in a second stage, adding semantics to the raw data. In order to do that, it is essential to find the most efficient individual mobility representation that satisfies the following requests: effective, understandable and easy to use.

Starting from Individual Mobility Networks, our contributions have focused on the implementation and improvement of some essential aspects for the correct use of these networks. An Individual Mobility Network (IMN) describes the mobility of an individual through a graph representation of her locations and movements, grasping the relevant properties of individual mobility and removing unnecessary details. An essential element of the IMN, which influences its performances, is the definition of *trajectory*.

Movements are performed by users or drivers in specific areas and time instants and

each movement is composed by a sequence of spatio-temporal points. The set of the trajectories traveled by a driver makes the driver’s individual history.

So, finding the right parameters to segment trajectories is essential for any application or goal we want to achieve with IMN.

Therefore, we present a set of user adaptive methods for solving the trajectory segmentation problem, a very common and useful task in mobility data mining, especially in preprocessing phases. Identifying the portions of trajectory data where movement ends and a significant stop starts is a basic, yet fundamental task that can affect the quality of any mobility analytics process. Most of the many existing solutions adopted by researchers and practitioners are simply based on fixed spatial and temporal thresholds stating when the moving object remained still for a significant amount of time, yet such thresholds remain as static parameters for the user to guess. To overcome these weak points we study the trajectory segmentation from a multi granularity perspective, looking for a better understanding of the problem and for an automatic, parameter-free and user-adaptive solution that flexibly adjusts the segmentation criteria to the specific user under study and to the geographical areas they traverse. The solutions proposed take into consideration the overall trajectory of the user, identifying an individual cut time threshold (each user can potentially have a different threshold) and also combining the information coming from the different users through the spatial regions they share. This process yields thresholds for trajectory segmentation which are not only user-adaptive, but also location-adaptive, thus taking into account that a stop at different places might require time intervals of different duration to be considered a significant stay and thus a trajectory cut point. The experiments show that the individual and collective adaptive strategies have a significant impact on the thresholds obtained, which lead to a performance improvement in terms of the metrics defined for this purpose.

### 1.1.1 Graph Embedding

The term *Graph Embeddings* defines the representation of graph properties (as nodes or edges lengths) in a vector or a set of vectors in a low dimensional space. Embedding should capture the graph topology, vertex-to-vertex relationship, and other relevant information about graphs, subgraphs, and vertices.

A graph embedding is a representation of graph vertices in a low-dimensional space. As mentioned before, an Individual Mobility Network (IMN) is a graph representation of the mobility history of an individual that highlights the relevant locations visited (nodes of the graph) and the movements across them (edges), also providing a rich set of annotations of both nodes and edges. Extracting representative features from an IMN has proven to be a valuable task for enabling various learning applications. However, it is also a demanding operation that does not guarantee the inclusion of all important aspects from the human perspective. A vast recent literature on graph embedding goes in a similar direction, yet typically aims at general-purpose methods that might not suit specific contexts. At this purpose, in

Sec. 3.4 we discuss the existing approaches to graph embedding and the specificities of IMNs, trying to find the best matching solutions. We also experiment with some representative algorithms and adapt them to meet the IMNs' particular characteristics better. Tests are performed on a large dataset of real vehicle trajectories.

## 1.2 Applications of IMNs

Individual Mobility Networks can be useful for several different objective. We introduce now some ways to apply this theoretical representation of individual mobility for specific purposes. In particular two different usage are showed:

- IMNs for electric vehicles simulations;
- IMNs for car crash forecasting;

The results and insights obtained with this works opened several research and practical questions that we would like to address in the future and also confirmed us that the IMN representation adopted, appears to be the right tool for enriching the data with higher-level semantics, such as our initial goal was.

### 1.2.1 Electric Mobility

Electric mobility appears to be one of the future ways to make cities more sustainable and improve the quality of life in urban environments. However, when it comes to private vehicles, users need to evaluate how their mobility lifestyle is going to change when their fuel-based vehicle is replaced by and electric one (EV).

While there are many advantages for using electric vehicles, the average user is still worried about changing her life style to EVs. This aversion is to be found in the common belief that moving to an electric vehicle can have a strong impact on their daily life. One of the biggest differences between a fuel-powered vehicle and a battery-powered vehicle lies in the immediate availability of energy needed to charge it. The time required to fill a fuel tank is usually less than a quarter of an hour, while a stop to recharge the battery of an electric vehicle based on the capacity of the battery, can easily take more time.

With this work we aim to propose a process that, through a mix of mobility data analytics, ad hoc trip planning and a simulation, is able to analyze the current fuel-based mobility of a user and quantitatively describe the impact of switching to EVs on her mobility life style. Exploiting the IMNs representation of human mobility, four simulation scenarios are considered, distinguished by the battery recharge options that the user might have in real life. The four options we considered are:

- the user can recharge at the public station, at home and at her work place;
- the user can recharge at the public station and at home;
- the user can recharge at the public station and at her work place;



- the user can only recharge at the public station;

For each scenario we calculate how much battery the user has to charge in each charging option and how much time she waste for charging, as well as how much her original mobility (performed with a combustion engine) is affected by the limits of EVs, evaluating the expected increment in travel times and distances.

In particular we develop a simulation framework for EVs based on a set of individual trips, that mirrors them according to EV constraints and battery recharge opportunities of each scenarios considered. We then define and implement a fast heuristics to compute the best path from an origin to a destination, taking into account the battery constraints and, where needed, computing a deviation to reach a recharge station. And finally we propose a process that, through a mix of mobility data analytics, ad hoc trip planning and simulation, can analyze the current fuel-based mobility of a user and quantitatively describe the impact of switching to EVs on their mobility lifestyle. As we aim to reproduce the study over large sets of users and long periods of time, the process turns to be scalable and completely automatic. The proposed approach turned out to be efficient and takes into consideration the main aspects involved in EV-based mobility: limited driving range, sparse recharge infrastructures, potentially long recharge times, the possibility of recharging at home/work, and so on. The experimentation performed over an Italian region shows how the electrification process is expected to generate only minor issues at the collective level (mainly, marginal increases in distance traveled and time spent at recharge stations), and yet individual users can expect slightly different impacts in they travel & refuel habits.

In general, we observe that the highest increases are observed when only public stations are available for recharging, which are strongly reduced by recharges at home, and slightly less by recharges at work. When both options are available, their synergy actually produces slight improvements.

### 1.2.2 Crash event risk prediction

The massive and increasing availability of mobility data enables the study and the prediction of human mobility behavior and activities at various levels. In Section 4.3, we tackle the problem of predicting the crash risk of a car driver in the long term. This is a very challenging task, requiring a deep knowledge of both the driver and their surroundings, yet it has several useful applications to public safety (e.g. by coaching high-risk drivers) and the insurance market (e.g. by adapting pricing to risk). We model each user with a data-driven approach based on the IMNs. The basic objective is not only to recognize the real risk level of a customer but also to understand possible causes [125]. Hence, we aim to reach two distinct results:

1. To predict the customer's risk score: given a car insurance customer, provide a risk score relative to the near future, e.g., the next year or the next month. We expect this estimate to be much dependent on how the customer drives, as well as on the conditions of the surrounding environment [77], [20]. Accordingly,

the methodology we propose is based on the computation of individual driving features, describing how much the user drives and how much dynamically, also related to the general characteristics of mobility in the places that the user visits.

2. To infer risk mitigation strategies: given a car insurance customer and her risk score, we would like to identify the characteristics of her driving behavior [307] that determine her risk score. From a prescriptive viewpoint, this is going to provide to the customer indications of how to reduce her risk score, with benefits for her (in terms of safety and insurance costs) and the insurance company (in terms of costs for accidents). The approach under investigation queries the predictive models adopted for understanding which features decided for the prediction [238].

However, since raw mobility data collected by car telematics and car insurance companies is limited to positions and events of the vehicle [191] with no vision of what happens around it, or further structured and complex information, in order to achieve our goals we need to augment the individual data with additional knowledge. About that, we proposed a solution consisting in extracting sophisticated features of the user's mobility, able to capture not only basic characteristics of her mobility, but also higher-level information derived from a network view of her mobility history as well as contextual knowledge directly inferred through analysis of the collective data of all users. On top of such features, machine learning models can be trained and successfully employed. Experiments on real data showed that our solution outperforms basic solutions based on state-of-art features, and a inspection of the prediction models through explainable AI methods allowed us to identify a few representative features associated with crash risk.

## 1.3 Explore Open Questions

### 1.3.1 Transfer Learning

In this section we underline the importance of studying geographical transfer learning in the mobility context and briefly introduce the achieved goal of this thesis. The objective is make mobility models extracted on a geographical area transferrable to other locations, i.e. applicable to data over a different territory achieving good performances. In particular, a process for extracting sophisticated descriptive features of a geographical area is provided, then it is exploited to analyze their relations with model transferability, and finally a set of geographical model transfer strategies is defined and tested over a prediction problem of interest. In particular, our goal is to verify the feasibility of a model transfer (a machine learning model is trained in the source domain and then transferred to the target domain) in the prediction of urban traffic. In our context, both source and target domains are cities with their mobility. The validation of the methodology and the case study are developed on Tuscany dataset.

The basic idea is that cities that are similar can be represented by the same model more easily than very different cities. For instance, a highly populated city with heavy traffic and users that frequently make long trips is expected to have mobility dynamics very different from small, country-side cities with low traffic. Therefore, any model capturing the first city (for instance a traffic prediction model) will probably be not very fit for making predictions in the second, even if the model itself was built in a location-oblivious way in order to be relocatable – i.e. it does not directly depend on the spatial position of the areas to model, and therefore it can be applied to any location. The approach proposed follows the similar-cities principle mentioned above and it is exploited inside the car crash prediction project. The approach is developed in three steps: first, a method to compare the similarity of cities is proposed, next, for each city a traffic prediction task is defined, which is approached through a standard machine learning solution (XGBoost, regression); finally, the prediction model of a city is applied to make predictions in each of the others, aiming to test whether similar cities show a better transferability of their models. We define an array of geographical transfer learning strategies based on the data and the models available in certain areas that can be applied to target areas individually or as an ensemble.

**Comparative City Studies through City Indicators.** For our transfer learning purposes we rely on a set of city indicators that can be retrieved for every area to evaluate the similarity between two or more areas. Classifying cities and other geographical units is a classical task in urban geography, typically carried out through manual analysis of specific characteristics of the area. The primary objective of our work is to contribute to this process through the definition of a wide set of city indicators that capture different aspects of the city, mainly based on human mobility and automatically computed from a set of data sources, including mobility traces and road networks. The secondary objective is to prove that such set of characteristics is indeed rich enough to support a simple task of geographical transfer learning, namely identifying which groups of geographical areas can share with each other a basic traffic prediction model. The experiments show that similarity in terms of our city indicators also means better transferability of predictive models, opening the way to the development of more sophisticated solutions that leverage city indicators.

Our work proposes several different strategies that exploit such weights in different ways, and provides an empirical comparison to find out the best one in terms of prediction performances. However, even if geographical transfer learning is a poorly explored area and the results discussed here represent only a first step, their are convincing that searching the optimal set of city indicators it is possible to have an efficient model transferability. More sophisticated solutions could be obtained by an appropriate combination of standard techniques (butions) and context-aware methods.

### 1.3.2 Explainability

As AI systems and machine learning-powered tools proliferate in our everyday life, both practitioners and critics are becoming more vocal about the need to know how they produce the outcomes they do. Moreover AI is fast becoming embedded in industries, economies and lives, making decisions, recommendations and predictions. These trends mean it's critical to understand how AI-enabled systems arrive at specific outputs. It's not enough for an AI algorithm to generate the right result knowing "the reason why" is now a business fundamental. The process needs to be transparent, trustworthy and compliant far removed from the opaque "black-box" concept that has characterized some AI advances in recent times. At the same time, these advances should not be stifled: AI's velocity underscores organizational competitive advantage in multiple use cases. From an AI system providing personalized real-time medical information to financial traders using AI algorithms to make deals within milliseconds, the solution might be found in explainable artificial intelligence (XAI).

In few words XAI is still one of the main challenges computer scientists are focusing on, despite the development of increasingly sophisticated solutions to understand the responses of machine learning models. That is why part of this thesis is devoted to study and test some optimal explainability solutions for our use cases. In particular, for the Car Crash Prediction work (described with all the details in Section 4.3) we decided to adopt the SHAP solution to understand the impact of the single features on the final outputs. SHAP, alias SHapley Additive exPlanations [181], is a local-agnostic explanation method that calculates feature importance based on the Shapley values. The goal of SHAP is to explain the prediction of an instance  $x$  by computing the contribution of each feature to the prediction. The SHAP explanation method computes Shapley values from coalitional game theory. The feature values of a data instance act as players in a coalition. Shapley values tell us how to fairly distribute the "payout" (= the prediction) among the features. We obtained very interesting explanation results following this approach.

## Setting the Stage

In this chapter we provide a background of the main topics treated in this thesis. We recall basic notions of data mining and human mobility, and also present an overview of user profiling models and methods.

In particular, we analyze the current state-of-the-art with respect to personal data models defined to characterize mobility habits. Finally a detailed section about Transfer Learning is shown in order to introduce the general aspects of the topic. The thesis will be focused more on the geographical transfer learning theme but a complete introduction of the topic is needed to contextualize. Besides that we mention the privacy and ethical aspects of mobility data trying to underline the dangers behind a misuse of human mobility data and analysing also what explainability should do to avoid this kind of issues.

### 2.1 Data Analytics Landscape

By Wikipedia definition [324], *Analytics* is the systematic computational analysis of data or statistics [253]. It is used for the discovery, interpretation and communication of meaningful patterns in data. It also entails applying data patterns towards effective decision-making. It can be valuable in areas rich with recorded information; analytics relies on the simultaneous application of statistics, computer programming and operations research to quantify performance. Organizations may apply analytics to business data to describe, predict, and improve their business performance and decision making.

Data analytics has become an extremely important and challenging problem in disciplines like computer science, biology, medicine, finance, and homeland security. This problem involves several aspects: first of all large volumes of data must be collected and stored relying on cleansing and filtering techniques; next, sophisticated algorithms are used to analyze the data and extract “useful” information; finally, various user interfaces can be used to visualize and understand the data.

Analyzing large amounts of data has become an extremely hard task, as the quantity

of information in the world is large and increasing exponentially. For example, various social networks like Facebook generate terabytes of data per day in the form of photos, videos, wall posts, etc., and will generate significantly more data in the near future. The size of today's data is unprecedented and cannot simply be analyzed with conventional data management techniques.

Nevertheless, being able to efficiently “make sense” out of big data is becoming even more important than ever in various areas. In computer science, web-scale data needs to be analyzed in order to understand global trends and user behavior. In biology, interpreting massive amounts of DNA and RNA sequencing data is essential for understanding complex biological systems. Already, the explosive growth of sequencing data has exceeded the growth rate of storage capacity.

In medicine, health devices generate huge amounts of data that reflect the condition of patients by monitoring their sleep, heart rate, and other health conditions. In finance, the stock market generates immense quantities of transaction data that can help companies maximize profit. While we have only listed a few examples, there are many other areas that are starting to exploit large amounts of information as well. One of the main challenges in data analytics is to collect data from multiple sources and combine them together so that data analysts can access and manipulate the information in a unified way.

## 2.2 Background on Mobility Data Analysis

In this Chapter we recall the basic notion of data mining and we present an overview of human mobility models and methods.

### 2.2.1 Mobility Data Science

Geographical movements across different countries are a natural phenomenon in human history [323]. Without necessarily thinking of big events, like the expansion of the Roman Empire or the conquest of the Americas, human movement has been the common thread slowly shaping our history worldwide. Even the idiom in which this work is written is an Indo-European language, which, as the word says, is the result of the displacement in centuries of civilizations, presumably tribes, from the Asian continent towards Europe. Mobility is a behavioral phenomenon, which has no meaning without taking into consideration who performs it. In [202] Marchetti defined few anthropological invariants in the way humans move worldwide, which basically points out the concepts of distance, mass and time. However today we know that human movement occurs for different reasons, hence it must necessarily be classified in different categories, mainly depending on the purpose. As an example, people migrate in order to improve their living conditions, or because of social distress [160], wars and natural catastrophes, but this phenomenon cannot be studied in the same way as we study people going to work in urban areas in the morning or like building evacuations in a fire drill. It is not only a matter of a different scale, it is a matter of behaviors and purposes. This will be reflected in

the way we, as scientists, address the study and modeling of human flows depicted by these behaviors. Every movement is linked to a single person and this person has a story to tell. Understanding the human mobility history is essential to have the bigger picture of human footprints representing the sum of behaviors. Mobility is undoubtedly a critical phenomenon in urban environments. In fact, it can be considered as one of the most important mechanisms underlying the structure and dynamics of cities. Indeed, cities are places where intensive buying, selling or exchanging goods is taking place, where individuals commute to work or meet with other individuals. An obvious means to achieve all this is transportation. Here is where technology enters the picture via the average and maximum velocity of different transportation modes. Data is central to our future and technological advancement. As we continue to urbanize and gather more data about our mobility patterns and urban systems, the challenge is not simply the amount of data, but how we manage it, critically understand its quality, and utilize findings to address our cities' most pressing challenges.

### 2.2.2 Data Mining and User Profiling

The process of digging through data to discover hidden connections and predict future trends has a long history. Sometimes referred to as "*knowledge discovery in databases*", the term "*data mining*" was not coined until the 1990s. But its foundation comprises three intertwined scientific disciplines: statistics (the numeric study of data relationships), artificial intelligence (human-like intelligence displayed by software and/or machines) and machine learning (algorithms that can learn from data to make predictions). Over the last decade, advances in processing power and speed have enabled us to move beyond manual, tedious and time-consuming practices to quick, easy and automated data analysis. The more complex the data sets collected, the more potential there is to uncover relevant insights. Retailers, banks, manufacturers, telecommunications providers and insurers, among others, are using data mining to discover relationships among everything from price optimization, promotions and demographics to how the economy, risk, competition and social media are affecting their business models, revenues, operations and customer relationships.

Even the study of human mobility can be observed from the data mining point of view, whose interest on the analysis of human movements, in form of trajectories, generated the new sub-field called mobility data mining [114]. While statistical physics seeks to discover which global models describe human mobility, data mining is aimed at discovering local mobility models. The former uses statistical laws which regulate basic quantities, the latter prefers to use micro-laws adjusting similarity or behavioral regularities in sub-populations.

Thanks to the improvements in mobile communication and positioning technology we can describe the large amounts of moving objects data in form of trips (also called trajectories as we will define better in the next chapters). In this regard, the Data Mining analysis step of the Knowledge Discovery process in Databases (or

KDD) [291] is the most intuitive and attractive approach to describe the structure of trajectories and to extract frequent spatio-temporal patterns: indeed, mining the spatio-temporal patterns means searching for concise representations of interesting behaviors of single or groups of moving objects.

*User profiling* term refers to the process of construction and extraction of a personal data model representing the user behavior generated by computerized data analysis. A personal data model highlights the systematic behaviors expressing the repetition of habitual actions, i.e., personal patterns. These patterns can be expressed as simple or complex indexes, behavioral rules, typical actions, etc. Users' profiles have several objectives: from one side they are employed to analyze and understand human behaviors and interactions. On the other side, they are exploited to make predictions, give suggestions, and to group similar users. Profiles can be classified as individual or collective according to the subject they refer to [139].

Despite user profiling has been deeply studied in fields like economy or in the World Wide Web, nowadays it is still an emerging field of research with respect to mobility, thanks to the great availability of GPS and GSM data we have.

A *personal profile* is a data model built considering the data of a single person. This kind of profiling is used to create a complete picture of the single individual enabling unique identification for the provision of personalized services. A strong point of individual profiling is that the computation is generally not time and space consuming because the data of a single user are limited. However, this limitation can also negatively affect, indeed it could not consider a valuable pattern recognized by other users because it is not enough systematic for the individual. For example, if a user  $u$  goes occasionally to the movie theatre on Saturday night, this trip could not be personally considered as a routine or a pattern if compared with the Home-Work-Home movements. However, if many people move from the same city of  $u$  to the same movie theatre on Saturday night, then this generates a pattern collectively recognized.

At the contrary, services are usually based on *global profiling*: people are classified or segmented within a certain class, based on the fact that their behavior aligns with a data model formed by global patterns constructed on the basis of a massive amount of data. A weakness of global profiles is that they do not consider personal patterns because only the general patterns recognized by all the users emerge from the mass. Furthermore, it can be computationally hard to extract global profiles because a large amount of data must be considered all at the same time. Finally, global profiling requires every user to share all her raw data at the most detailed level.

Conversely, *collective data models* are about personal models generated by individual profiling that are aggregated without distinguishing the individuals. The difference with global models is that the collective profiles consider the personal patterns extracted from the individuals as a unique model, while in the global ones the patterns extracted are those related to the data of all the individuals, representing the behaviors of the mass. Finally, we can refer to combined or hybrid models when two



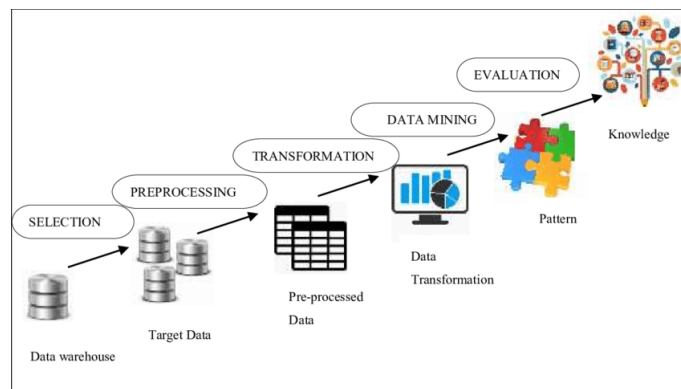


Figure 2.2.1: *The process of Knowledge Discovery in Databases.*

or more of the previous ones are merged in some way. Different models are generally combined to overtake weak points and to exploit strong points. An example of combined approach is the hierarchical one. It uses the individual profile as long as it is useful to solve the problem of a certain person, then it switches to collective patterns. Another example in stochastic applications is to mix the two different profiles according to a certain parameter or probability.

### 2.2.3 Human Mobility Data

Modeling human mobility in a city is tightly related to geographical patterns and spatial distributions. Understanding individual movements brings useful insights for a variety of applications, such as urban planning [220], security [165], migration studies [37], disease spread and traffic prediction (transportation planning)[181]. Researchers have tried to use surveys [195] from travel or tourist centers in a traditional way to study mobility patterns; however, thanks to new technologies, finding a dataset to analyze people’s mobility is not a big concern anymore.

Recently, a significant effort has been made with different types of datasets, including phone call records (CDR)[63], WiFi or RFID [340], global positioning system (GPS) [280] and location-based social network (LBSN) data, in order to obtain useful information from geographical movements. In this area, researchers have tried to tackle various questions as *Does human mobility follow any model or pattern? Is it possible to extract significant patterns to define mobility models? If yes, how?*

The various methodologies applied to study human mobility have changed substantially in the last century. Availability of digital data on large geographical and time scales and enhanced multidisciplinary across different fields are only few of the ingredients that have radically changed the way we study human mobility and our understanding of the heterogeneous set of phenomena that go under this same label. In the following, we will review the data and methodologies used over the past century until the present.

The quick evolution and wide diffusion of technologies for the localization of de-

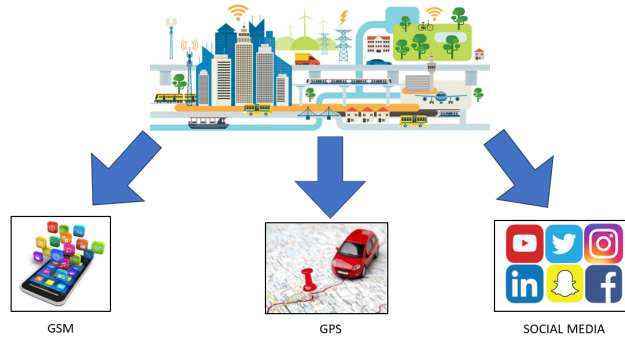


Figure 2.2.2: *Some of the most used mobility data in data mining.*

vices (especially smart-phones and vehicles' GPS) as well as location-based services, is leading to the production and collection of large and diversified traces of human mobility, every day more detailed and pervasive. These traces potentially contain a huge amount of information that might allow to infer models of human mobility spaces at unprecedented levels of precision and depth. They would be key enablers of many applications, ranging from monitoring urban traffic features to reconstruct inter-city mobility demands and region-scale structures, which could help in making modern urban spaces more sustainable, efficient and comfortable for citizens. The ubiquitous applications of mobile and handheld devices lead to an explosion of multi-source data correlated with human mobility, providing a novel and comprehensive view to study urban human mobility patterns. These datasets are collected passively, for example, call detail records (CDR), credit card, smart card, their purpose is not collecting mobility data but to register all the transactions. But these multi-source heterogeneous datasets record people's travel trajectories and imply the potential mobility patterns. In smart cities, we can collect a variety of data leveraging network and wireless communication technology, which captures people travel trajectories in their daily life and depicts the spatiotemporal characteristics of urban human mobility.

In this section, we introduce the main data types used for human mobility research as shown in Fig. 2.2.2 and compare their advantages and disadvantages.

### GPS Data

The GPS (Global Positioning System) is a satellite navigation system that provides geolocation and time information to a GPS receiver anywhere on or near the Earth where there is an unobstructed line of sight to four or more GPS satellites. The GPS does not require the user to transmit any data, and it operates independently of any telephonic or Internet reception, though these technologies can enhance the usefulness of the GPS positioning information.

Each satellite continually broadcasts timing signals which specify the moment the message was transmitted and the precise positioning information. The receiver com-

putes the distance to each satellite by determining the transit time of each message it receives. These distances along with the satellites' locations are used to calculate the position of the receiver, in form of latitude, longitude and other information like elevation, direction and speed.

GPS-enabled devices provide us with all the required information for trajectory tracking, giving access to accurate, time-stamped locations of each tracked moving point. Today GPS receivers are not very expensive, so it is possible to find them embedding in many devices we use everyday: smart phones, vehicles, smart watches, and so on. They allow to track human mobility very well.

In this thesis we mainly use a massive real-life GPS dataset, namely the Octo dataset, obtained from tens of thousands private vehicles with on-board GPS receivers. The owners of these cars are subscribers of a pay-as-you-drive car insurance contract, under which the tracked trajectories of each vehicle are periodically sent to a central server for antifraud and anti-theft purposes. This data set has been shared for research purposes by Octo Telematics Italia S.r.l. [2], the leader for this sector in Europe.

The GPS device automatically turns on when the car starts, and the sequence of GPS points that the device transmits every 30 seconds to the server creates the global trajectory of a vehicle [230]. When the vehicle stops no points are registered. These stops are then exploited to split the global trajectory into several sub-trajectories in the most meaningful way, that correspond to the travels performed by the vehicle. Clearly, the vehicle may have stops of different duration, corresponding to different activities.

In the next chapters we will deal with the segmentation of trajectories in detail, showing the strategies adopted and implemented.

The principal data source for the works presented in this thesis consists of GPS trajectories of private car in Rome, London and Tuscany, provided within the scope of the EU Horizon 2020 "Track & Know" programme.

## **GSM Data**

Nowadays mobile phones are perhaps the most used technological device by all citizens. In the Italian film "Perfetti Sconosciuti" ("Perfect Strangers") mobile phones are defined as "the black box of our life" because they know almost more about us than ourselves. Individuals carry them in their daily routine offering a good proxy to study structure and dynamics of human social behavior. Indeed, phone records capture information about both social links and human displacements: each time we make a call a social relationship of some kind is expressed, and the tower that communicates with our phone is recorded by the carrier, effectively tracking our position. The GSM (Global System of Mobile Communications) is the most popular standard for mobile phones in the world, nowadays used by more than 1.5 billion people across more than 210 countries and territories. The versatility of GSM systems make international roaming very common between mobile phone operators, enabling subscribers to use their phones in many parts of the worlds.

GSM network is compound by a number of base stations, each responsible for a particular spatial area (known as “cell” or “tower”). In this way, for each GSM-enabled device it is possible to collect information about the base stations at different timestamps assuming its movement. A GSM-enabled device can be tracked by collecting all the communication signals transmitted (cell, signal strength) between this device and the networks infrastructure or by studying the log of the outgoing calls (UserID, data and time of the call, duration of the call, the cell where the call began, the cell where the call ended). However, in both levels the accuracy of trajectories that can be collected is very low since GSM data are coarse in space because they express locations with the granularity of a cell tower sector, providing an uncertainty approximately 1.5 square meter or even higher in less populated areas. The most detailed level of available information is the network cell and not a spatial point. Data about calls are generally stored in form of CDRs (Call Detail Records), describing each phone call performed by the users. Each call is represented by a tuple with timestamp, caller and callee identifiers, duration of the call, and the geographical coordinates of the tower serving the call. Such mobility traces are more accurate in densely populated areas, where much more phone towers are installed to carry the heavier load. This means that in rural areas, where a single tower usually covers several square kilometers, short movements are not collected. Besides the mobility dimension, GSM data also provide us the social dimension, since we can reconstruct from the calls a weighted directed call graph, where nodes are users and edges are interactions between users. The weight of an edge can be either the total number of calls or the total duration of calls between the users during the period of observation.

Even though mobile phones are carried by the same person during the daily routine offering a good proxy to capture individual trajectories they do not provide an accurate spatial information. On the contrary, GPS traces provide high resolution location data, storing the geodetic coordinates with an average sampling rate of few seconds. Mobile phone data pertain to general mobility while GPS data pertain only to cars, yet they are much less detailed than GPS trajectories. Moreover the GSM sampling rate is usually very low (it depends on the calls duration) so the granularity is often very poor.

The choice of using only GPS data for the work of this thesis is therefore legitimate since the studies were made only using cars travels also considering that our main applications are related to vehicle mobility.

## **Social Media Data**

Nowadays, the ever growing popularity of social media systems and the ubiquitous use of smart devices is generating a huge amount of data that is freely available and that covers all aspects of user behavior and life, such as the behavior in social media systems and Internet in general, economic activities, visited places, and user preferences and opinions[199]. In such a scenario, users represent sensors that continuously generate a stream of data that can be exploited by everyone to infer information about collective user behaviors and to analyze what the users want to

disclose about the so-called spatial self [272].

Most of the social media platforms use APIs to provide access to their data. Usually the APIs make multiple functions available based on a set of parameters that allow to perform several activities, such as to download a stream of data in real time, to specify a time window, to specify a set of keywords, or to specify a bounding box. The use of these functions represents a valuable method to create a collection of geolocated data coming from social media.

Social media check-ins data is also a valuable source collected by social network providers. More and more users record their social connections and geographical locations through Twitter, Facebook, Foursquare, and Flickr. The geotagged data is made up of coordinates, time, photos, and comments. Therefore, the movement trajectories of users can be obtained from the sequence of published locations. By analyzing the information, we can calculate some important metrics and identify patterns: radius of gyration, jump length, visit frequency. Scellato et al. [270] investigate three location-based social networks from a perspective of network science and discover several small-world properties; Itoh et al. [149] explore social media data on Twitter and smart card data on the Tokyo subway and extract abnormal situations on mega-city subway networks; Ni et al. [218] propose a hybrid approach to predict the subway passenger flow based on social media activities and improve the prediction accuracy. Compared with other data sources, social media data has its unique characteristics such as more social information, which provides a multi-dimensional view of studying human mobility patterns. But the data is uploaded personally, so it is likely to have also false information.

## 2.2.4 Mobility Metrics

In this section, we discuss the validation methods and metrics which are used for the human mobility models. We make a distinction between trajectory-based metrics and network-based metrics. The first category is the result of individuals' mobility traces, thus regards the characteristics of each single trip performed by an individual: duration, length, distance, and so on. The second ones involve considering the human mobility as a complex network and study it as a graph. Its performance is based on metrics showing the effects of human mobility in terms of networks. Both kind of metrics are important and complementary for our purposes.

### Trajectory-based Metrics

Trajectory is the most intuitive expression of human mobility and contains many features and laws of human behaviors.

- **Distance** is the most fundamental quantity to measure about paths followed by an individual. The Euclidean distance is the most common distance used to obtain the trip length for origins and destinations' coordinates while the geodesic one is the most popular for longitude and latitude attributes [328].

- **Duration** is the elapsed time during a journey, i.e. the difference between the departure time and the end time. The metric reflects the time spent in traveling, capturing the basic characteristics of human mobility [278].
- **Radius of gyration** is the root mean square distance of an individual's location from her center of mass of the motions [118]. It is defined as:

$$Radius_g = \sqrt{\frac{1}{n} \sum_{i=1}^n (r_i - r_0)^2} \quad (2.2.1)$$

where  $r_i$  represents the  $i$ th position of a user, and  $r_0$  is the center of mass of her travel trajectories,  $r_0 = \sum_{i=1}^n r_i/n$ . Radius of gyration is an important metric to measure the typical distance traveled by people and it has a correlation to the mutual distance of visited locations and the total number of visits. Radius of gyration provides a more complete view of how individuals travel around their centers of mass.

- **Interval** is the elapsed time between two consecutive trips and for instances it depends on how the segmentation of the trajectories is done. It often represents the time spent in a location or in a taxi operation case, the time taxi drivers cruise for a new passenger after passengers get off. So the metric measures the time of non-occupied state for taxi and reflects travel demands indirectly. Obviously, a taxi with shorter interval will be more efficient.
- **Entropy** is the most basic metric measuring the degree of predictability of an individual's human mobility [282]. It is related to the frequency of visitation and capture the full spatiotemporal pattern in a person's travel trajectory. People with a high entropy show a low heterogeneity of visiting location. We formulate the metric as follows:

$$Entropy_g = - \sum_{i=1}^n p_g(i) \log_2 p_g(i) \quad (2.2.2)$$

where  $p_g(i)$  denotes the historical probability that the user  $g$  visits the location  $i$ , characterizing the heterogeneity of visiting patterns.  $n$  represents the number of visited locations.

### Network-based Metrics

The different types of data originated by several sources provide a novel view of studying human mobility from a perspective of complex network [118]. A solution to model human behaviors is by graph theory that allows to explore the characteristics of human dynamics in an easy way. In the graph, the nodes represent a set of locations or POIs visited by people, and an edge represents the related pairs of locations in historical trajectories. Furthermore, it can be determined which node is the most influential in a network and how well a network is optimized with respect to network performance.

- **Degree Centrality:** is defined as the number of ties that a node has [73].

$$C_D(i) = \sum_{j=1}^n a_{ij} \quad (2.2.3)$$

where  $C_D(i)$  denotes the degree centrality of node  $i$ .  $\sum_{j=1}^n a_{ij}$  is the number of links between node  $i$  and the other nodes ( $n - 1$ ). Degree centrality identifies the most important nodes in a network. A node with high degree centrality has more links than others, which is used for understanding human dynamics [250].

- **Betweenness centrality** is a measure of centrality in a graph based on shortest paths. For every pair of nodes in a graph, there exists at least one shortest path between the vertices such that the number of edges that the path passes through is minimized. The betweenness centrality is computed as follows:

$$C_B(k) = \sum_{i \neq j \neq k \in V} \frac{\alpha_{ij}(k)}{\alpha_{ij}} \quad (2.2.4)$$

where  $C_B(k)$  represents the betweenness centrality of node  $k$ .  $\alpha_{ij}$  is the total number of shortest path from node  $i$  to node  $j$  and  $\alpha_{ij}(k)$  is the number of these paths through node  $k$ . According to its definition a node with high betweenness centrality greatly impacts the transmission of information flowing between others [78]. Therefore, it is important to prevent nodes with high betweenness centrality from failing in a network.

- **Closeness centrality:** is a measure of centrality in a network, calculated as the reciprocal of the sum of the length of the shortest paths between the node and all other nodes in the graph. It is formulated as follows:

$$C_C(i) = \frac{1}{\sum_{j \neq i} d(j, i)} \quad (2.2.5)$$

where  $C_C(i)$  is the closeness centrality of node  $i$ .  $d(j, i)$  is the sum of node  $j$  and node  $i$ . A node with a high closeness centrality is clearer to all the other nodes in the network, and its movement law reacts to the motion law of other nodes. Therefore, the patterns of high closeness centrality node are critical to predicting human mobility in the future.

### 2.2.5 Individuality vs Collectivity

Human mobility refers to the geographic displacement of human beings, seen as individuals or groups, in space and time. This displacement is described by an origin, a destination, and a specific trajectory in between. There are many different ways to categorize the mobility models. Social scientists categorise mobility (spatial mobility) by its utility: mobility that happens inside the place of residence; migration

(international and inter-regional mobility); travel with the purpose of tourism or business and day-to-day journeys such as commuting and running errands [183].

Physicists describe mobility by spatiotemporal scale: long-term mobility that is likely to cover large displacement, e.g., migration, and short-term mobility whose displacement is constrained by 24 hours in a day. They see mobility as a diffusion process that is characterised by both randomness and regularity [281].

In transport geography, researchers see the mobility as individual behaviour that generates flows of population. At the individual level, the mobility trajectory is a time series of visits to various locations. Individuals' mobility trajectories can be aggregated to study the flows of people travelling between different locations/regions. Depending on the spatiotemporal scale of the aggregation, an origin-destination matrix (OD matrix) can be constructed with the origins and the destinations of all trips. Using this taxonomy, this chapter continues to review the literature with these two perspectives: individual and collective mobility models.

### Collective Mobility Models

Population mobility models mainly focus on the mobility patterns of collective people between two regions in urban scenarios. This type of model can predict the distribution of migratory flows at some time in the future according to the population of regions.

- **Gravity model:** The gravity model is inspired by Newton's law of gravitation [346]. It assumes that the volume  $T_{ij}$  of people flow between locations  $i$  and  $j$  is in direct proportion to the population size of the two locations and is in inverse proportion to the distance  $d_{ij}$  between them [14]. The gravity model is defined as

$$T_{ij} = x_i^\alpha x_j^\beta / f(d_{ij}) \quad (2.2.6)$$

where  $\alpha$  and  $\beta$  are two exponents.  $x_i$  and  $x_j$  denote the population of locations  $i$  and  $j$ .  $f(d_{ij})$  is a function of distance between origins and destinations, which can approximate the empirical data such as power law and exponential.

In addition to estimating human flow, the gravity model is used for measuring the intensity of interaction (calls and trade) between two regions. The form of the function is tunable according to the scenarios. In urban planning, the distance may not measure the travel cost between specified locations, and trip duration may be a better alternative.

The gravity model is widely used in public transportation management [161], geography [97], and social economics [205].

- **Radiation model:** Simini et al. [279] extend the gravity model introducing the radiation model, a stochastic process able to capture the local mobility decisions of individuals,



analytically deriving mobility fluxes that require as input only information on the population distribution. The radiation model predicts mobility patterns in good agreement with mobility and transport patterns observed in a wide range of phenomena, from long-term migration patterns to communication volume between different regions. Given its parameter-free nature, the model can be applied in areas where mobility measurements are not available, significantly improving the predictive accuracy of most of the phenomena affected by mobility and transport processes. It is defined as:

$$T_{ij} = p_i x_i x_j / (x_i + N_{ij})(x_j + N_{ij}) \quad (2.2.7)$$

where  $p_i$ ,  $x_i$ , and  $x_j$  are the same to the representation of above-mentioned models.  $N_{ij}$  represents the total number of population between locations  $i$  and  $j$ . The specific range is a circle with location  $i$  as its center and  $d_{ij}$  as its radius, excepting for locations  $i$  and  $j$ .

### Individual Mobility Models

Individual mobility models mainly characterize the mobility patterns of individuals based on multi-source spatio-temporal trajectory datasets. This type of model introduces some travel features to modeling human mobility, i.e., radius of gyration, entropy, trip interval, and trip displacement.

- **Brownian motion:** Brownian motion originally explains the motion law of particles hovered in a liquid [84] [95]. The process of motion is accompanied by multiple collisions and stochastic steps and directions. The probability density function is represented as

$$P(x, t) = \frac{1}{\sqrt{2\pi t\sigma^2}} e^{-\frac{(x - \mu kt)^2}{2t\sigma^2}} \quad (2.2.8)$$

where  $\sigma^2$  and  $\mu$  denote the variance and mean of the stochastic distances. Brownian motion is also a kind of random walk. We can define Brownian motion as a limit of the non-continuous symmetrical random walk. Einstein et al. [302] illustrate the principle of Brownian motion in detail and provide a strong evidence of the existence of atoms and molecules.

- **Lévy flight:** A Lévy flight is a random walk in which the step-lengths have a Lévy distribution, a probability distribution that is heavy-tailed. When defined as a walk in a space of dimension greater than one, the steps made are in isotropic random directions [326]. The flight step  $l$  is approximated by a power law with the characteristics of a fat-tailed distribution.

$$P(l) \sim l^{-(1+\beta)}, 0 \leq \beta \leq 2 \quad (2.2.9)$$

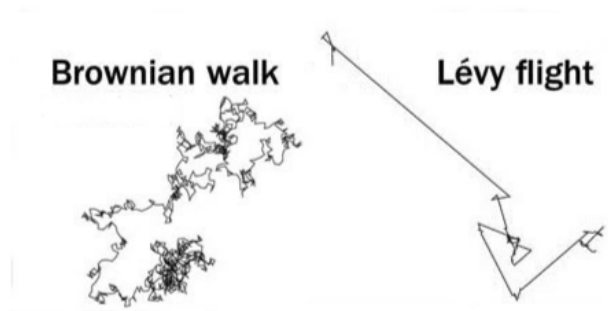


Figure 2.2.3: *Brownian motion (left) is described as a random walk in which all the steps give the same contribution. Lévy-flight (right) occurs when the trip is dominated by a few very large steps*

where  $\beta$  is an exponent of displacement. About the intrinsic mechanisms of these method we can cite some examples. Gonzales et al. [118] prove that the power-law distribution of jump length is related to the characteristics of individual mobility and population heterogeneity. Han and Jiang et al. [134] demonstrate that human mobility is impacted by the topology of urban transportation systems. Jiang et al. [156] also find the above-mentioned factor by analyzing the human mobility in urban road networks.

## 2.2.6 Human Mobility Patterns

Human mobility patterns reflect many aspects of life, from the global spread of infectious diseases to urban planning and daily commute patterns [301]. In recent years, due to the increasing availability of mobile-phone records, global-positioning-system data and other datasets capturing traces of human movements numerous statistical patterns in human mobility have been revealed. The efforts to mine significant patterns within large-scale, high-dimensional mobility data have solicited use of advanced analysis techniques, usually based on machine learning methods. These empirical observations suggest that human mobility are barely random, but follow predictable rules. For instance, Song et al. [281] proposed the Exploration and Preferential Return (EPR) model, which does not fix the set of most visited locations but allows them to emerge naturally during the evolution of the mobility process. They propose two basic mechanisms that together describe human mobility: exploration and preferential return. Exploration is a random walk process with truncated power law jump size distribution. Preferential return reproduces the propensity of humans to return to the locations they visited frequently before [341]. On the other hand, gravity models have a long history of use in describing and forecasting the movements of individuals, goods and services. This class of urban models, derived from Newton's law of gravity, characterize the distribution of trips between locations and predict the degree of migration interaction between two of

them. About that the work of Zipf [346] provides a theoretical motivation for movement between two cities  $a$  and  $b$  being governed by a  $P_a P_b / d$  relationship, where  $P$  is the city population and  $d$  is separation distance. In other words, the author establishes that movements follow a sort of “gravity law”: the probability that an individual or a group of individuals move between two locations is inversely proportional to the separation distance of the two locations.

More often mobility pattern methods are used in the context of a particular application, providing a service that is based on data about a particular entity that is being tracked and modeled. We can group applications by classes, aiming at representing domains of functions that can be accomplished by each application while using particular datasets [301]. Through an iterative process, the studies are classified in three main categories according to the aim of the model that is used by the application: user modeling, place modeling, and trajectory modeling. Figure 2.2.4 exemplifies the three application classes focusing on modeling the (a) human user, (b) places the user moves through, and (c) trajectories of the user movements. User modeling applications analyze the mobility of a single user (or object) for extended periods of time (Fig. 2.2.4a). In such applications, the model can predict where a particular user will be at different times of the day. For example, in homeland security applications, targeted users can be modeled by the distribution of their geographic locations over time in order to trigger an alarm if an abnormal situation occurs.

Place modeling applications analyze the characteristics of a geographic location or a set of locations. For example, in Fig. 2.2.4b, the model can predict the number of incoming and outgoing people in a place (say a large store), profile its traffic, and classify the type of place according to the mobility patterns of people around it.

Trajectory modeling applications analyze a set of spatial-temporal points that reflect a trajectory, defined as a movement pattern through a set of locations of a single object or a set of objects and time. In contrast to user modeling, in trajectory modeling, the identities of the moving objects are not necessarily a factor in the analysis; thus, for example, all the moving objects along the modeled trajectory can be analyzed aggregately. In contrast to place modeling, the entity in trajectory modeling is a route between geographic locations, rather than a single location. For example, Fig. 2.2.4c visualizes a trajectory that may be used in modeling road segments or road networks by an application that predicts traffic conditions.

## 2.2.7 Personal location detection

One of the key tasks in mobility data analysis (and a necessary preprocessing step for many applications) is detecting the locations of users [16]. The objective is to identify the users’ personal location, i.e., the areas where users perform their activities, based on the analysis of the locations (essentially, GPS points) where they have stopped. Examples of locations are home, workplace, supermarket, gym, fuel station, etc. More precisely, given a set of users GPS stop observations, i.e., coordinates in which the users have stopped, the location detection problem consists in grouping

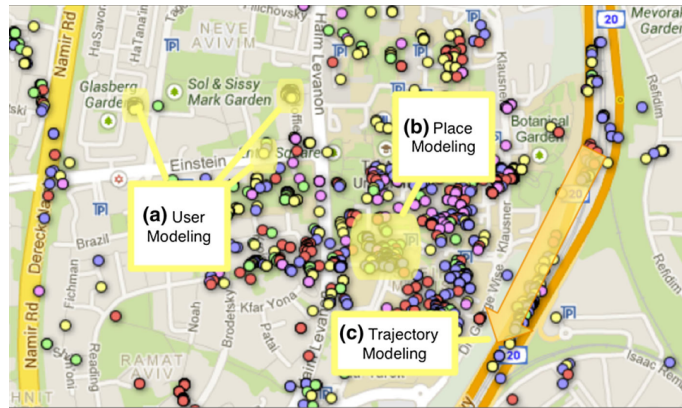


Figure 2.2.4: *Visualization of the three application classes: (a) user modeling where the object of analysis is a single user (represented by circles of the same color), (b) place modeling where the object is a geographic area (visited by different users; each is represented by a circle of its own color), (c) trajectory modeling where the object is a set of spatial–temporal points by the same user (represented by a yellow arrow). The map was generated using Google Maps*

together the observations corresponding to the same location. Correctly discovering such personal locations is an important problem with a wide range of applications. In the literature, this problem is typically addressed using a grid partitioning of the studied area or generic clustering algorithms like DBSCAN [90] or OPTICS [17]. However, this type of clustering methods have various drawbacks. First, some of them are focused on specific optimization criteria, such as maximizing compactness or density connectivity, which does not always correspond exactly to the notion of locations, and therefore, the results, though optimal with respect to its own criteria, are not good locations.

Second, in some cases, the algorithms need parameters that are not easy to guess (e.g., the size of the cell for the grid partitioning and the radius and minimum points for DBSCAN) and that should be tuned ad hoc for the data of each user analyzed. Indeed, in most cases an experienced analyst or some expensive self-tuning procedure might be needed to select accurately the parameters. On the other hand, in most cases such parameters are fixed for all users, while each individual might show specific features that require a treatment different from the others.

TOSCA (two-step clustering algorithm) [128] is an approach that tries to overcome these drawbacks, providing a parameter-free algorithm explicitly shaped for personal location detection. The two steps of TOSCA are realized by combining two clustering methods and a statistical analysis approach. TOSCA enables in this way to produce high-quality clusters with a low computational cost. The idea behind TOSCA comes from the need to detect the locations of the users in an efficient way without sacrificing the clustering quality and, most importantly, without any tuning phase for the parameters. Extensive experimentation showed that center-based clustering methods tend to incorrectly identify subgroups of observations that should belong to the same location. The side effect of such constraints is that the result

usually splits real locations into several pieces that are connected with each other in a relatively loose way. On the other hand, single-linkage and density-based clustering methods are very good at spotting such loose connections, with the drawback of not distinguishing well those loose connections that are actually boundaries with other clusters. By exploiting these observations, the two main steps of TOSCA work as follows (see Fig. 2.2.5): (1) extract (sub-)clusters and corresponding medoids through center-based methods. X-means [239] algorithm was selected through empirical evaluations; and (2) cluster the medoids through a single-linkage hierarchical algorithm [276]. Stop the iterative cluster aggregation (or, equivalently, cut the dendrogram resulting from a complete run of the algorithm) through a statistically determined threshold on the increase in the distance between the clusters to be merged at each iteration. The cut criteria considered in TOSCA come from the outlier detection theory [223]. The distribution of the difference of the distances in the dendrogram returned by single linkage experimentally shows a sudden spike indicating the change in trend in the aggregation of the clusters.

It has been shown how, in contrast to algorithms commonly used in the literature, TOSCA automatically detects a good distance threshold for the clusters produced, thus adapting the clustering to the individual mobility behavior of each user in the data [128]. Therefore, it is perfectly suitable as autofocus clustering algorithm for analyzing individual mobility data. TOSCA evaluation against a large set of competitors over data generated from a null model and a mobility-like model shown that both in the mobility-like model and in the real case study TOSCA performs better than the generalpurpose algorithms producing the desirable clustering for personal mobility data mining (see Fig. 2.2.6).

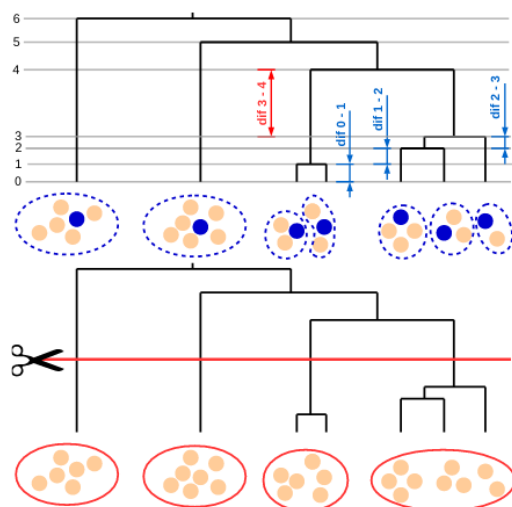


Figure 2.2.5: Orange points represent the stop observations input of TOSCA. Blue dotted circles correspond to the X-Means clusters and the blue points to their medoids, which are then processed by Single Linkage. On the resulting dendrogram we highlight the differences among the element distances. The red interval represents a possible cut. The image comes from [128].

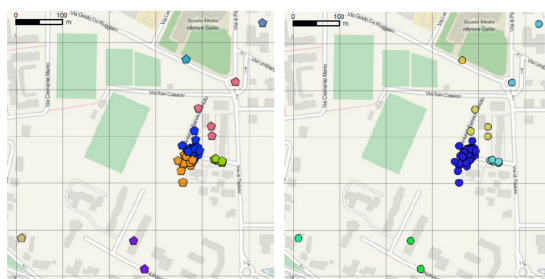


Figure 2.2.6: Personal locations detected with X-means (left) and TOSCA (right). Different colors denote the different clusters (personal locations) to which each point is assigned [16].

## 2.3 Transfer Learning

Transfer learning is a machine learning method where a learning model developed for a first learning task is reused as the starting point for a learning model in a second learning task[304]. It is a research problem that focuses on storing knowledge gained while solving one problem and applying it to a different but related task. It also focuses on recognizing which part of knowledge can be shared and which is strictly connected to the original problem. A key argument of this thesis will be the transfer learning technique applied to the human mobility topic. We will call it *Geographical Transfer Learning* and it will be explained in the next pages in details. Before that

an overview of what transfer learning is and what are the areas in which it is mostly applied will be presented.

### 2.3.1 Basic Definitions

Learning new knowledge and skills is one of the most important capabilities for human beings. Based on personal own studying experience and previous knowledge stored in brain, we are able to learn similar knowledge in a simplified way without studying it from the beginning. Our brain is able to draw on our previous knowledge and experience in order to minimize the time and energy needed to learn a new task. For instance, learning how to play tennis table can be easier for someone which already plays tennis, since both sports need a racket to play and may share some common knowledge and strategies. This is the study on how human beings learn new knowledge by individual way to transfer information preprocessed before to learn similar new information [8].

In traditional supervised machine learning, developers create a model that performs a certain task. For this task it is often necessary to have a large amount of labeled data available to be divided into train and test dataset. In this way it is possible to train the algorithm and optimize the process. This labeled data originates from a certain place, a so-called domain. Without the proper amount of input data, the model is most likely not able to modify its weights well enough to reach a certain accuracy. In particular, this problem exists when using a random initialization for weights and biases at the beginning of the training phase of a ML algorithm. The usual approach is to train the model with the labeled data from a certain domain and thus make it perform as good as possible for the respective domain. Although this might suffice in many cases, the difficulties arise quickly if we want to use the same network on a slightly different environment. For example, when a model is trained to perform on detecting people on pictures during the day then the algorithm will perform worse if applied to images with people at night time. The reason for this behavior comes from the training stage. Indeed the algorithm learns a certain level of bias and errors influencing its output. These biases have been adapted to perform in the best way possible on the training data it has seen. Therefore, due to the bias, which is optimized on pictures of people at daytime, the model is most likely not able to generalize. Hence, it won't be able to perform decently on pictures from a new domain [264]. This is where transfer learning comes into play. With the use of transfer learning, the problem depicted above can be solved without the need to train a new network on huge amounts of labeled data from the new domain. In transfer learning, a model A is trained on a dataset from a source domain for a source task. By doing so, the algorithm gains insight on how to solve the source task and tries to maximize its performance.

According to this theory, the prerequisite of transfer is that there needs to be a connection between two learning activities. Fig. 2.3.2 shows some intuitive examples about transfer learning.

In this section, we introduce some essential notations and definitions for the topic

understanding. First of all, we give the definitions of a “domain” and a “task”, respectively. Both notations and definitions match those from the survey paper by Pan [226].

**Definition 2.3.1** (Domain). A domain  $D$  is composed of two parts, i.e., a feature space  $\chi$  and a marginal distribution  $P(X)$ , where  $X = \{x_1, \dots, x_n\} \in \chi$

So, for example, if our learning task is document binary classification, then  $\chi$  is the space of all term vectors,  $x_i$  is the  $i_{th}$  term vector corresponding to some documents, and  $\chi$  is a particular learning sample. In general, if two domains are different, then they may have different feature spaces or different marginal probability distributions.

**Definition 2.3.2** (Task). A task  $T$  consists of a label space  $Y$  and a decision function  $f$ , i.e.,  $T = \{Y, f\}$ . The decision function  $f$  is an implicit one, which is expected to be learned from the sample data.

From a probabilistic viewpoint,  $f(x)$  can be written as  $P(y|x)$ . In our document classification example,  $Y$  is the set of all labels, which is *True*, *False* for a binary classification task, and  $y_i$  is “*True*” or “*False*”.

Basically we consider the case where there is one source domain  $D_S$ , and one target domain,  $D_T$ , as this is by far the most popular of the research works in the literature. More specifically, we denote the source domain data as  $D_S = \{(x_{S_1}, y_{S_1}), \dots, (x_{S_{n_S}}, y_{S_{n_S}})\}$ , where  $x_{S_i} \in X_S$  is the data instance and  $y_{S_i} \in Y_S$  is the corresponding class label. In the document classification example,  $D_S$  can be considered as a set of term vectors together with their associated true or false class labels. In an analogous way, we denote the target domain data as  $D_T = \{(x_{T_1}, y_{T_1}), \dots, (x_{T_{n_T}}, y_{T_{n_T}})\}$ , where the input  $x_{T_i}$  is in  $X_T$  and  $y_{T_i} \in Y_T$  is the corresponding output.

**Definition 2.3.3** (Transfer Learning). Given a source domain  $D_S$  and learning task  $T_S$ , a target domain  $D_T$  and learning task  $T_T$ , transfer learning aims to help improve the learning of the target predictive function  $f_T(\cdot)$  in  $D_T$  using the knowledge in  $D_S$  and  $T_S$ , where  $D_S \neq D_T$ , or  $T_S \neq T_T$ .

Then Transfer Learning is a technique where a model trained on one task is re-purposed on a second related task. Transfer learning and domain adaptation refer to the situation where what has been learned in one setting, is exploited to improve generalization in another setting. In the above definition, a domain is a pair  $D = \{X, P(X)\}$ . Thus the condition  $D_S \neq D_T$  implies that either  $X_S \neq X_T$  or  $P_S(X) \neq P_T(X)$ . This case respect to transfer learning is defined as heterogeneous transfer learning while the case where  $X_S = X_T$  with respect to transfer learning is defined as homogeneous transfer learning. If we consider our document classification example, this means that between a source document set and a target document set, either the term features are different between the two sets (e.g., they use different languages), or their marginal distributions are different. The case of traditional machine learning is  $D_S = D_T$  and  $T_S = T_T$ .



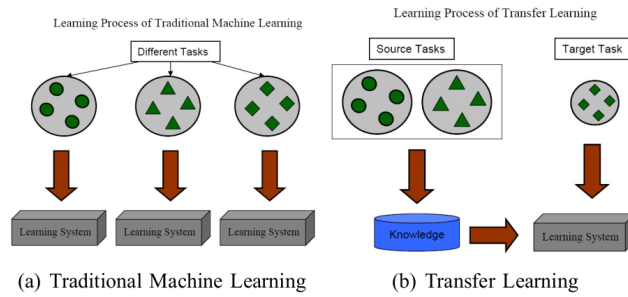


Figure 2.3.1: *Traditional Learning vs Transfer Learning*

During the process of transfer learning, the following three important questions must be answered:

- *What to transfer?*: This is the first and the most important step in the whole process. It is important to understand which part of the knowledge can be transferred from the source to the target in order to improve the performance of the target task. When trying to answer this question, we try to identify which portion of knowledge is source-specific and what is common between the source and the target.
- *When to transfer?*: There can be scenarios where transferring knowledge for the sake of it may make matters worse than improving anything (also known as negative transfer). The aim is utilizing transfer learning to improve target task performance/results and not degrade them.
- *How to transfer?*: Once the what and when have been answered, it is possible to proceed towards identifying ways of actually transferring the knowledge across domains/tasks. This involves changes to existing algorithms and different techniques.

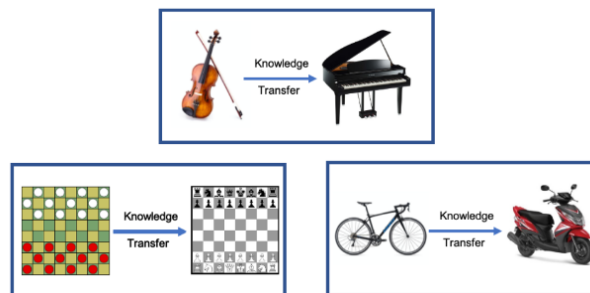


Figure 2.3.2: *Some intuitive examples of transfer learning*

### 2.3.2 Transfer Learning Scenarios

In literature it is possible to find transfer learning tasks related to problems such as multi-task learning and concept drift in the deep learning framework but also in many other applications or areas of study. It is identified with many representative approaches that include predictive modeling problems too. Two common approaches are the Develop Model Approach and the Pre-trained Model Approach.

The first one consists in selecting a source task related to a predictive modeling problem. The data there will be some relationship in the input data, output data, and/or concepts learned during the mapping from input to output data. Then the model is developed for this first task: it must be better than a naive model to ensure that some feature learning has been performed. In conclusion, the model fit on the source task can then be used as the starting point for a model on the second task of interest, this may involve using all or parts of the model, depending on the modeling technique used, or just a selection of them. Sometimes, the model may need to be adapted or refined on the input-output pair data available for the task of interest.

The second approach is more common in the field of deep learning and consists in select a pre-trained source model from all the available models (many research institutions release models on large and challenging datasets that may be included in the pool of candidate models from which to choose). Then the pre-trained model can be used as the starting point for a model on the second task of interest. This may involve using all or parts of the model, depending on the modeling technique used. And in conclusion tuning the model in order to adapt it or refine on the input-output pair data available for the task of interest.

Assuming somebody considers reusing a pre-trained network with a new dataset, it is very likely that the original and the new dataset are situated in one of the four following categories inspired by [96]. All four scenarios are depicted in Figure 2.3.3.

- **Scarcity of data & low similarity.** This approach is the least promising in terms of transfer learning. In a neural network the lower layers of the network serve as basic feature extractors. In image recognition, for example, in the first layers the network learns to recognize edges, corners and contours. Later layers will then learn more complex structures. At last, the output layers identify objects on the images. When having a dataset in this state, the only possibility to reuse the original network is by trying to freeze the first layers and to fine tune the more advanced ones. The theory behind this approach is to retain the capability of extracting the features that are similar in the original and the new datasets. Transferred to the image recognition example, to detect objects, every image classifier needs to learn how to recognize corners and edges first. The goal is to keep this skill for the new dataset. This option though, as mentioned above, is not known to promise very good results.
- **Scarcity of data & high similarity:** In this scenario, the new dataset is rather small and thus its size does not suffice to reach a decent accuracy by training a neural network from scratch. But since the new dataset and the

original input to the pre-trained network are highly similar, it is plausible that by adjusting the output layers of the original network the accuracy on the new dataset will be on a decent level. In this case, the lower layers of the original network serve as a feature extractor for the general problem statement that confronts both of the datasets.

- **Richness of data & low similarity:** If the new dataset contains a big amount of input-output pairs which are highly dissimilar compared to the original dataset, training a neural network from scratch would be the better approach. There are two main reasons for this. First, in this situation, the network can learn all the needed features from the big new dataset, thus there is no need to inherit these skills from a pre-trained algorithm. Second, as the datasets do not have a lot in common, spoiling the network with the original dataset would not help with converging to the correct solution for the new dataset, as the weights will adapt into the wrong direction.
- **Richness of data & high similarity:** Having a big dataset which is fairly similar to the original dataset is obviously the optimal scenario. This state indicates that there is no need to discard any of the previously acquired skills of the algorithm as the features between both datasets are similar. In order to ensure a high accuracy with the pre-trained model all the layers should not be changed. In this case, the only modification needed to apply the neural network to the new dataset is to keep the layers but to fine tune the pre-trained model of the algorithm in order to minimize the original influence of the dataset on the weights and biases. Fine tuning means to reload and retrain the pre-trained network with the new data.

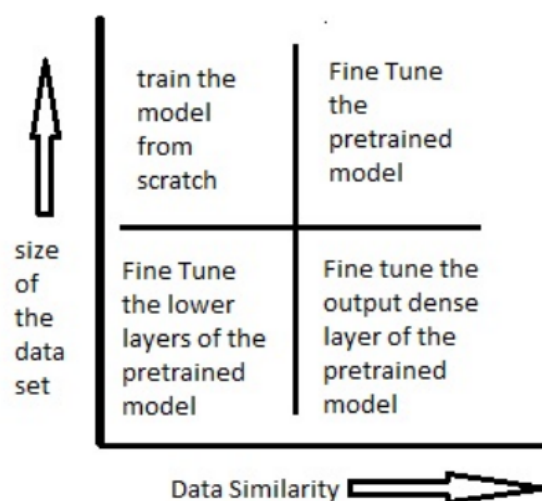


Figure 2.3.3: A schema of the four typical scenarios [96].

Another way to divide the transfer learning approaches is the one introduced by Pan. [226] that will be presented in the next pages. We can consider three different types of transfer learning methods: *Inductive transfer learning*, *Transductive transfer learning* and *Unsupervised transfer learning*.

### Transfer in Inductive Learning

In an inductive learning task, the objective is to induce a predictive model from a set of training examples [304]. Often the goal is classification, i.e. assigning class labels to examples. Examples of classification systems are artificial neural networks and symbolic rule-learners. Another type of inductive learning involves modeling probability distributions over interrelated variables, usually with graphical models. The predictive model learned by an inductive learning algorithm should make accurate predictions not just on the training examples, but also on future examples that come from the same distribution. In order to produce a model with this generalization capability, a learning algorithm must have an inductive bias [207], that is a set of assumptions about the true distribution of the training data.

The bias of an algorithm is often based on the hypothesis space of possible models that it considers. For example, the hypothesis space of the Naive Bayes model is limited by the assumption that example characteristics are conditionally independent given the class of an example. The bias of an algorithm can also be determined by its search process through the hypothesis space, which determines the order in which hypotheses are considered. For example, rule-learning algorithms typically construct rules one predicate at a time, which reflects the assumption that predicates contribute significantly to example coverage by themselves rather than in pairs or more.

Transfer in inductive learning works in this way: the target task is different from the source task, but some data in the target domain is labeled as well. The data in the source domain could be labeled or not. So the target domain data are required to induce the knowledge learned in the source task for the target task (see 2.3.4). The left hand side of Figure 2.3.4 depicts how the network converges to a solution with no prior knowledge about the problem statement. Clearly, the algorithm needs many backpropagation steps to tune the weights into the right direction. For the tuning of the weights to function properly, a big dataset is necessary in order to give the network a sufficient amount of references for the optimal solution. The way this is done varies depending on which inductive learning algorithm is used to learn the source and target tasks. Some transfer methods narrow the hypothesis space, limiting the possible models, or remove search steps from consideration. Other methods broaden the space, allowing the search to discover more complex models, or add new search steps.

One area of inductive transfer applies specifically to Bayesian learning methods. Bayesian learning involves modeling probability distributions and taking advantage of conditional independence among variables to simplify the model. An additional

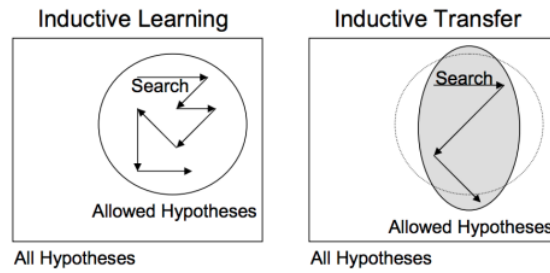


Figure 2.3.4: *Inductive learning can be viewed as a directed search through a specified hypothesis space [207]. Inductive transfer uses source-task knowledge to adjust the inductive bias, which could involve changing the hypothesis space or the search steps [304].*

aspect that Bayesian models often have is a prior distribution, which describes the assumptions one can make about a domain before seeing any training data. Given the data, a Bayesian model makes predictions by combining it with the prior distribution to produce a posterior distribution. A strong prior can significantly affect these results (see Figure 2.3.5). This serves as a natural way for Bayesian learning methods to incorporate prior knowledge – in the case of transfer learning, source-task knowledge. Marx et al. [203] use a Bayesian transfer method for tasks solved by a logistic regression classifier. The usual prior for this classifier is a Gaussian distribution with a mean and variance set through cross-validation. To perform transfer, they instead estimate the mean and variance by averaging over several source tasks. Raina et al. [251] use a similar approach for multi-class classification by learning a multivariate Gaussian prior from several source tasks.

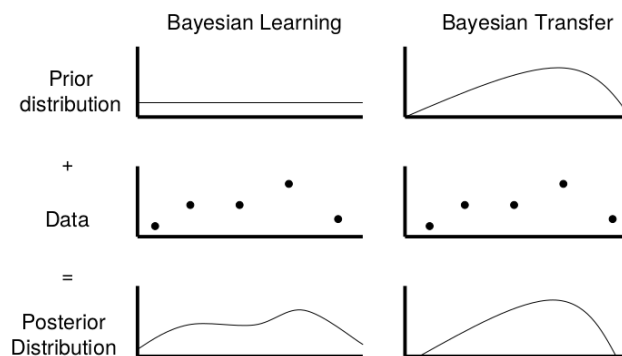


Figure 2.3.5: *Bayesian learning uses a prior distribution to smooth the estimates from training data. Bayesian transfer may provide a more informative prior from source-task knowledge. [304].*

Another setting for transfer in inductive learning is hierarchical transfer. In this setting, solutions to simple tasks are combined or provided as tools to produce a

solution to a more complex task (see Figure 2.3.6). This can involve many tasks of varying complexity, rather than just a single source and target. The target task might use entire source-task solutions as parts of its own, or it might use them in a more subtle way to improve learning. Sutton and McCallum [287] begin with a sequential approach where the prediction for each task is used as a feature when learning the next task. They then proceed to turn the problem into a multi-task learning problem by combining all the models and applying them jointly, which brings their method outside our definition of transfer learning, but the initial sequential approach is an example of hierarchical transfer.

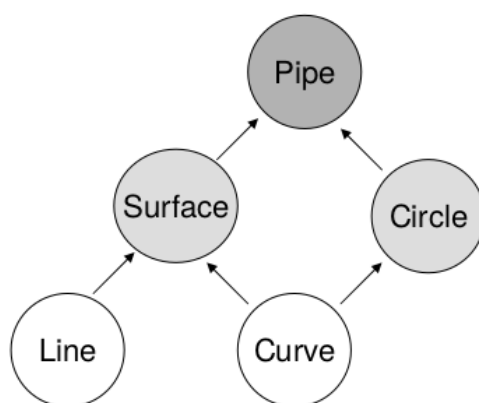


Figure 2.3.6: *An example of a concept hierarchy that could be used for hierarchical transfer, in which solutions from simple tasks are used to help learn a solution to a more complex task. Here the simple tasks involve recognizing lines and curves in images, and the more complex tasks involve recognizing surfaces, circles, and finally pipe shapes.*[304].

### Transductive transfer learning

Among machine learning paradigms, unsupervised transductive transfer learning is useful when no labeled data from the target domain are available at training time, but there is accessible unlabeled target data during training phase instead.

In contrast to inductive learning, transductive learning techniques have observed all the data beforehand, both the training and testing datasets. We learn from the already observed training dataset and then predict the labels of the testing dataset. Even though we do not know the labels of the testing datasets, we can make use of the patterns and additional information present in this data during the learning process. So, a Transductive Transfer Learning method is a method that, given a labelled training set and an unlabelled object set (and optionally two unlabelled training sets) from two different but related domains  $S$  and  $T$ , generates predicted labels for all documents without using any general rule.

Basically, the main differences of a transductive transfer algorithm with respect to an inductive one is that in the former, unlike in the latter, there is an object set which is observed at training time, and we generate no general hypothesis but only predicted labels. The main difference of a transductive transfer learning algorithm with respect to a transductive one is instead that in the second the training set and the object set are not the same, since they originate from two different domains  $S$  and  $T$ . In [254] authors propose a novel unsupervised transductive transfer learning method to find the specific and shared features across the source and the target domains. The proposed learning method maps both domains into the respective subspaces with minimum marginal and conditional distribution divergences. It discriminates the classes of both domains via maximizing the distance between each sample-pairs with different labels and via minimizing the distance between each instance-pairs of the same classes.

### Unsupervised transfer learning

Existing transfer learning methods mainly focus on how to fix the discrepancy between supervised tasks. However, unsupervised learning has achieved remarkable advances in recent years and has become an interesting topic in the deep learning community. In this case the question we want to answer is: Is it necessary to do transfer learning between unsupervised tasks, and how to do it?

As a typical unsupervised learning paradigm, predictive learning has shown great research significance in discovering the underlying structure of unlabeled spatiotemporal data without human supervision and learning generalizable deep representations. In the unsupervised case the target task is different from the source task and the labeled data are unknown both in the source and target domain. So this type of transfer learning methods focuses on tasks like clustering or dimensional reduction with the goal of sharing the gained knowledge in terms of ways of clustering or feature selections.

The literature contains a large body of contributions on inductive and transductive transfer learning, since they are the natural extensions of supervised learning problems: once a great deal of effort has been expended to develop a model on a supervised, clean, and controlled dataset, transfer learning aims at maximising the return on investment by making the model applicable in as many real-life applications as possible. Research on unsupervised transfer learning, however, has been more limited. Most works focus on the machine learning tasks of clustering and dimensionality reduction [226] [292].

Unsupervised Domain Adaptation (UDA) has been widely studied to transfer knowledge across different data distributions [144]. Critically, by assuming all the domains are sharing the same label space, UDA learns to either align feature distributions [109] or map the relationships between the decision boundaries and feature representations so that the boundaries are valid for both domains [174]. However, UDA is not always practical as it cannot enumerate all the categories for model training let alone exhaustively collecting and annotating the data. In [335], authors propose a novel differentiable framework named transferable memory that provides

diverse understandings of the underlying, complex data structure of the target domain. Technically, they perform unsupervised knowledge distillation on the memory states of multiple pretrained recurrent networks, and then introduce a new gating mechanism to dynamically find the transferable part of the distilled representations. The work is also inspired by the idea of knowledge distillation [141], which transfers knowledge from larger models into smaller, faster models without losing too much generalization ability.

### 2.3.3 Avoiding Negative Transfer

Given a target task, the effectiveness of any transfer method depends on the source task and how it is related to the target. If the relationship is strong and the transfer method can take advantage of it, the performance in the target task can significantly improve through transfer. However, if the source task is not sufficiently related or if the relationship is not well leveraged by the transfer method, the performance with many approaches may not only fail to improve but it may actually decrease. Negative transfer learning occurs when the information learned from a source domain has a negative effect on a target learner instead. More formally, given a source domain  $D_S$ , a source task  $T_S$ , a target domain  $D_T$ , a target task  $T_T$ , a predictive learner  $f_{T_1}(\cdot)$  trained only with  $D_T$ , and a predictive learner  $f_{T_2}(\cdot)$  trained with a transfer learning process combining  $D_T$  and  $D_S$ , negative transfer occurs when the performance of  $f_{T_1}(\cdot)$  is greater than the performance of  $f_{T_2}(\cdot)$ . The topic of negative transfer addresses the need to quantify the amount of relatedness between the source domain and the target domain and whether an attempt to transfer knowledge from the source domain should be made. Extending the definition above, positive transfer occurs when the performance of  $f_{T_2}(\cdot)$  is greater than the performance of  $f_{T_1}(\cdot)$ .

Ideally, a transfer method would produce positive transfer between appropriately related tasks while avoiding negative transfer when the tasks are not a good match. In practice, these goals are difficult to achieve simultaneously. Approaches that have safeguards to avoid negative transfer often produce a smaller effect from positive transfer due to their caution. Conversely, approaches that transfer aggressively and produce large positive-transfer effects often have no protection against negative transfer.

One way of approaching negative transfer is to attempt to recognize and reject harmful source-task knowledge while learning the target task. The goal in this approach is to minimize the impact of bad information, so that the transfer performance is at least no worse than learning the target task without transfer. At the extreme end, an agent might disregard the transferred knowledge completely, but some methods also allow it to selectively reject parts and keep other parts. Option-based transfer in reinforcement learning ([76]) is an example of an approach that naturally incorporates the ability to reject bad information.

Another way for avoiding negative transfer is if there exists not just one source task, but a set of candidate source tasks. In this case the problem becomes choosing the



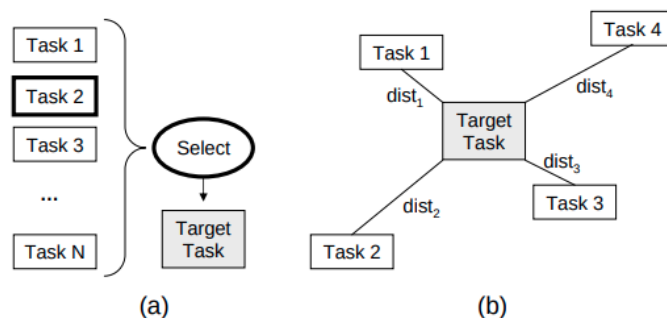


Figure 2.3.7: (a) One way to avoid negative transfer is to choose a good source task from which to transfer. In this example, Task 2 is selected as being the most related. (b) Another way to avoid negative transfer is to model the way source tasks are related to the target task and combine knowledge from them with those relationships in mind [304].

best source task (see Figure 2.3.7). An example of this approach is the transfer hierarchy [40], who order tasks by difficulty. Appropriate source tasks are usually less difficult than the target task, but not so much simpler that they contain little information. Given a task ordering, it may be possible to locate the position of the target task in the hierarchy and select a source task that is only moderately less difficult.

### 2.3.4 Applications

Nowadays transfer learning methods are applied in several sectors of machine learning and used a lot in various applications in specific areas such as medicine, transportations, image classification and text recognition. In this section a few examples are mentioned:

- Transfer learning for NLP: Textual data presents all sorts of challenges when it comes to ML and deep learning. These are usually transformed or vectorized using different techniques. Embeddings, such as Word2vec and FastText, have been prepared using different training datasets. These are utilized in different tasks, such as sentiment analysis and document classification, by transferring the knowledge from the source tasks. Besides this, newer models like the Universal Sentence Encoder and BERT definitely present a myriad of possibilities for the future.
- Transfer learning for Audio/Speech: Similar to domains like NLP and Computer Vision, deep learning has been successfully used for tasks based on audio data. For instance, Automatic Speech Recognition (ASR) models developed for English have been successfully used to improve speech recognition performance for other languages, such as German. Also, automated-speaker identification is another example where transfer learning has greatly helped.

- **Transfer learning for Computer Vision:** deep learning has been quite successfully utilized for various computer vision tasks, such as object recognition and identification, using different CNN architectures. In [336] authors present their findings on how the lower layers act as conventional computer-vision feature extractors, such as edge detectors, while the final layers work toward task-specific features.
- **Geographical Transfer Learning:** the various types of mobility models involved in this thesis are expected to be highly dependent on the specific geographical area under study. For instance, it has been empirically verified that the trip purpose prediction models work very well in the areas where they were extracted, their performances degrade dramatically if applied to areas with different characteristics (see [257]). At the same time, not all areas of interest are equally well covered by data, due to the non-homogeneous penetration of tracking devices, making it difficult to build different models for different areas. For those reason finding a way to share models and knowledge between different geographical area is essential. In the next chapters this problem will be explained in detailed and some solutions will be presented and motivated.

## 2.4 Ethical Aspects of Data Science

GPS devices, smartphones, and social media, which are used to register people's locations, time, and social activities include also the users' private information.

The majority of the models and algorithms for predicting individual mobility require the sharing of the historical travel information of users. In addition, some blogs on social media possess users' photos, events, and social relationship. For those reasons the privacy problem for people providing personal information becomes crucial [318]. The trajectory data contains daily travel routes. Attackers may use the spatial and temporal correlations hidden in a user's trajectory data to deduce their mobility patterns and identify their home and workplaces. Several techniques have been proposed to protect users' trajectory data from privacy leak. For example existing privacy-preserving technologies include clustering-based [4], generalization-based [215] and suppression-based methods [145].

We already saw that the explosive growth in the quantity and quality of personal data has created a significant opportunity to generate new forms of economic and social value. For that reason, there is a need for the same kinds of rules and frameworks that exist for other asset classes. Citizens are more worried every day about what companies and institutions do with their data, and ask for clear positions and policies from both the governments and the data owners. Despite this increasing need, there is no unified view on privacy laws across countries. Until 2000s, The European Union regulated privacy by Directive 95/46/EC (Oct. 24, 1995) and Regulation (EC) No 45/2001 (December 18, 2000). The European regulations were based on the notion of "non-identifiability". The regulation on privacy in the EU was then revised by the comprehensive reform of the data protection rules proposed on January 25, 2012 by the European Commission, which will be applied on May 25, 2018 in the form of Regulation, i.e., the General Data Protection Regulation (GDPR) [246].

While legal frameworks evolve, ethical concerns and guidelines are changing as well. Authors in [99] consider the impact of social media on society and the ethical attention is reflected by social networks continuing to update privacy policies and settings, by newsrooms making publishing guidelines on how they use material sourced from social media platforms, and by the continuous shifts in what is or is not considered appropriate when individuals post on social media platforms.

Moreover, both active and passive data collections also raise questions. Even though users are publishing messages and personal information on public networks, many users do not consider that anyone other than their close friends and family will see their posts. But, while many users are aware of the information they have logged into social networks, they are much less aware of the data being collected from them. World Economic Forum warns both people and organizations. It points out that people have the right to be informed about the potential impact of their content being shared widely. This concept can easily be extended to other types of data, not just the social networks ones. For that reason, a part of this thesis concerns privacy risk in human mobility. In one of the next chapters, the possible hidden risks of the

misuse of mobility data will be explained in detail.

However, World Economic Forum is not the only entity that invokes transparency. Indeed, transparency is one of the most essential part in ethics and it is related to several aspects of the big data process, such as seeking permission of users, explanation of terms of use, and data usage after the collection. In [98], the OECD framework is presented. It is a unique forum where governments work together to address the economic, social and environmental challenges of globalisation along with fundamental principles that should be respected in data usage process: collection limitation (data collected are the minimum necessary and they must be obtained by lawful and fair means), data quality (personal data should be relevant to the purposes for which they are to be used and they must be complete and up-to-date), purpose specification (purposes should be specified before of data collection), use limitation (data must be used and disclosed only for the specified purpose), security safeguard (data must be protected by reasonable security safeguards), openness (about development, practices and policies), individual participation (individuals should have the right to control, rectify or have the data erased) and accountability (data controllers should be accountable for complying with measures regarding the other principles). In [222], there is a short list together with some practical examples of good and bad practices, of the six key principles they consider essential in the data management: (i) to highlight the users need and public benefit from the start of the definition of the methods; (ii) to use data and tools with the minimum intrusion necessary; (iii) to create robust data science models, studying the peculiarities of the data and the presence of potential discrimination features; (iv) to be aware of public perception, understanding how people expect their data to be used; (v) to be clear and open about data, tools and algorithms, providing explanation in plain English; and (vi) to keep data secure, following the guidelines provided by the Information Commissioner's Office 15 . Finally, the Council of Europe [221] drafts some guidelines too. The majority of ethical principles are highly shared among different institutions, and many of them had been included in the new EU Regulation before and in the GDPR now.

### 2.4.1 General Data Protection Regulation (GDPR)

The General Data Protection Regulation (EU) 2016/679 (GDPR) is a regulation in EU law on data protection and privacy in the European Union (EU) and the European Economic Area (EEA). It also addresses the transfer of personal data outside the EU and EEA areas. The GDPR's primary objective is to enhance individuals' control and rights over their personal data and to simplify the regulatory environment for international business. Superseding the Data Protection Directive 95/46/EC, the regulation contains provisions and requirements related to the processing of personal data of individuals (formally called data subjects in the GDPR) who are located in the EEA, and applies to any enterprise—regardless of its location and the data subjects' citizenship or residence—that is processing the personal information of individuals inside the EEA.

The GDPR was adopted on 14 April 2016 and became enforceable beginning 25 May 2018. As the GDPR is a regulation, not a directive, it is directly binding and applicable but does provide flexibility for certain aspects of the regulation to be adjusted by individual member states [325]. This last aspect heralds the hidden criticisms since it does not include a plan of fixed rules. Those limitations will be discussed more in Sec. 4.4.4 in the mobility data context

GDPR is composed of 11 chapters that go from principles and right to accessing to data, from transferring of personal data and competences, to implementing Acts. In particular, in Article 5 GDPR state some of the ethical principles considered fundamental, such as data minimization, transparent information, purpose and limitation and accountability. GDPR also introduces some novelty with respect to the Directive 95/46/EC, such as explicit references to Data protection by design and by default, to data protection impact assessment and new obligations of data processors. A Data Controller is a *"natural or legal person, public authority, agency or other body which, alone or jointly with others, determines the purposes and means of the processing of personal data"* (Article 4 (7)). A Data Processor is a *"natural or legal person, public authority, agency or other body which processes personal data on behalf of the controller"* (Article 4 (8)).

Data processor has, among its obligations:

- guarantee to implement appropriate technical and organizational measures in such a manner that processing ensure the protection of the rights of the data subject, where processing is to be carried out on behalf of a controller;
- inform the controller of any intended changes concerning the addition or replacement of other processors;
- processes the personal data only on documented instructions from the controller (including the categories of processing carried out and any transfer to a third country);
- takes all the data protection measures required also for the Data Controller.

The data protection measures should be applied to any information concerning an identified or identifiable natural person. Identified data (e.g., name and social security number) are directly linked to the individual, whereas identifiable data (e.g., nickname or address) are attributable to a specific person through some additional information. This data protection measures could consist, among other things, of minimizing the processing of personal data, pseudonymizing personal data as soon as possible, transparency with regard to the functions and processing of personal data, enabling the data subject to monitor the data processing, enabling the controller to create and improve security features. This processes are fundamental in the mobility framework too. For the purposes of this thesis it is essential to investigate the hidden risks behind an uncontrolled use of mobility data. As it is explained in Sec.4.4.4 it is very easy to exploit simple GPS data to reveal personal and sensitive information about users.

The GDPR regulation is a strong tool to protect the privacy rights of citizens but it still shows some lacks and interpretations errors that could be improved.

About this, it is important to cite the recent Breyer Case versus the Court of Justice of the European Union [60]. This case primarily concerns the question whether a website visitor's dynamic IP address constitutes personal data for a website publisher, when another party (an internet access provider) can tie a name to that IP address. In particular, the Court finds that an IP address constitutes personal data for the website publisher, if that publisher has the legal means to obtain, from the visitor's internet access provider, additional information that enables the publisher to identify that visitor [46]. This is the first ruling by the EU Court on the protection of personal data against a content provider. The functional approach of the Court of Justice is decisive in outlining, through interpretation, a community notion of "personal data" that binds all member states. At the same time, the Court's ruling offers an interpretation of Article 7, letter f) [60], of the same directive, stating that it constitutes the legitimate interest of an online media service provider to protect itself from possible cyber attacks, and reiterates the non-absolute nature of the right to protection of personal data. What was specified by the EU Court also appears of particular importance in view of the entry into force (in May 2018) of the new community regulations on the protection of personal data (Regulation n. 2016/679). After this case, law seemed to be clear about the protection of individuals with regard to the processing of personal data and on the free movement of such data. In essence data must be interpreted as meaning that a dynamic IP address registered by an online media services provider when a person accesses a website that the provider makes accessible to the public constitutes personal data within the meaning of that provision, in relation to that provider, where the latter has the legal means which enable it to identify the data subject with additional data which the internet service provider has about that person.

But apparently the law needs changes and updates continuously. Recently The European Data Protection Supervisor (EDPS) has lodged an appeal against the judgment of the General Court delivered on April 26, 2023, in the case of Single Resolution Board (SRB) vs European Data Protection Supervisor (EDPS) (Case T-557/20), seeking the annulment of the entire judgment under appeal [61]. The background of the case involves a dispute between the SRB and the EDPS regarding data protection regulations and the processing of personal data during the SRB's decision-making process concerning Banco Popular Español, S.A. Several complaints were submitted to the EDPS regarding alleged violations of Regulation 2018/1725 by the SRB's handling of personal data. The General Court discussed the treatment of personal data by the SRB during a resolution scheme decision and the subsequent handling of comments by individuals participating in the decision-making process [189]. The General Court found merit in the first plea raised by the SRB, and it annulled the EDPS's revised decision without delving into the second plea. The appeal by the EDPS seeks to challenge the General Court's interpretation of key data protection regulations and principles in light of the specific circumstances of the case [189]. The outcome of the appeal will have implications for data protection

practices within EU institutions. The EDPS claims that the General Court misinterpreted Article 4(2) and 26(1) of Regulation 2018/1725 by failing to consider the principle of accountability.

In conclusion the judgment above seems to confirm even more, compared to what has already been said by the Authorities and Courts, the adoption of an approach related to the interpretation of the concept of "identifiability". In this way, many data that were previously considered pseudonymized (and therefore personal) could fall into the category of anonymized data, simplifying the processing operations of many data controllers (also the Breyer case should be re-examined in the light of the new decisions). It is important, however, to point out, that the Court of First Instance did not claim that, in the case at hand, the data received by the third company were non-personal, but merely provided guidance on how to assess identifiability, which must use a relative approach from the perspective of the person receiving the pseudonymised/anonymised data. This does not exclude that, given the concrete situation, the recipient has the means to identify the persons concerned. A case-by-case assessment will therefore be necessary once again.

### 2.4.2 Privacy Aspects of Human Mobility Analysis

Nowadays, our daily life is centered on data. Whether or not we are aware of it, our simple everyday interactions with through digital devices produce a myriad of data, that is combined to create Big Data. We leave traces relating to our movements via our mobile phones and GPS devices, to our relationships within social networks, to our habits and tastes from query logs and records of what we buy.

These digital breadcrumbs are a treasure trove as a way to discover new patterns in human activities and a way to understand better many aspects of human behavior that it was impossible to study or analyze just a few years ago. The resulting data can also enable a totally new class of services that can improve directly and sensibly our society or provide ways to tackle and solve problems from new perspectives. The other side of the coin is the question of privacy: since the data describe our life at a very detailed level, privacy breaches can occur along with inferences that reveal the most personal details.

Most of these data are of sequential nature, such as time-stamped transactions, users' medical histories and trajectories. They describe sequences of events or users' actions where the timestamps make the temporal sequentiality of the events powerful sources of information. Unfortunately, such information often might unveil sensitive information that require protection under the legal frameworks for personal data protection. Thus, when such data has to be released to any third party for analysis, privacy-preserving mechanisms are utilized to de-link individual records from their associated users [27]. Privacy-preserving methods aim at preserving statistical properties of the data while removing the details that can help the re-identification of users. The challenge to researchers around the world is to share data without revealing private information of the users, and for that they need to protect the information using data anonymization techniques [103]. Several approaches provide

a worst-case probabilistic risk of user re-identification as a measure for how safe the anonymised data is [208]. However, these solutions may work to make registered users anonymous, but they are insufficient for data combined attacks. After all, with reference to tracking apps for fighting covid-19, the EDPB clarified 'that location data thought to be anonymised may in fact not be. Mobility traces of individuals are inherently highly correlated and unique. Therefore, they can be vulnerable to re-identification attempts under certain circumstances' [248].

## 2.5 Explainability

Machine Learning (ML) is at the heart of many recent technological and scientific developments. Its ability to find patterns in large volumes of data is revolutionizing several sectors; financial services, health-care, retail and many more [49]. For many years, the main research focus in ML was to maximize the predictive performances of the algorithms. This has led to the construction of increasingly sophisticated models able to capture the extreme nonlinearity of the relationships between the variables and the highly interactive processes that generated the data. The increasing complexity of such empirical models made it almost impossible to understand their internal reasoning, even to their developers [228]. The typical example of this trend is represented by the Deep Learning techniques [119], a subset of Machine Learning techniques which hierarchically combines a set of basic units in several layers. Deep Learning has become popular thanks to its astounding predictive performances in several tasks such as computer vision, speech recognition, and natural language processing. However, there is no intrinsic way to understand why the algorithm has come to its conclusions.

### 2.5.1 Lack of Transparency

Machine Learning models which have opaque internal reasoning are called "black boxes" and underlie many decision support systems (DSS). In the last decade, these ML-based DSS gained popularity also in supporting decisions that have safety-critical policy consequences. However, their lack of interpretability raised concerns among the scientific community and in society at large. Indeed, it was proven that such black boxes are easily misled by biases in the data. For example, in [255] it is shown that the amount of snow in a picture was how a machine learning algorithm learned to distinguish pictures of husky dogs from pictures of wolves, this because all the pictures labeled as husky dog contained snow in the background. Even if this might sound like a silly problem, it is easy to imagine how similar biases in the process of data collection could influence the decisions taken in a judicial or military context.

Although a machine learning model provides predictions with high precision and profound accuracy, what most matters to a decision-maker is how it came to that decision and therefore how it came about the prediction was determined or how a particular instance is classified by the algorithm. Doshi-Velez and Kim [85] answer



this question by stating that: *"the problem is that a single metric, like accuracy, it is an incomplete description of most of the real world tasks"*. Always the same authors explain that there are cases and applications where it is not necessary to provide an explanation in terms of interpretability.

Furthermore, it was shown that such black boxes are really sensitive to adversarial examples [33], i.e., inputs maliciously modified to mislead the ML algorithm into the wrong output while appearing unchanged to the human eye. In [229] the authors developed a technique able to modify the image of a stop sign in a way that's too subtle for the human eye, but that is also sufficient to make a Deep Neural Network (DNN) wrongly classify it as a yield sign. This kind of attacks could cause a self-driving car to have dangerous behaviors.

Moreover, some ethical concerns were raised when it was shown how ML algorithms could learn human prejudices from the data [338] and perpetuate them in their decision-making process. One example of unfairness in ML-based DSS is the COMPAS recidivism algorithm developed by Northpointe. COMPAS was developed to predict if a defendant should receive a release on bail. Analyzing the false positive rates, the investigative journalism organization ProPublica, has shown that according to the profiling scores provided by this software a black who did not re-offend was classified as high risk twice as much as whites who did not re-offend, and white repeat offenders were classified as low risk twice as much as black repeat offenders. [152]. This is just an example that nevertheless highlights the criticality behind the use of ML models. The bias issue needs to be considered a priority, especially when working with personal data.

The COMPAS case demonstrates Artificial Intelligence limits: until there isn't a model able to completely eliminate human bias will always be necessary to rely on human judgment and not only on algorithms. The consequences of the deployment of black-box algorithms can be especially harmful when human lives are at direct stakes, as in the health-care system. Thanks to the ability of ML algorithms to leverage large volumes of health-related data, ML-based DSS have the potential to help physicians in their diagnosis, predict the spread of diseases and identify groups of high-risk patients [49]. Even though state-of-the-art predictive performances are achieved by black-box algorithms, more straightforward and less accurate models have been historically preferred for predictive tasks in this field. The reason behind this is well exemplified in [55] where the authors employ a high-performance intelligible model to predict the risk of death for pneumonia patients. Thanks to the intelligibility of their model, they discovered that to the asthma patients were wrongly attributed a lower risk of dying from pneumonia complications. This reflected a real correlation in the data; in fact, patients with both asthma and pneumonia were immediately hospitalized, and they were given a more aggressive treatment which lowered their death risk compared to the general population. However, if asthma patients are not hospitalized, their death risk from pneumonia complications is higher compared to the general population. This is a typical example where a critical variable is left out (the aggressivity of the treatment), and therefore the outcome of the ML model is wrong.

In the same way, explainability is crucial for gaining a deeper understanding of mobility too. Recognizing mobility patterns and highlighting biases in the model's reasoning is essential also in the urban framework. It is important to develop mobility-related explanations that provide examples and counter-examples to validate trajectories and crowd flows from different perspectives. While models rely on many features, either external ones (e.g., weather data, POIs) or spatio-temporal ones, it is not clear what the role of each feature is to the model's prediction or generation [193]. Designing explainable models for human mobility is essential to gain valuable knowledge for possible users, such as policymakers and urban planners.

### 2.5.2 Need of interpretability

All these examples underline where the need for an interpretable ML comes from; a ML algorithm might have high predictive performances without being suitable for the real purpose it was built for, in other words, there is a mismatch between the metric used to evaluate the algorithm and its real objective [185]. Seen from this angle, explaining a model behavior can be considered as a further step in evaluating its performances, taking into account not only its predictive performances but also how well it helps in its real-world task. As pointed out in [267] the majority of questions asked to ML-based DSS are intrinsically causal, especially when the final purpose is to attend the real world or to take some policy decisions. The problem is that ML algorithms are not equipped to reason in causal terms, they only know how to reason in statistical terms; hence they do not have the tools necessary to reason on real-world interventions [236].

The increasing demand for interpretable algorithms created a new community of researchers whose aim is to build techniques able to explain the behavior of black box algorithms [125] in order to make them applicable and reliable in critical domains. The growing interest in these kinds of techniques outside of academia is exemplified by the US "Defense Advanced Research Projects Agency" (DARPA) initiative "eXplainable Artificial Intelligence (XAI)" [130]. Furthermore, the European Union introduced several measures to tackle some legal and ethical concerns about personal data privacy and data processing that will also impact ML practices. These measures were introduced in the GDPR to face the highly debated presence of a "right to an explanation" of the data subject when his or her data are used in the process of "solely" automated decision-making that envisages "significant effects" on him or her. As argued by [197] there is the need to combine transparency and comprehensibility of ML algorithms to make them intelligible for the data subject. Indeed the extensive use of ML-based DSS and weakly regulated use of personal data create a dangerous imbalance in terms of knowledge, and thus power, between the data processor, i.e., the person or organization which processes the personal data, and the data subject, i.e., the person whose personal data have been collected [263].

### 2.5.3 Interpretable Models

Saying that a model is interpretable means that it has the ability to explain or to present in understandable terms to a human [85]. In the literature, there is a small number of models that are recognized as inherently interpretable: linear models, decision trees and IF-THEN rules.

- **Linear Models:**  $y_1 = \beta_0 + \beta_1 x_{i1} + \dots + \beta_N x_{iN}$  are considered interpretable because of their functional form; once the parameters  $\beta$  which link each feature  $x_{ij}$  to the output  $y_i$  are learned from the data, it is possible to assess straightforwardly how much each feature change the output, i.e., a unit increase of the value in feature  $x_{ij}$  will increase the value of the output of a factor  $\beta_j$ , given that all the other features stay the same.
- **Decision Tree:** are considered interpretable because of their graphical representation which allows the user to have a full picture of the model behavior at once. Following the path from the root to leaf, it is straightforward to understand how the algorithm classifies each input. Furthermore, the hierarchical structure of the tree allows the user to have a sense of the relative importance given to each feature.
- **IF-THEN Rule:** are considered interpretable because they have a textual representation which is similar to human language. They are related to decision trees since a path from root to leaf is an IF-THEN rule

Each of this model is considered inherently interpretable because it is straightforward to see how its internals work. However, the fact that we can inspect their internal decision-making process does not guarantee that they meet all the requirements for interpretable models present in the literature. Indeed, the notion of interpretability of ML algorithms has still no formal technical definition, it can vary from paper to paper and it is often vaguely defined. It is possible to find conditions under which the previous models cannot be considered interpretable. For example, if we consider a linear model with hundreds of thousands of parameters, it is difficult to claim that it is interpretable, given that a human being will never be able to contemplate the entirety of the model at once. More generally, a model could be interpretable in principle but not in practice, e.g., its size (the number of parameters involved, the number of nodes in the tree) could make impossible for a human to follow the process from input to output in a reasonable time.

The reason why these inherently interpretable models are not widely used in real-world applications is that there is a tradeoff between interpretability and predictive performances. To reach high predictive performances using simple models such as those considered inherently interpretable is often necessary to heavily engineer the features to the point the interpretability is lost.

It appears clear that the problem of interpretability in machine learning is not an easy one. There are several dimensions that must be taken into account when

studying the ways a machine learning model can be considered interpretable. In [85] the authors try to lay the foundations for a more rigorous framework to evaluate interpretability. They propose a taxonomy which helps to frame the interpretability problem into several categories; application-grounded, human-grounded and functionality-grounded. For each category, they propose a methodology to evaluate interpretability and advise some possible benchmarks. They also try to delineate which explanatory-needs different tasks might have in common by listing some interpretability dimensions, e.g., different tasks might have in common some time constraints; a decision must be taken in a short amount of time; consequently there is the need for a concise explanation. Their work suggests that there is not a unique way to define interpretable models because it depends on the context. The authors also offer a high-level perspective on the dimensions that characterize interpretability; they differentiate the models from a technical point of view contrasting those who have a global level of interpretability and those who have a local one, and then they emphasize some properties that must be taken into account when providing an explanation to a user:

- **Global and local interpretation:** a model is considered globally interpretable when the user can follow and understand each step of the process that goes from input to output, whereas a model is considered locally interpretable if it allows the user to understand the reasons behind one single prediction, one at a time.
- **Time limitation:** the interpretability of a model might be tightly related to the time a user is willing to spend in understanding its behavior.
- **Nature of user expertise:** since different users might have different background knowledge and expertise, what can be considered interpretable by them might differ. For example, a user with a mathematical background might consider more interpretable a model comprised of sophisticated formulas than a logic-based model such as a decision tree.

In [185] the author identifies two main categories of solutions which confer interpretability: the first one is transparency, which is the ability to inspect the internals of the models in order to elucidate how it works. Linear models with few parameters are considered transparent models. The second is the post-hoc interpretability or explainability, which is the ability to provide explanations for the model behavior without the understanding of how it works. This is the same kind of interpretability that a human being has, she can make a decision and then explain herself without the need of understanding how her brain works. The author also warns about the possible pitfalls of both approaches to interpretability; model transparency might prevent the use of more powerful models and post-hoc interpretability might mislead the user because, in principle, a model might be trained in order to provide wrong but plausible explanations. In our framework, we deal in the explainability context in Sec. 4.3 where we use the Shapley Values (a special techniques described

in the next paragraphs) to understand the post-hoc interpretability of our results providing explanations for the model behavior.

### 2.5.4 Interpretability techniques

In [125] the authors identify two main categories of methods to open the black boxes.

- *transparent box design or explanations by design*, the techniques which fall in this category are the ones who try to build high-performance algorithms together with their explanations relying on the transparency notion of interpretability.
- *reverse engineer problem or black box explanation*, the techniques which fall in this category are the ones who try to construct explanations for the black-box behavior relying on the post-hoc notion of interpretability.

The authors further refine the reverse engineer problem in three sub-categories of methods which address different problems:

- The black box model explanation problem, which is solved by building an interpretable model able to mimic the global behavior of the black box.
- The black box outcome explanation problem, which is solved by building interpretable models which explain the model behavior for particular instances.
- The black box inspection problem, which is solved by providing textual or visual representations explaining why the black box make some predictions more often than others.

The method we focused on in Sec 4.3 is the one which solve the black box outcome explanation problem.

The intuition behind these kinds of methods is that even if the decision boundary learned by the black box in the feature space can be arbitrarily complex, locally it can always be faithfully approximated by a simpler, more interpretable model.

In [255] the authors present LIME (Local Interpretable Model-Agnostic Explanations), a technique to explain the prediction of any classifier. LIME works perturbing a human interpretable representation of the selected instance, querying the black box on the perturbed sample, assigning a weight to each perturbation according to its distance from the input, and feeding the resulting weighted input-output training set to an interpretable model. In this case, the interpretable model is a sparse linear classifier whose weights are interpreted as features importance. The two key points of LIME technique are the initial transformation of the input representation into a human-interpretable one and the generation of the instance neighborhood by random perturbations of the input. During the initial transformation, the original representation of the input in the feature space  $\vec{v} \in \mathbb{R}^d$  is transformed into a binary vector  $\vec{v} \in \{0, 1\}^d$  which represents the presence or absence of interpretable

components. For example, if the original input is an image, the interpretable representation will be a binary vector containing the presence or absence of contiguous patches of similar pixels. Then this interpretable representation of the input is perturbed to generate the feature space neighborhood. In LIME this perturbation is entirely random.

In [124], the authors present LORE (Local Rule-Based Explanations). LORE works generating a balanced neighborhood of the selected instance through a genetic algorithm, querying the black box on such sample and then feeding the resulting input-output training set to a decision tree. A local explanation given by LORE is a pair composed by a logic rule extracted from the decision tree and a set of counterfactual rules explaining which features should be changed in order to invert the black box decision. The two key points of LORE technique are the logic explanation augmented by counterfactual rules and the genetic generation of the neighborhood. The use of a logic rule together with counterfactual rules provide a high expressiveness of the explanation making it more human-comprehensible. Furthermore, the genetic approach to neighborhood generation yields to a neighborhood which is denser in the boundary regions of the predictor, i.e., a more balanced neighborhood in terms of classification decisions compared to a randomly generated one.

In [256], the authors present Anchors, an extension of LIME which gives explanations in the if-then decision rule form. The proposed method produces a set of decision rules called anchors which are sufficient conditions for the black box prediction. This approach helps the final user to understand the coverage of each explanation, i.e., the region where the explanation applies. Indeed, in regions of the feature space where the anchor holds the black box decision is almost always the same.

Among the important works to refer to we mention the **Shapley Values** [194], a strong method in which an additive method assesses the importance of variables through the expected conditional value of the original model. The Shapley Values come from game theory: imagine we have a predictive model, then the "game" is reproducing the outcome of the model, while the "players" are the features included in the model. What Shapley does is quantifying the contribution that each player brings to the game. This means that Shapley Values quantify the contribution that each features brings to the prediction made by the model. Shapley values are based on the idea that the outcome of each possible combination (or coalition) of features should be considered to determine the importance of a single one [315].

# Act I: Human Mobility as a Complex Network

## 3.1 What is a Complex Network?

Network science is a scientific discipline that studies the interconnections among diverse physical, engineered, information, biological, cognitive, semantic and social networks. A network is defined as a set of nodes connected by links. The study of this subject was born in the eighteenth century when the legendary mathematician Leonhard Euler solved the famous “Seven Bridges of Konigsberg ” problem [94]. In his short paper of 1736, he inadvertently started the immense branch of graph theory, the basis for our thinking about networks. The city of Konigsberg in Prussia (now Kaliningrad, Russia) was set on both sides of the Pregel River, and included two large islands, Kneiphof and Lomse, which were linked to each other, or to the two mainland portions of the city, by seven bridges (as illustrated in the below figure to the left). The problem was to draw a walk through the city that would cross each of those bridges once and only once. Euler, recognizing that the important constraints were the four bodies of land and the seven bridges, drew out the first known visual representation of a modern graph.



Figure 3.1.1: *The “Seven Bridges of Konigsberg ” problem representation*

A modern graph, as seen in Fig 3.1.1, is represented by a set of points, known as vertices or nodes, connected by a set of lines known as edges. This abstraction from a concrete problem concerning a city and bridges etc. to a graph makes the

problem tractable mathematically, as this abstract representation includes only the information important for solving it. At the end, Euler actually proved that this specific problem has no solution since the difficulties he faced was about the development of a suitable technique of analysis. Moreover he did not have the possibility to subsequently test what established this assertion with mathematical rigor. From there, the branch of math known as graph theory lay dormant for decades. In modern times, however, its applications are finally exploding.

The field of graph theory continued developing to provide answers to many arrangement, networking, optimization, matching and operational problems. Graphs can be used to model many types of relations and processes in physical, biological, social and information systems, and has a wide range of useful applications such as:

- Finding communities in networks, such as social media.
- GPS/Google maps to find the shortest path home.
- DNA sequencing

In this section we provide an introduction about complex network in general and a brief resume of the main discoveries and researches in network science. Complex weblike structures describe a wide variety of systems of high technological and intellectual importance. For example, a cell is best described as a complex network of chemicals connected by chemical reactions; the Internet is a complex network of routers and computers linked by various physical or wireless links; trends and ideas spread on the social network whose nodes are human beings and edges represent various social relationships. These systems represent just a few of the many examples that have recently motivated the scientific community to investigate the mechanisms that determine the type of complex networks. The desire to understand such systems has brought along significant challenges as well. Physics, a major beneficiary of reductionism, has developed several tools to predict the behavior of a system as a whole from the properties of its constituents. We now understand how magnetism emerges from the collective behavior of millions of spins, or how do quantum particles lead to such spectacular phenomena as Bose-Einstein condensation or superfluidity. The success of these modeling efforts is based on the simplicity of the interactions between the elements: there is no ambiguity as to what interacts with what, and the interaction strength is uniquely determined by the physical distance. While for many complex systems with nontrivial network topology such ambiguity is naturally present, in the past few years we increasingly recognize that the tools of statistical mechanics offer an ideal framework to describe these convoluted systems as well. These developments have brought along new challenges for statistical physics and many links to several other topics.

In mathematical terms a network is represented by a graph:



**Definition 3.1.1.** A graph  $G$  consists of a collection  $V$  of vertices and a collection of edges  $E$ ,  $G = (V, E)$ . Each edge  $e \in E$  is said to join two vertices, which are called its end points. If  $e$  joins  $u, v \in V$ , we write  $e = \langle u, v \rangle$ . Vertex  $u$  and  $v$  in this case are said to be adjacent. Edge  $e$  is said to be incident with vertices  $u$  and  $v$ , respectively.

We will often write  $V(G)$  and  $E(G)$  to denote the set of vertices and edges associated with graph  $G$ , respectively. It is important to realize that an edge can actually be represented as an unordered tuple of two vertices, that is, its end points. For this reason, we make no distinction between  $\langle v, u \rangle$  and  $\langle u, v \rangle$ : they both represent the fact that vertex  $u$  and  $v$  are adjacent. A graph that does not have loops (edge joining the same vertices) or multiple edges in the same two nodes is called simple. When a simple graph has every pair of distinct vertices connected by a unique edge is called complete. [311]

Having no ordering is not always convenient. The need for associating a direction with the edges of a graph leads to the notion of a directed graph.

**Definition 3.1.2.** A directed graph  $D$  consists of a collection of vertices  $V$ , and a collection of arcs  $A$ , for which we write  $D = (V, A)$ . Each arc  $a = \overrightarrow{\langle u, v \rangle}$  is said to join vertex  $u \in V$  to another (not necessarily distinct) vertex  $v$ .

Once we have a directed graph, we need information about the travel frequency of every edge. The latter is obtained using a weighted graph: each arc has a weight that gives an indication on how many times it is traveled during a given period. A weight is a real-valued number associated with an edge.

**Definition 3.1.3.** A weighted graph  $G$  is a graph for which each edge  $e$  has an associated real-valued number  $w(e)$  called its weight. For any subgraph  $H \subseteq G$ , the weight of  $H$  is simply the sum of weights of its edges:  $w(H) = \sum_{e \in E(H)} w(e)$ .

A commonly adopted convention for weighted graphs is to simply write that  $w(\langle u, v \rangle) = 0$  when vertices  $u$  and  $v$  are not adjacent. This also means that for each edge  $e \in E(G)$  we demand that  $w(e) > 0$ . We often use weighted graphs to find subgraphs with a maximal (or minimal) weight. In particular, we can use them to determine the distance between two vertices, which is formally defined as follows.

**Definition 3.1.4.** Consider an undirected graph  $G$  and two vertices  $u, v \in V(G)$ . Let  $P$  be a path between  $u$  and  $v$  ( $(u, v)$ -path) having minimal weight among all  $(u, v)$ -paths in  $G$ . The weight of  $P$  is known as the (geodesic) distance  $d(u, v)$  between  $u$  and  $v$ . Path  $P$  is called a shortest path  $(u, v)$ -path, or a geodesic between  $u$  and  $v$ .

### 3.1.1 Graph Representations

It should be clear from the presentation so far that graphs can be drawn in different ways, but also that when considering their formal definition, they are merely described in terms of vertices and edges. Let us now pay attention to how we can

conveniently represent graphs. This issue is particularly important when we need to represent very large graphs for automated processing by computers.

There are several ways to represent graphs. Maybe the most appealing one is to use an *adjacency matrix*. Consider a graph  $G$  with  $n$  vertices and  $m$  edges. Its adjacency matrix is nothing else but a table  $A$  with  $n$  rows and  $n$  columns with entry  $A[i, j]$  denoting the number of edges joining vertex  $v_i$  and  $v_j$ . The adjacency matrix presents the following properties:

- An adjacency matrix of an undirected graph is symmetric, that is for all  $i, j$ ,  $A[i, j] = A[j, i]$ . This property reflects the fact that an edge is represented as an unordered pair of vertices  $e = \langle v_i, v_j \rangle = \langle v_j, v_i \rangle$
- A graph  $G$  is simple if and only if for all  $i, j$ ,  $A[i, j] \leq 1$  and  $A[i, i] = 0$ . In other words, there can be at most one edge joining vertices  $v_i$  and  $v_j$  and, in particular, no edge joining a vertex to itself.
- The sum of values in row  $i$  is equal to the degree of vertex  $v_i$

As an alternative, we can also use an Adjacency list of a graph as its representation. Given any graph, its adjacency matrix is made up of a square binary matrix whose row and column indices are the names of the vertices of the graph (see an example in Figure 3.1.2). Therefore, the adjacency list is convenient for node operations (i.e., insert, delete or add nodes), and the space cost is only  $O(|E|)$ , which benefits effective representations for large sparse graphs.

The Incidence matrix  $M$  of a graph  $G$  is another tool to describe its representation. It consists of  $n$  rows and  $m$  columns such that  $M[i, j]$  counts the number of times that edge  $e_j$  is incident with vertex  $v_i$ . Note that  $M[i, j]$  is either 0, 1, or 2: an edge can be only not incident with vertex  $v_i$ , it has vertex  $v_i$  as exactly one of its end points, or is a loop joining vertex  $v_i$  with itself.

Again, the following properties are easy to verify:

- A graph  $G$  has no loops if and only if for all  $i, j$ ,  $M[i, j] \leq 1$ .
- The sum of all values in row  $i$  is equal to the degree of vertex  $v_i$ .
- Because each edge has exactly two, not necessarily distinct end points, we know that for all  $j$ , the sum of each column  $M[i, j] = 2$ .

One of the problems with using either an adjacency matrix or an incidence matrix is that without further optimizations, the total number of elements for representing a graph is  $n \times n$  or  $n \times m$ , respectively. This is not very efficient when having to deal with very large graphs, especially when the number of edges is relatively small.

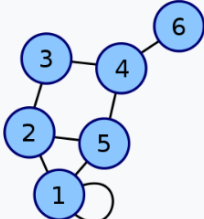
Labeled graph	Adjacency matrix
	$\begin{pmatrix} 2 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$ <p data-bbox="903 1182 1054 1205">Coordinates are 1–6.</p>

Figure 3.1.2: *Different ways to represent a graph.*

## 3.2 Individual Mobility Networks

The digital traces of human mobility can describe movements with extraordinary precision and can be exploited for realizing the most disparate location-based services [271]: recommender systems [23], personalized journey planners [122], carpooling systems [36], etc. Such services are based on appropriate modeling of human behavior, able to capture, for instance, the small set of actions that users typically repeat frequently [118] such as visiting a limited number of places [281]. In this section we introduce an approach for extracting the user's personal mobility model, which can enable several applications, from simulating the user's mobility, to predicting future events, such as crashes. The work stems from early results in [307] where the concepts of mobility profile and routines are defined, later used in [305] to build a trajectory predictor, and extends the initial proposal of [257], where a first simplified network-based model for mobility was used for classification tasks. The resulting model, named Individual Mobility Network (IMN) will be the basis for several other analysis tools described in the other sections of this thesis. An Individual Mobility Network (IMN) describes the individual mobility of a person through a graph representation of her locations and movements, grasping the relevant properties of individual mobility and removing unnecessary details. The challenge of defining a user's mobility model is commonly addressed in literature using three kinds of approach: Markov chains, mixture of general laws and pattern discovery.

An example belonging to the first category is [153] which characterizes and classifies user's Point of Interests (POIs) according to their relevance for the user: mostly visited POIs, occasionally visited POIs, exceptionally visited POIs, and build a Markov chain using the movements and the stops duration to weight the chain. In [233] the authors use Markov chains and a mixture of data-driven mobility laws to generalize the user's behavior from the geography, and to describe them in terms of their preferential exploration or return tendencies [234]. Then they use the models to generate synthetic trajectories maintaining some properties of original data, e.g. number of locations per user, radius of gyration, mobility entropy.

In [148] a non-parametric Bayesian method for modeling collections of timestamped events is proposed. The authors use a Dirichlet process for learning a set of intensity functions which form a basis set for representing individual time-periods which are exploited for "unusual" events detection. A general law approach is also followed by [118] where the authors state that human trajectories show a high degree of spatio-temporal regularity, each individual being characterized by a time-independent characteristic travel distance and a significant probability to return to a few highly frequented locations. The work in [175] presents a data-driven approach which uses mobility patterns. Those patterns are used to predict the future positions of a user when it matches the pattern premises.

Other approaches related to pattern discovery consider also external factors such as social relationship. In [316] a locations recommender is presented, based on past user behavior, the locations' venues, the social relationships and the similarity among users. In [67] it is empirically observed that the movements of a user can be

classified as short and long distance travels, where the former is based on periodic behaviors, while the latter is more likely to happen due to social influence.

Besides the type of model, another crucial aspect is the strategy adopted for managing the spatial and temporal dimensions. Most of the existing approaches discretize them using grids computed over the whole data. In [12] the authors map the real position of the users in an hexagonal grid. [343] makes use of a grid-based technique to extract interesting locations. This over-generalization of the locations simplifies the problem and makes it more manageable. The model we propose differs from those described above in various aspects: our approach is user-centric, no global knowledge is pushed from the expert standardizing the users in any aspect (e.g. discretization of space, number of locations, etc.); it is fully automatic and adaptive, no parameters are used by the analyst to drive the discovery of regularity (e.g. frequency thresholds).

### 3.2.1 Problem Formulation

The problem we face consists in summarizing the personal mobility of a user in a compact, yet rich network-based formalism that captures some of the essentials of the trips she performs: where she moves (places), how (trips), when (time of day, day of week, etc.) and how long (length and duration of trips and stops). We first introduce two basic concepts: trajectory, which is the basic input data type; and personal mobility history, representing the trips traveled by a user in a specific period [50].

**Definition 3.2.1** (Trajectory). A trajectory is a sequence  $t = \langle p_1, \dots, p_n \rangle$  of spatio-temporal points, each being a tuple  $p_i = (lon_i, lat_i, ts_i)$  that contains longitude  $lon$ , latitude  $lat_i$  and timestamp  $ts_i$  of the point. The points of a trajectory are chronologically ordered, i.e.,  $\forall 1 \leq i < n : ts_i < ts_{i+1}$

Given a trajectory  $t$  we refer to its  $i$ -th point  $p_i$  with the notation  $t[i]$ , and to its number of points with  $t.n$ . Also, we indicate the longitude, latitude and timestamp components of point  $t[i]$  respectively with the notation  $t[i].lon$ ,  $t[i].lat$ , and  $t[i].ts$ . For a timestamp  $ts$  we indicate its associated date and time of the day with  $date(ts)$  and  $time(ts)$  respectively.

**Definition 3.2.2** (History). The history  $H_u^{d,d'}$  of a user  $u$  is the set of trajectories traveled between dates  $d$  and  $d'$ :  $\forall t \in H_u^{d,d'} : d \leq date(t[1].ts) \leq date(t[t.n].ts) \leq d'$ .  $H_u^{d,d'}$  is denoted  $H_u^{d,d'}$  when  $d = d'$ , and  $H_u$  when  $d = -\infty$  and  $d' = \infty$ .

Our objective is to learn a personal mobility model by observing the personal mobility history. Most of the approaches in the literature related to modeling mobility behaviors suffer from various weaknesses. Indeed, in order to reduce the complexity of the problem generated by accurate GPS data, a very common procedure consists of employing forms of spatio-temporal discretization like a simple spatio-temporal grid. On one hand, this makes easier to find frequent or interesting areas and mobility patterns. On the other hand, it affects the precision of the applications they are

aimed for, since they can only infer areas with a granularity imposed by the apriori discretization. This weakness sometimes is overtaken by adopting smarter forms of spatial discretization, like clustering algorithms. Most of these algorithms require a parameter setting (e.g. the radius to decide when two points should belong to the same cluster, etc.) that is generally imposed to be equal for all the users. Unfortunately, it has been extensively proved that those kind of settings are frequently incorrect for the personal data of an individual user and might cause an algorithm to fail in finding the true patterns, or make the algorithm report patterns that do not really exist. Finally, also temporal discretization precludes a model from considering the time continuity. The model we developed tries to overcome the above limitations, and is able to capture the systematic presences of the user in her most frequent locations, and the routinary movements that lead the user from a location to another one. Moreover, the personal mobility model is built (i) without requiring any apriori spatial or temporal discretization, (ii) in an auto-adaptive fashion and without the need of any form of parameter tuning for different users, (iii) keeping time granularity sufficiently fine to approximate continuity. Given a user  $u$  and her history  $H_u = t_1, \dots, t_n$ , in the following of this section we define the components of the personal mobility model  $P_u$ .

**Definition 3.2.3** (Locations). *Given the stops of user  $u$ , we define locations  $L_u = L_1, \dots, L_k$  as a partitioning of her stops into disjoint sets of similar stops.*

Besides locations, the mobility of a user is characterized by movements, i.e., trajectories with a similar purpose.

**Definition 3.2.4** (Movements). *Given the history  $H_u$  and locations  $L_u$  of user  $u$ , we define her movements  $M_u = M_1, \dots, M_m$  as a partitioning of  $H_u$  into disjoint sets such that  $\forall M \in M_u : L, L' \in L_u$  s.t.  $M \subset H_u, \cup_{M \in M_u} M = H_u$  and  $M, M' \in M_u \wedge M \neq M' \Rightarrow M \cap M' = \emptyset$  where  $\forall m \in M_u, m = a_0, \dots, a_m, \text{ s.t. } \sum_{i=0}^{k-1} |m_i| = |H_u|$  and  $\forall M \in M_u, \exists L, L' \in L_u$  s.t.  $\text{start}(a) \in L \wedge \text{end}(a) \in L' \vee \text{start}(a) \in L' \wedge \text{end}(a) \in L$ .*

In other words, a movement is a set of trajectories which start from a location  $L$  and end in a location  $L'$ . Each trajectory belongs only to a movement. The locations and movements can be thought conceptually combined together using a network data structure which links the elements in a natural way from a mobility point of view, and is enriched with various spatio-temporal summaries of the original mobility, generating a individual mobility network :

**Definition 3.2.5** (Individual Mobility Network). *Given a user  $u$ , we indicate with  $G_u = (L_u, M_u)$  her individual mobility network, where  $L_u$  is the set of nodes and  $M_u$  is the set of edges. Given an aggregation operator  $agg$ , for each node  $l \in L_u$  we define the following functions:*

- $\omega(l) =$  number of trips in  $H_u$  reaching location  $l$ ;
- $\delta(l) = agg(\{\text{durations of stops in } l\})$ ;

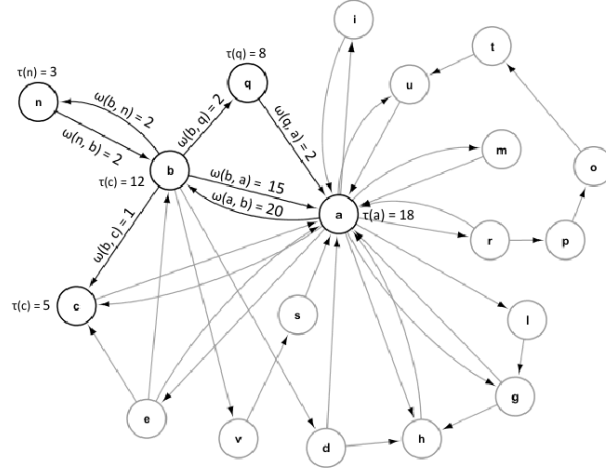


Figure 3.2.1: The IMN extracted from the mobility of an individual. Edges represent the existence of a route between locations. The function  $\omega(e)$  indicates the number of trips performed on the edge  $e$ , while  $\delta(x)$  the total time spent in a location  $x$

- $\rho(l) = \text{agg}(\{\text{arrival times of trips reaching } l\})$ ;
- $\pi_t(l) = \text{agg}(\{\text{durations of trips reaching } l\})$ ;
- $\pi_d(l) = \text{agg}(\{\text{lengths of trips reaching } l\})$ ;

Operator *agg* can return either a single value (e.g. median) or a n-ple (e.g. average and standard deviation, or quartiles). The same functions are also defined on edges (movements)  $m = (l_i, l_j) \in M_u$  in a similar way, this time considering only trips that start from  $l_i$  and reach  $l_j$ .

Nodes represent locations and edges represent movements between locations. We attach to both nodes and edges statistical information by means of structural annotations: edges provide information about the frequency of movements through the  $\omega$  function; nodes provide an estimation of the time spent in each location through the  $\tau$  function. To clarify the concept of IMN, let us consider the network in Fig. 3.2.1. It describes the IMN extracted from the mobility of an individual who visited 19 distinct locations. Location *a* has been visited a total of 18 time units (days in the example), since  $\tau(a) = 18$ . The edge  $e = (a, b)$  has weight  $\omega(e) = \omega(a, b) = 20$ , indicating that the individual moved twenty times from location *a* to location *b*.

The IMN of an individual is an abstraction of her mobility behavior. A location is an abstract entity without any reference to the geographic space. It can be interpreted as a subjective point of interest, a place around which the mobility of that individual gravitates. This allows the modeling of locations that are meaningful only for that individual, like his home or work place, etc. Accordingly, given the IMNs of two distinct individuals we are not able to determine whether they have visited the same location. But it allows to compare the behavior of multiple users

in different locations.

This, on the other hand, allows us to hide the actual places visited by the individual, providing a protection layer of sensitive information.

### **Conclusion**

In this section we have introduced IMNs, which represent a fundamental tool for analysis at the individual level, as it will be testified by the following sections, each presenting analysis tasks based on them. IMNs easily allow to identify regular travel habits and to link different parts of the mobility of the individual. Next sections will also show how further enrichment of basic IMNs is possible through deeper analysis of the individual data and its combination with collective and contextual information.



## 3.3 Segmentation

### Introduction

In mobility analytics one of the fundamental concepts is *movement*, namely the part of mobility data that describes a transfer from one place where the individual (or the object) was staying, to another one where the user will stop. Identifying movements in the raw stream of positions, for instance the continuous flow of GPS traces of a vehicle, is essential to many tasks, as it enables the development of mobility data models [257, 129] and applications like carpooling [127, 36], trajectory prediction [306] and car crash prediction [126], which are based on stop locations and the transitions between them.

Errors in identifying stops and movements greatly affect the results of modeling, and therefore the overall performances.

While it is simple to define a *stop* in geometrical terms, it is much less clear how to define *significant stops*, i.e. stops that might have some meaning for the user (for instance, stopping to do some activity before leaving), as opposed to stops that are simply incidental (for instance, due to a small traffic jam).

Practitioners in the mobility analytics domain defined several simple strategies to select stops in the mobility data stream (a brief account of literature on this topic is provided in the next section), each of them apparently capturing well some specific concept or some application-specific idea of stop. For instance, some solutions simply identify the moments where the object did not move, based on some thresholds, while others select the stops that have a duration compatible with some specific task, for instance discarding stops at a supermarket if their duration is physically too short to be able to enter, buy and exit. However, most existing solutions suffer from two important limitations: *(i)* they are based on critical thresholds that the user needs to choose accurately, and in most cases it is difficult to understand what value is the best; *(ii)* such thresholds are global, i.e. the same threshold value applies to all the individuals, irrespective of any distinctive characteristics they might have or of the places they visit. The reason of the latter is that, while an overall evaluation might be performed to select a global threshold, doing that separately for each individual might be impossible if their number is huge.

In this part of the thesis we try to overcome the limitations highlighted above, providing a general methodology that inspects the mobility of the individual, and identifies segmentation thresholds that apparently match their mobility features. The process allows to get rid of any input parameter, adapts thresholds to each individual and is completely automatic, thus applicable to large pools of users. Moreover, we extend the aforementioned approach by observing the typical stops of other users for areas in which the single individual behavior is not reliable due to low number of stops, and use the collective behavior to infer a suggested time threshold for the individual in those areas.

**Related Works** Segmentation is a technique for decomposing a given sequence into homogeneous and possibly meaningful pieces, or *segments*, such that the data in

each segment describe a simple event or structure. Segmentation methods are widely used for extracting structures from sequences, and are applied in a large variety of contexts [297], such as: time series [140, 50], genomic sequences [180, 235, 252], text [168], video data.

In the latter case, for instance, [200] proposes a trajectory segmentation approach for image motion, formulating it as an optimization problem aimed at minimizing the error between the observed motions and the segments approximating them.

However, it is generally adopted in preprocessing steps and it is generally not sufficiently analyzed and evaluated. Various simple approaches are currently adopted in practice, the most common being based on spatial and/or temporal constraints. This is the case, for instance, of the application paper in [307] where human trajectories are identified through a predefined rule based on a pair of spatio-temporal parameters regulating the end of a trajectory and the start of the subsequent one [131], where the trajectory is divided into subsequent trips if the time interval of “nonmovement” exceeds a certain threshold. In [344] it is described a change-point-based segmentation approach for GPS trajectories according to the transportation means adopting a universal threshold for determining whether a segment is “walk” or “nonwalk”. The work in [48] presents a theoretical framework that computes an optimal segmentation through computational geometry methods based on several criteria (e.g., speed, direction, location disk) that must be satisfied in each partition, thus making the approach local. However, the approach is rather general and not clearly applicable to the human trajectory context, where a trip can be complex and not showing the geometrical/movement uniformity the method looks for. Each criterion mentioned above corresponds to thresholds that the user must set, without clear guidelines on how to choose them. Finally, we remark that the implicit objective of such solutions is to identify the situations where the trajectory physically stops, regardless of its significance for the user. That allows to overcome the lack of a specific signal in the input data (e.g. car switch on/off) and the presence of artifacts introduced by GPS errors (e.g. the coordinates of an object change even if in reality it does not move), yet it does not distinguish between significant stops and irrelevant ones, which is a more semantic classification.

The authors of [332] segment the trajectories in two steps. The first segmentation is performed by means of simple policies aimed at splitting trajectories with respect to temporal and/or spatial predefined constraints. Then, the trajectories are divided into *stops* and *moves* observing variations of the speed of the object.

If the variations of the speed is below a speed threshold and there is a sufficient number of observations, then the portion of trajectory is annotated as a stop. The speed threshold is not general but changes according to the user behavior and also to the surrounding of the stop. In [277] it is defined a measure of the density of the points in the neighbourhood of each trajectory point, the Spatio-Temporal Kernel Window (STKW) statistics. To determine the start and end points of segments, the algorithm looks for maximal changes in STKW values. The focus of the approach is on capturing changes of transportation mode, including stops, which are simply points with low speed.

Besides to these methodologies, several other solutions to the trajectory segmentation problem have been proposed in the literature, yet with objectives different from ours. For example, cost-function based strategies were presented in [163, 162], while clustering-based ones are introduced in [173, 176], and a method based on interpolation kernels is described in blue[92, 93]. All these approaches are more focused on splitting a movement into homogeneous parts, rather than discovering significant stops, which is the purpose of this work.

From a more specific perspective, we can frame our proposal as a methodology for *stop-detection*, the segmentation being a consequence of selecting stops as cutting points. Along this direction, [298] presents a kernel-based algorithm to detect stop locations and estimate stop durations. The method does not analyze the points sequentially, and instead builds a kernel density surface from which it extracts local maxima that become activity location candidates from which to derive the stay time. In [6] it is presented an algorithmic framework for criteria-based segmentation of trajectories through a start-stop matrix that stores the relation between a trajectory and a criterion. In the criteria-based setting, segments are chosen such that the movement inside each segment is homogenous w.r.t a given criterion (e.g., on speed). The work in [342] describes a solution that derives the users' activity locations and times from data collected by their phones (GPS, GSM, WiFi, etc.). The main steps of the procedure consist in generating a first set of candidate stops according to predefined spatial/temporal windows, then in checking frequently visited places and in merging them, and finally in removing extra stops. A refinement of this procedure is presented in [268]. In [80] it is described a procedure that starts from fixed atomic segment of a homogeneous state, i.e., not moving or moving very little), and iteratively extends the segment until a new state is found. Similarly, [68] illustrates a method for threshold settings for stop detection focusing on periods of non-movement. In [146] stop points are detected using a density-based spatial clustering algorithm where a temporal criterion and gaps are also taken into account. Similarly, in [117] it is proposed a refined version of the DBSCAN clustering algorithm combined with SVM to identify the activity of stop locations. Finally, also [151] describes a cluster-centric trajectory segmentation approach exploiting movement characteristics such as position, direction, and speed of moving objects. Compared to these solutions, our proposal has a twofold objective, since we aim at simultaneously labeling a point as a stop and to refine the trajectory among two consecutive stops.

### 3.3.1 Trajectory Reconstruction

We start by defining trajectory segmentation based on a spatial and a temporal threshold, in a way similar to standard approaches in literature.

**Definition 3.3.1** (Individual Trajectory). Given a user  $u$ , her *Individual Trajectory*  $T_u$  is the sequence of  $n$  points  $T_u = \langle p_1, \dots, p_n \rangle$  that describes her position in time, where each point  $p \in T_u$  is defined as a triple  $p = (p.x, p.y, p.t)$ , representing its spatial coordinates  $x$  and  $y$ , and the corresponding timestamp  $t$ . Moreover, points

are in chronological order, i.e.  $\forall 1 < i \leq n. p_{i-1}.t < p_i.t$ .

First of all, we associate to each point a value corresponding to the time needed to move away from it farther than a spatial threshold:

**Definition 3.3.2** (Pseudo-Stop Duration). *Given an individual trajectory  $T = \langle p_1, \dots, p_n \rangle$  and a spatial threshold  $\sigma$ , the Pseudo-stop duration associated to point  $p_i$  is defined as  $SD(T, i) = \min\{p_j.t - p_i.t \mid i < j \leq n \wedge d(p_i, p_j) > \sigma\}$ , where  $d$  represents the geometrical Euclidean or geographical distance.*

Notice that the last point  $p_n$  will have  $SD(T, n) = \min \emptyset = \infty$ . Then, we define a partitioning of trajectories into segments, each terminating with a point having an high pseudo-stop duration:

**Definition 3.3.3** (Segmented Trajectory). *Given a trajectory  $T = \langle p_1, \dots, p_n \rangle$ , a spatial threshold  $\sigma$  and a temporal threshold  $\tau$ , we define the  $(\sigma, \tau)$ -segmentation of  $T$  as  $T^{\sigma, \tau} = \langle S_1, \dots, S_m \rangle$ , such that:*

- (i)  $\forall i$  s.t.  $1 \leq i \leq m$ ,  $S_i$  is a subsequence  $\langle p_s, p_{s+1}, \dots, p_e \rangle$  of  $T$
- (ii)  $\bigcup_{i=1}^m \text{set}(S_i) = \text{set}(T)$  and  $i \neq j \Rightarrow \text{set}(S_i) \cap \text{set}(S_j) = \emptyset$
- (iii)  $\forall S = \langle p_s, p_{s+1}, \dots, p_e \rangle \in T^{\sigma, \tau}$ ,  $SD(T, e) > \tau \wedge \forall j$  s.t.  $s \leq j < e : SD(T, j) \leq \tau$

where  $\text{set}(I) = \{p \in I\}$ .

Conditions (i) and (ii) imply that the segments of the segmented trajectory of  $T$  form a partitioning of the elements of  $T$  in the strictly mathematical sense. Moreover, condition (iii) states that all the points in a segment are movement points, i.e., their pseudo-stop duration is smaller than the given threshold, excepted the last point. Therefore, each point in  $T$  that has a high pseudo-stop duration will act as a split point, and corresponds to a distinct partition in  $T^{\sigma, \tau}$ . Also, an implicit consequence of the definition is that partitions are maximal, i.e., they cannot be extended with additional points and still satisfy the requirements of Definition 3.3.3.

### Problem Formulation

Existing trajectory segmentation methods assume that the same rules and the same parameters should apply to all moving objects. Since different objects can show very different movement characteristics, the above assumption leads to make choices that on average fit best the dataset, yet potentially making sub-optimal choices on single individuals.

Our objective is to overcome this limitation, making the segmentation process adaptive to the individual and taking into consideration their overall mobility. Our problem statement extends the traditional formulation of segmentation as a threshold-based operation, thus the core issue is to find good parameter values for each user.

**Definition 3.3.4** (Individual Cut Threshold Problem). *Given an Individual Trajectory  $T_u$ , and a global spatial threshold  $\sigma$ , the problem is to identify the temporal threshold  $\tau$  that yields the optimal segmentation  $T^{\sigma,\tau}$ .*

We notice that the problem definition requires a user-provided parameter  $\sigma$ . However, as it will be commented later in more detail, this is a single global threshold that only depends on location accuracy and is therefore expected to be rather easy to select for a given data source type.

In this work we also consider a generalization of the problem, where each user is actually associated to a set of thresholds instead of just one. In particular, we assume that the correct temporal threshold can depend on where the user is moving in each specific moment. We do that by first revising our definition of segmentation:

**Definition 3.3.5** (Space-Adaptive Segmented Trajectory). *Given a trajectory  $T = \langle p_1, \dots, p_n \rangle$ , a space partitioning  $G$  that maps points to geographical cells, a spatial threshold  $\sigma$  and a function  $\tau_G : G \rightarrow \mathcal{R}$  that associates a temporal threshold to each cell in  $G$ , we define the  $(\sigma, \tau_G)$ -segmentation of  $T$  as  $T^{\sigma,\tau_G} = \langle S_1, \dots, S_m \rangle$ , such that:*

- (i)  $\forall i$  s.t.  $1 \leq i \leq m$ ,  $S_i$  is a subsequence  $\langle p_s, p_{s+1}, \dots, p_e \rangle$  of  $T$
- (ii)  $\bigcup_{i=1}^m \text{set}(S_i) = \text{set}(T)$  and  $i \neq j \Rightarrow \text{set}(S_i) \cap \text{set}(S_j) = \emptyset$
- (iii)  $\forall S = \langle p_s, p_{s+1}, \dots, p_e \rangle \in T^{\sigma,\tau}$ ,  $SD(T, e) > \tau_G(G(p_e)) \wedge \forall j$  s.t.  $s \leq j < e : SD(T, j) \leq \tau_G(G(p_j))$

where  $\text{set}(I) = \{p \in I\}$ .

The change basically consists in replacing the fixed threshold  $\tau$  of the user with a set of values, one for each geographical cell visited by the user, formalized as a function from cells to thresholds. The problem now, therefore, becomes how to find the assignment of thresholds  $\tau_G$ .

**Definition 3.3.6** (Individual Space-Adaptive Cut Threshold Problem). *Given an Individual Trajectory  $T_u$ , a space partitioning  $G$  and a global spatial threshold  $\sigma$ , the problem is to identify the set of temporal thresholds  $\tau_G$  that yields the optimal space-adaptive segmentation  $T^{\sigma,\tau_G}$ .*

Since the number of moving objects can be very large, the process must be completely automatized and require no human intervention. In Section 3.3.2 we will introduce a simple and effective approach to solve the first problem, and thus find a suitable value of  $\tau$  for each user, also providing some basic guidelines to choose a value for the global spatial parameter. Then, Section 3.3.3 will revise the approach to tackle the space-adaptive problem definition, considering a more flexible context where the temporal threshold of a user can also change based on the areas visited, thus in principle yielding different values for different spatial locations.

### 3.3.2 Self-Adaptive Trajectory Segmentation

The first solution proposed for the individual cut threshold problem consists in fixing the spatial threshold to a global value (i.e. to be used for all users) and then in studying the segmentations that we would obtain by applying different temporal thresholds. We will start describing the process for choosing the temporal threshold, which is the central part of the solution, and later discuss how the spatial one can be chosen.

#### Methodology

When very small values of  $\tau$  are used, the segmentation obtained will contain a huge number of very short segments, till the extreme case where each point forms its own segment. As the threshold is increased, more and more segments will merge together, since some of the former splitting points will fall below  $\tau$ . The process is expected to gradually enlarge the trajectory segments by first including simple slowdowns (i.e. not really stop points), then temporary stops (e.g. at traffic lights), and so on.

Our approach consists in (virtually) monitoring such progression, and detecting the moment where an anomalous increase in the number of segments is observed, which represents a sort of *change of state*. This follows the same kind of idea adopted in various unsupervised classification contexts, such as the *knee method* for deciding the number  $k$  of clusters for the  $k$ -means algorithm [294], or analogous solutions to choose the radius for density-based clustering (e.g. DBScan).

In our solution, rather than relying on visual or similar heuristic criteria, we will base the threshold selection on a statistical test. In particular, we will adopt the Modified Thompson Tau Test [47] which, basically, checks whether a given value fits the distribution of the rest of the data or not [128]. Since we look for anomalous values in a sequence, we apply the test iteratively, comparing each value  $n(t)$  (the number of segments obtained with  $\tau = t$ ) against the values  $n(t')$  obtained for larger thresholds  $t'$ . This process yields a set of thresholds that have an anomalous number of partitions as compared to the successive thresholds. Among them, we simply choose the smallest one, thus deciding to select the segments that emerge at the first *change of state*, also representing shorter and finer granularity movements.

The procedure, named ATS (self-Adaptive Trajectory Segmentation) is summarized in Algorithm 1. Step 1 collects the pseudo-stop durations  $SD$  of all the points  $i$  of the input trajectory, and step 2 computes the frequency  $F$  of each value, basically representing the number of new segments obtained using that value as  $\tau$  w.r.t. the previous smaller thresholds. In our implementation such frequency distribution is computed through smoothed histograms, grouping values into bins of 1-minute width. Figure 3.3.1(left) shows the frequency distribution of a sample trajectory, the vertical line corresponding to a possible cut point. The resulting set of segments obtained is described in Figure 3.3.1(right) in terms of segments duration. Finally, step 3 applies the Modified Thompson Tau Test to all possible cut thresholds, corresponding to all the non-zero values of frequency function  $F$ , to identify

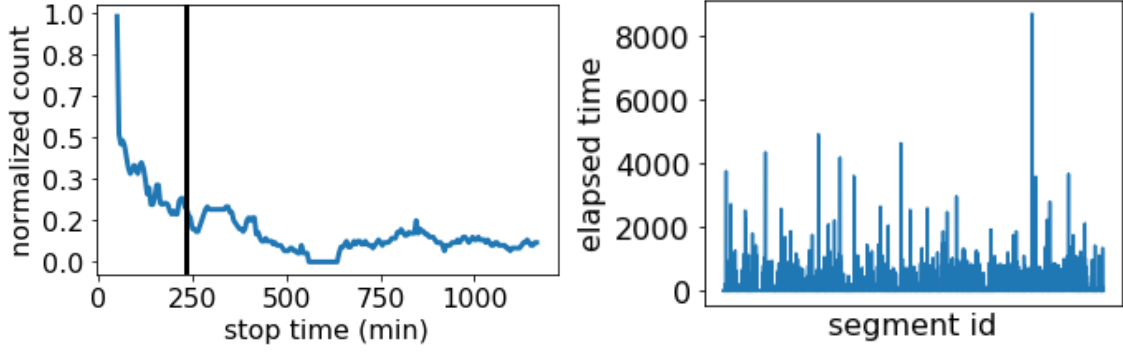


Figure 3.3.1: *Frequency distribution of pseudo-stop durations for a user trajectory (left), and the durations of the segments obtained using a specific threshold to cut the trajectory (right). The threshold used corresponds to the vertical line on the left image.*

the frequency values that appear to be anomalous with respect to the frequency of larger thresholds. Among all these candidate thresholds, step 4 selects the smallest one as value for  $\tau$ .

**Computational Complexity.** The cost of Algorithm 1 is dominated by step 1, since the computation of each pseudo-stop duration ( $SD$ ) could in principle require to scan all the remaining points of the individual trajectory, thus yielding a  $O(n^2)$  cost, where  $n$  is the size of the individual trajectory. However, in practical applications the trajectory portion needed for each  $SD$  is relatively small, leading to a quasi-linear cost. The remaining parts of the algorithm can be realized in linear time, including the Modified Thompson Tau Test, which can be computed for each point through incremental updates, and considering that the actual size of  $F$  (namely, not considering empty bins where  $F(a) = 0$ , which can be simply omitted) is at most  $n = |T|$ .

---

**Algorithm 1:**  $ATS(T, \sigma)$

---

**Input** : Individual trajectory  $T$ , spatial threshold  $\sigma$

**Output:** Cut threshold  $\tau$

- 1  $S = \langle SD(T, i) \mid 1 \leq i \leq |T| \rangle$ ;
  - 2  $F =$  frequency distribution of  $S$  values with 1-minute bins  
( $F(a) = |\{a \in S\}|$ );
  - 3  $C = \{t \mid t \in range(F) \wedge TT(F(t), \langle F(t') \mid t' > t \rangle) = true\}$ ;  $TT(a, B) =$   
Modified Thompson Tau Test of  $a$  vs. set  $B$
  - 4 **return**  $\min C$
-

### Fixing the spatial threshold

In our approach, the threshold  $\sigma$  represents the minimum distance between two (consecutive) points that can be considered as a movement, and the temporal parameter is indeed measured as the time needed to make a movement. A simple way to fix its value is to adopt the minimum value that, according to the accuracy of our dataset, cannot be mistaken for a positioning error, for instance due to GPS uncertainty. In our experiments we adopt road vehicle GPS traces that are expected to have errors not larger than 10 meters, therefore we could fix  $\sigma = 20$  (the worst case distance between two points that have the maximal error in opposite directions). We decided to slightly increase it to 50 in order to stay on the safe side, also to take into account that errors are slightly higher than average in urban centers, which is the application context where our experiments are performed. Since we do not have data sources from other kinds of transport (ships, planes, etc.) the selected threshold seems to meet our purposes. However, empirical results confirm that the value of the global parameter  $\sigma$  is not critical, as our approach shows a low sensitivity to it. For this reason, the value we chose in our experiments (50 meters) can be considered a good guess for generic vehicle GPS data. Other data sources with a higher spatial uncertainty might require larger values, to be ascertained through a (one-shot) analysis of the data produced.

### 3.3.3 Individual and Collective Adaptive Trajectory Segmentation

The solution described in the previous section strictly follows the problem formulation of  $(\sigma, \tau)$ -segmentation given in Definition 3.3.3, thus implicitly assuming that a user has a single, optimal threshold that applies well in any area where they move. Clearly, common sense suggests that this is an artificial assumption, and the threshold that is correct for a user in a given place, might be not optimal for the same user in a different location.

To loosen such assumptions, we adopt here the more general notion of space-adaptive segmented trajectory, introduced in definition 3.3.5, and a corresponding strategy to adapt the thresholds also to geographical locations.

The problem, now, consists in finding for each user an assignment of thresholds  $\tau_G$  that provides a (potentially different) threshold value for each geographical cell in the space partitioning  $G$ . We identify here three possible approaches:

1. *Local individual approach*: following the same idea of *ATS*, we could restrict the statistical test used to fix the threshold  $\tau$  only to the points of the user that fall in a given cell  $g \in G$ . While very appealing, empirical evaluations show that the data samples associated to each cell are too small to apply the test, with very few exceptions. For this reason, alternative solutions were considered.
2. *Local collective approach*: this solution assumes that the time threshold  $\tau$  is actually a function of the location, and does not directly depend on the user.



Therefore, each cell  $g$  is associated to a data sample composed of all the points of all users that fall in  $g$ . This greatly increases the sample size, yet losing the identity of the single user.

3. *Wisdom-of-the-crowd collective approach*: the idea here is that each user brings an opinion about what is the correct threshold, built from their own mobility data (and therefore their own  $\tau$  found through *ATS*), and over each cell  $g$  all users vote for the best local threshold value, each vote having a weight proportional to the frequency of visit of the cell. This is a simple application of the classical “wisdom of the crowd” principle [108].

Approaches 2 and 3 provide a candidate threshold value  $\tau^*$  for a user in a given cell, which can be seen as a suggestion that all users provide as alternative to the individual value  $\tau$ . Our proposal is to replace  $\tau$  with  $\tau^*$  whenever the former has a weak relation with the cell, i.e. for those locations that the user visited only rarely, and that therefore were not significantly involved in the computation of the global  $\tau$ . Both collective approaches result into a mapping  $G_C : G \rightarrow \mathcal{R}$  that associates each geographical cell in  $G$  to a suggested  $\tau^*$ . The procedure that implements the management of such suggestions is the same for both approaches, is named ACTS (self-Adaptive and Collective Trajectory Segmentation), and is summarized in Algorithm 2.

---

**Algorithm 2:** ACTS( $T, \sigma, G_C, min\_stops$ )

---

**Input** : Individual trajectory  $T$ , spatial threshold  $\sigma$ , Cell grids and associated collective threshold multisets  $G_C$ , Minimum number of stops  $min\_stops$ .

**Output:** Cut thresholds  $\eta$

```

1  $\tau = ATS(T, \sigma)$ ;
2  $G_I = \{ (g, freq) \mid g \in G_C \wedge freq = |\{p \in T \mid p \in g\}| \}$ ; // visited cells and
   frequency
3  $\eta = \emptyset$ ;
4 for  $(g, freq) \in G_I$  do
5    $\mu = mode(S)$  for  $(g, S) \in G_C$ ;
6   if  $freq \geq min\_stops$  then
7      $\eta = \eta \cup \{(g, \tau)\}$ ; // individual threshold prevails
8   else
9      $\eta = \eta \cup \{(g, \mu)\}$ ; // collective threshold prevails
10 return  $\eta$ 

```

---

Besides the individual user trajectories  $T$  and the spatial threshold  $\sigma$ , the ACTS procedure takes as input the cell grid  $G_C$  containing the pseudo-stop times of all the observed users grouped per cell, and the minimum number of stops  $min\_stops$  that an individual user can have in a cell in order to consider the cell “frequently visited”. In the first step, ACTS retrieves the user adaptive threshold  $\tau$ . After that, it

identifies the subset  $G_I \subseteq G_C$  of cells visited by the user, with their visit frequencies. Then, for each cell  $g$  (lines 4–10), if the cell is frequently visited, i.e., the user has in that area at least  $min\_stops$  points, then the individual global threshold  $\tau$  is used (line 7), otherwise we take the most frequent value among those associated to the cell  $g$  (lines 5 and 9).

In order to specify what kind of threshold suggestions we are using in the ACTS procedure, we will refer to it as  $ACTS_{Local}$  when  $G_C$  is obtained through the local collective approach (number 2 of the list above), and as  $ACTS_{WOTC}$  when the Wisdom-of-the-crowd approach is used.

**Spatial Grid Definition.** In principle, any definition of grid  $G$  can be applied to ACTS, provided that it is a partition of space that covers all points in our users' trajectories. In our experiments we opted for a regular grid, which is the most commonly adopted solution in literature, and in particular we implemented it through a standard *geohashing*. *Geohash* [210] is a very efficient mapping of locations into rectangular grids, and allows to change its spatial granularity in a transparent way. Its main limitation is in the fact that grids are predefined worldwide, and the spatial granularity can be changed in a limited set of configurations, the size of each cell doubling when we move from one granularity level to the next one. Other, more sophisticated space partitioning strategies are used in literature, such as regular exagonal grids [283] or quad-tree based adaptive partitioning [101], yet evaluating all of them is out of the scope of this thesis. Given an encoding length  $h$ , Geohash associates each pair of latitude-longitude coordinates to a string of  $h$  letters and digits, which corresponds to define a partitioning into square or rectangular cells, each cell corresponding to the set of points that have the same encoding. In particular, we will consider three levels:  $h = 5$ , resulting into cells of diameter  $\sim 4.8$  Km;  $h = 6$ , with diameter  $\sim 1.22$  Km; and  $h = 7$ , with diameter  $\sim 0.152$  Km.

Algorithm 3 summarizes the overall process, including the generation of grid  $G$  and collective suggestions  $G_C$ , for both variants of ACTS.

## Evaluation Measures

The reconstruction error generally used for evaluating segmentation problems [43] just measures how well each segment can be approximated with one value, and thus seems not to fit with trajectory segmentation. Therefore, similarly to clustering evaluation, we propose three internal evaluation measures [294]. Let  $T$  be the sequence of  $n$  points and  $T_S = \langle S_1, \dots, S_m \rangle$  its segmentation. We denote with  $A_t = duration(T) = p_n.t - p_1.t$  the total elapsed time from the first point of  $p_1 \in T$  to the last point  $p_n \in T$ , and  $A_d = length(T) = \sum_{i=1}^{n-1} d(p_i, p_{i+1})$  the total distance covered by the trajectory, computed by considering every couple of subsequent points in  $T$ . Let  $M_t = \sum_{S_i \in T_S} duration(S_i)$  be the sum of the segments' duration, i.e., the time spent driving, and  $M_d = \sum_{S_i \in T_S} length(S_i)$  be the sum of the segments' length, i.e., the distance traveled. Then, we define the following measures:

- *time precision*:  $TP = 1 - M_t/A_t$
- *distance coverage*:  $DC = M_d/A_d$

---

**Algorithm 3:** ACTS<sub>ALL</sub>(Method,  $\mathcal{T}$ ,  $\sigma$ ,  $h$ ,  $min\_stops$ )

---

**Input** : Method to apply (Local or WOTC), Individual trajectories of all users  $\mathcal{T}$ , spatial threshold  $\sigma$ , geohash level  $h$ , Minimum number of stops  $min\_stops$ .

**Output:** Segmented trajectories  $\mathcal{T}^*$

- 1  $G = \{geohash(lat, lon) | T \in \mathcal{T} \wedge (lat, lon) \in T\};$
- 2 **if** Method = ‘Local’ **then**
- 3      $G_C = \{(g, \tau^*) | g \in G \wedge S = \langle SD(T, i) | T \in \mathcal{T} \wedge T[i] \in g \rangle \wedge \tau^* =$   
         $ATS_{geo}(S, \sigma)\};$
- 4 **if** Method = ‘WOTC’ **then**
- 5      $G_C = \{(g, \tau^*) | g \in G \wedge S = \langle ATS(T, \sigma) | T \in \mathcal{T} \wedge T \cap g \neq \emptyset \rangle \wedge \tau^* =$   
         $mode(S)\};$
- 6  $\mathcal{T}^* = \emptyset;$
- 7 **for**  $T \in \mathcal{T}$  **do**
- 8      $\tau_G = ACTS(T, \sigma, G_C, min\_stops);$
- 9      $\mathcal{T}^* = \mathcal{T}^* \cup T^{\sigma, \tau_G};$      //  $(\sigma, \tau_G)$ -segmentation, see Def. 3.3.5
- 10 **return**  $\mathcal{T}^*$

---

- *mobility f-measure:*  $MF_\beta = (1 + \beta^2) \cdot TP \cdot DC / ((\beta^2 \cdot TP) + DC)$

Time precision and distance coverage capture two conflicting effects of segmentation, namely the time covered by stops and the distances covered by the segments (i.e. the movement points). Indeed, a very *aggressive* segmentation will identify a large number of stop points, yielding a high time precision, yet this will make segments shorter, significantly reducing the distance coverage. Similarly, a very *loose* segmentation will yield exactly the opposite results. Any segmentation choice will yield a trade-off between them. Analogously to the f-measure adopted in Information Retrieval, which is a combination of precision and recall measures, our *mobility f-measure* accounts for both aspects simultaneously. In the experiments we adopt  $\beta = 0.25$ , which weighs *time precision* higher than *distance coverage* by augmenting the relevance of missing precision in stop detection. The reason is that *i*) it is relatively easy to guarantee an high distance coverage, and *ii*) the main focus of the work is on the temporal aspects of trajectory partitioning.

### 3.3.4 Experiments

We experimented the proposed trajectory segmentation approaches ATS and ACTS over real datasets of GPS vehicle traces. The results commented in the following refer to 2000 users of the area of Rome (Italy), and London (UK). The means and standard deviations of the sampling rate for the users analyzed are  $12194.67 \pm 22575.66$  and  $4385.76 \pm 9359.14$ , for Rome and London respectively. The high values and their high variability is due to the presence of several long time gaps, typically due to parking periods.

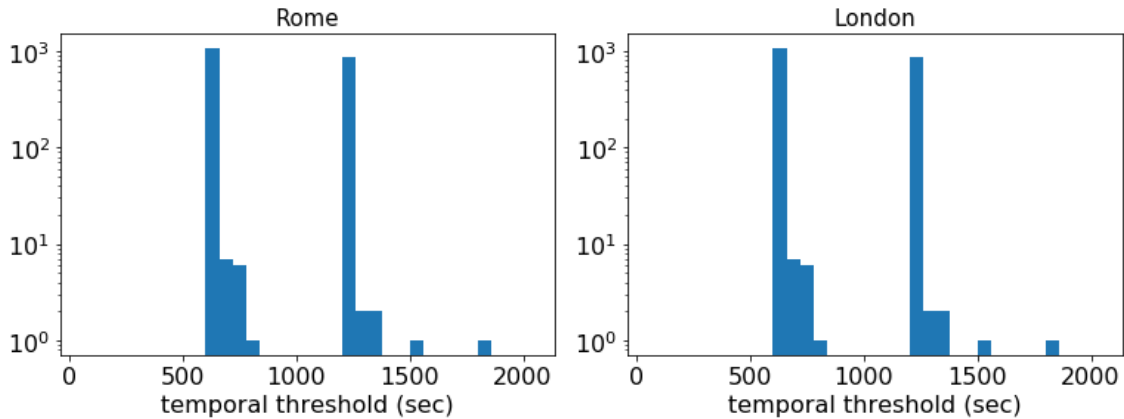


Figure 3.3.2: *Time threshold distributions for trajectories obtained with ATS in Rome and London. The peaks show the ideal thresholds to be set to build the trajectories.*

In the following, we first analyze the personal temporal thresholds returned by ATS, and then provide a quantitative and qualitative evaluation of the results for understanding the benefits of the novel method with respect to existing ones. We compare, in particular, against the common trajectory segmentation method based on fixed parameters ( $FTS_{\text{temp-thr}}$ ) as proposed in [307]. Moreover, we consider a baseline consisting in a random trajectory segmentation method that splits the sequence of points  $T = \langle p_1, \dots, p_n \rangle$  into  $m$  equal-length segments (*i*) with  $m$  randomly extracted between 2 and  $n/2$  ( $RTS_1$ ), or (*ii*) with  $m$  set to the number of segments returned by ATS ( $RTS_2$ ).

Next, we show the results obtained with the two variants of ACTS,  $ACTS_{LOC}$  and  $ACTS_{WOTC}$ , thus evaluating the impact of considering geography and collective behaviors in the definition of individual temporal thresholds. Here, we compare our proposed solutions against a state-of-the-art approach for trajectory segmentation exploiting a completely different strategy but relying exactly on the same input data format. We name HEH-D the proposal described in [146] for detecting stop points using the DBSCAN method. In summary, HEH-D first runs DBSCAN on the GPS observations only considering the spatial dimension. Then, it further separates the points in each cluster that have a temporal gap between each other larger than  $q$  seconds, and turns into noise the spatio-temporal clusters composed by less than  $k$  points. Finally, all the noise points are sorted chronologically and modeled as trajectories while those in the clusters are treated as stop points. According to the suggestions in [146], we adopted the following parameters setting:  $min\_pts = 5$ ,  $\epsilon = 50$  meters,  $q = 210$  seconds. Also, for the parameter  $k$  we evaluated all values between 2 and 6, and eventually selected  $k = 2$  since it yields the best results in terms of mobility f-measure. Additionally, we also experimented with a variant of this method that replaces DBSCAN with OPTICS, named HEH-O in the experiments, for which we adopted the same parameters specified for HEH-D. The idea is that

method	$MF_{.25}$	$TP$	$DC$	$ratio_{sr}$	$\#segms (avg \pm std)$
ATS	<b>.951</b>	<u>.951</u>	<u>.981</u>	<u>0.049</u>	837.34 $\pm$ 854.52
FTS <sub>120</sub>	.925	<b>.996</b>	.456	<b>0.015</b>	592.26 $\pm$ 652.78
FTS <sub>1200</sub>	<u>.948</u>	.947	<b>.997</b>	0.053	746.28 $\pm$ 733.96
RTS <sub>1</sub>	.279	.268	.722	0.700	2094.85 $\pm$ 2472.36
RTS <sub>2</sub>	.124	.118	.877	0.883	899.59 $\pm$ 926.03

Table 3.3.1: *Evaluation on Rome data. The first three columns show the measures illustrated in Section 10. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of segments.*

method	$MF_{.25}$	$TP$	$DC$	$ratio_{sr}$	$\#segms (avg \pm std)$
ATS	<u>.955</u>	<u>.953</u>	<b>.999</b>	<u>0.047</u>	433.92 $\pm$ 513.72
FTS <sub>120</sub>	<b>.958</b>	<b>.961</b>	.944	<b>0.040</b>	1131.83 $\pm$ 1431.81
FTS <sub>1200</sub>	.952	.950	<b>.999</b>	0.050	359.55 $\pm$ 410.61
RTS <sub>1</sub>	.267	.256	.695	1.007	2833.72 $\pm$ 4203.05
RTS <sub>2</sub>	.035	.033	.958	1.008	445.65 $\pm$ 527.97

Table 3.3.2: *Evaluation on London data. The first three columns show the measures illustrated in Section 10. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of trajectories.*

OPTICS typically performs better than DBSCAN when clusters in the data have variable densities, and that might help improving the quality of the segmentation.

Finally, we conclude the section with an evaluation of run times of our methods when the number of users and the duration of their trajectories vary.

### Self-Adaptive Temporal Threshold (ats)

We observe in Figure 3.3.2 the distribution of the time thresholds selected by ATS for each user (vertical axis represents value frequencies in log-scale).

Although every user has her own mobility with its own mix of regular and more erratic behaviors [231], we observe two clear peaks in the distributions for both Rome and London. This means that ATS mainly recognizes two different types of users regarding to the minimum duration of the stops. This supports the intuition behind our approach, namely to have a self-adaptive procedure selecting a personalized best temporal threshold for each user. Selecting one single threshold value for all the data might negatively affect the segmentation of some users, partitioning their trajectories either too much or too little. The first peak is at about 600 seconds ( $\sim$  10 minutes), while the second peak is around 1200 seconds ( $\sim$  20 minutes). These values correspond to the temporal thresholds that the ATS procedure uses to cut each trajectory. There is also a minority of users having values relatively far from the peaks.

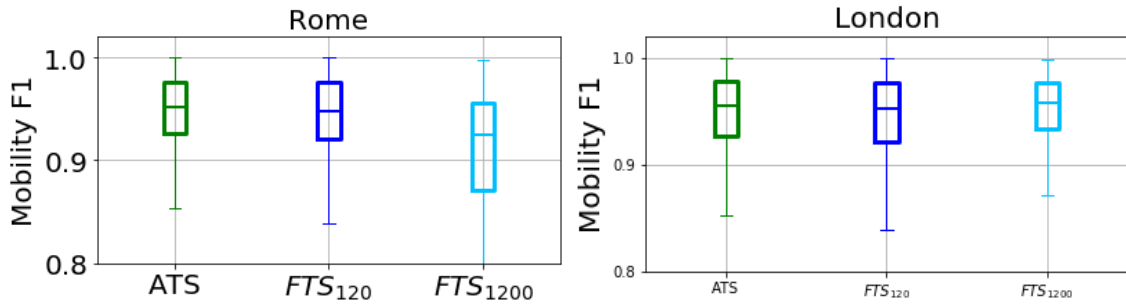


Figure 3.3.3: *Boxplots for the  $MF_{.25}$  results. On the Rome data ATS yields better results than the FTS solutions, while in London all three produce almost the same results. The variability of ATS results is consistently smaller than the other methods, which is a sign of robustness.*

**Comparison of Evaluation Measures** In this section we compare the ATS approach with the baseline methods taken into account. In Tables 3.3.1 and 3.3.2 we report the results obtained on our two cities. The first three columns show the evaluation measures described above. The fourth column shows the ratio between the average sampling period of movement points (thus discarding the stop portions of the user’s trajectory) and the average sampling period of the full trajectory, while the last one reports the average number of segments obtained and its standard deviation. In general, we can observe that the best results were obtained with the ATS and FTS methods, both for Rome and London. Analyzing the ratio (fourth column) we can see that values are low for both ATS and the FTS ones, meaning that the long stops are ignored (i.e. they are recognized as real stops) and just the short ones are considered. On the contrary, with the random approaches the ratio is bigger because the algorithm function evaluates all stops in the same way. Looking at the number of segments it is possible to note that FTS and ATS methods produce different quantities, especially the  $FTS_{120}$  produces less segments in the Rome case and much more in London. About the last two approaches, the  $RTS_1$  method works with a random number of segments, so it is normal that the final result differs from the others, while the  $RTS_2$  takes as number of segments the same of the ATS approach so we expect to achieve similar results.

For the evaluation measures we can see that ATS reached the goal we expected, i.e. yielding a quality of results which is always comparable or higher than fixed-threshold approaches and more robust. Indeed, for both Rome and London the values obtained by ATS are compatible with the FTS results, even better in the  $MF_{.25}$  for Rome and in the distance coverage for London. In particular, in the Rome example, having a high  $MF_{.25}$  values means that also the time precision and the distance coverage are well correlated, leading to satisfying result. Looking at the  $FTS_{120}$  results, we can note that the time precision is high but the distance coverage is very low, because the algorithm builds short trajectories with few points. An analogous reasoning can be done analyzing the  $FTS_{1200}$  method, which produces

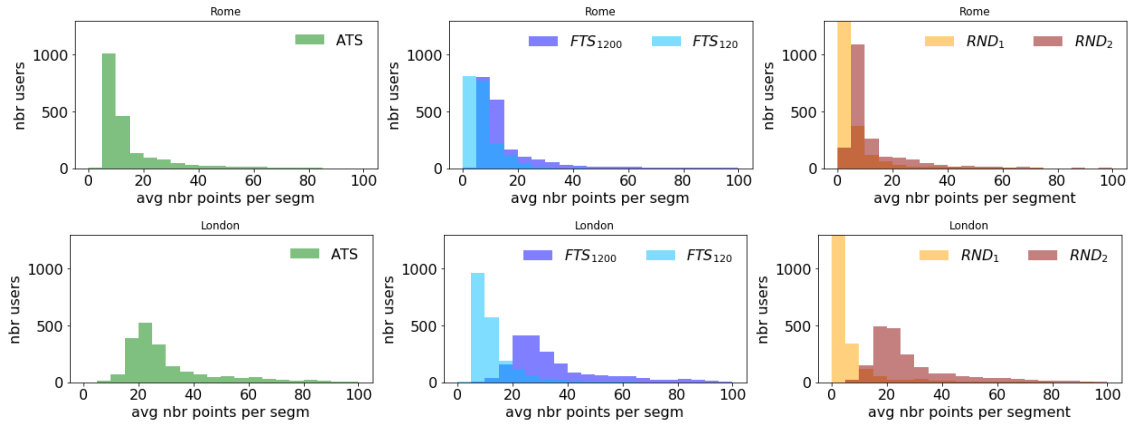


Figure 3.3.4: *Distributions of average number of points per segment obtained by ATS.*

an excellent distance coverage score but a lower time precision. The ATS solution reaches a good balance, thanks to its adaptive behaviour that allows to control and correct the trajectory fragmentation, and all its evaluation measures are always either the best or the second best of the group.

For a better understanding of the quality of ATS, the distribution of  $MF_{.25}$  values for the different approaches on the two datasets is shown in Figure 3.3.3 through a boxplot visualization. For the Rome case we can observe that with the ATS approach the median value is the highest (closest to 1) and the inter-quartile range is smaller than the other two, meaning that we have a smaller variability and thus more robust results. The London case appears to be different, and the best  $MF_{.25}$  values are obtained with the  $FTS_{1200}$ , although the median is very similar to ATS and the box is only slightly narrower. This indicates that in some contexts the flexibility introduced by ATS might be not required, and it only reaches performances similar to those of simpler solutions.

**Comparison of Segmentation Statistics** In the following we analyze other statistical indicators on the trajectory segments extracted by the various methods. Indeed, discovering some hidden correlations between trajectory features and the segmentation approach could lead to a better understanding of the problem and highlight other relevant aspects. In Figure 3.3.4 we report the distributions of the average number of points per segment for Rome and London. For all methods, the majority of segments have less than 20 points, probably meaning that most of the trips take place within the city. However, in the distribution tails some long trajectories with more points emerge. We observe that the distribution peaks of ATS place somehow in between the peaks of the two FTS variants (though closer to  $FTS_{1200}$ , especially in London) thus finding a trade-off between them. Moreover, we can see that the distributions are different in the two cities: London has a wider distribution than Rome, meaning that the first one has a larger variety of trips.

Figure 3.3.5 shows the distributions of the average number of segments per user.

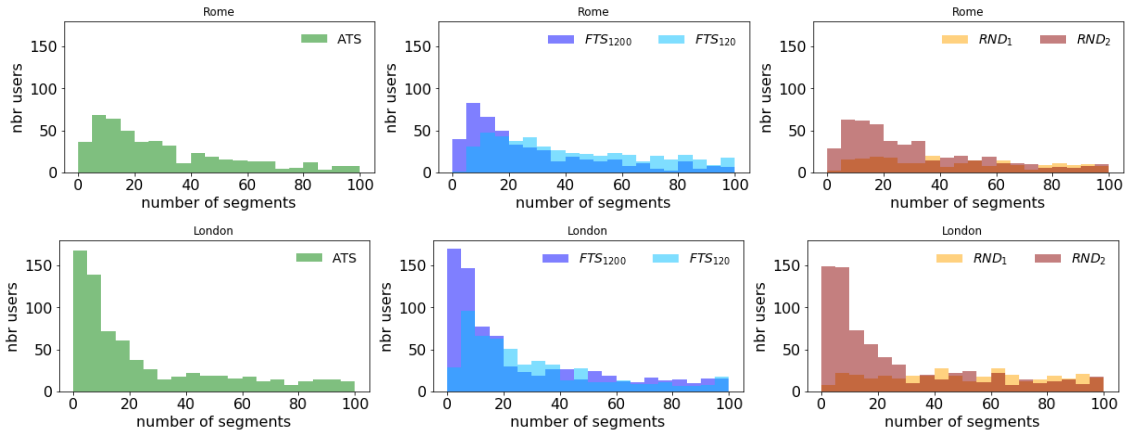


Figure 3.3.5: *Distribution of the number of trajectory segments over Rome (top) and London (bottom) with each segmentation method (on the columns, grouped by family).*

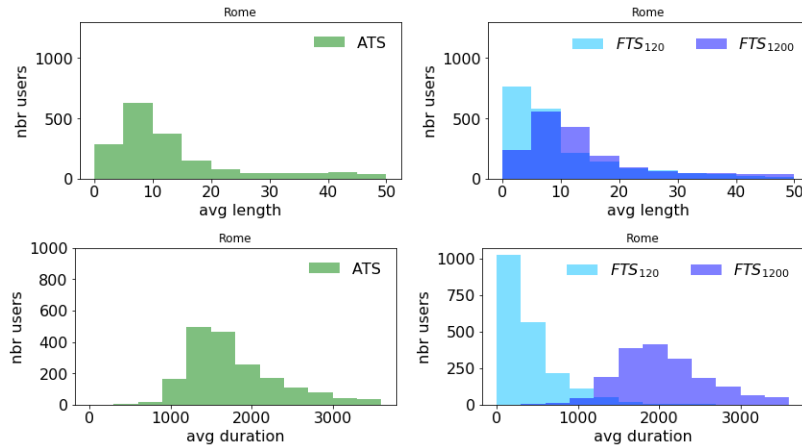


Figure 3.3.6: *Distributions of the average length (top) and duration (bottom) for the trajectory segments returned by ATS (left) and FTS (right) for the area of Rome.*

In London most of the users have less than 20 trajectory segments. The peak of the distribution is between 5 and 10. Between 30 and 100 segments the distribution remains stable at a small value larger than zero. In Rome we observe a similar result with a peak between 15 and 20. Also in this case, the peak of ATS distribution tends to stay in the middle of the FTS ones.

In Figure 3.3.6 we compare the distribution of average length and average duration of the segments returned by ATS (left) and FTS (right) for the area of Rome. With the ATS method the peak value is around 10km, thus confirming that most of the trips are short, and likely to take place around the city. With the FTS methods the peak position depends on the temporal threshold imposed: with a threshold of 1200 seconds the average distance is similar to ATS, while with 120 seconds it becomes lower and close to 5 km. The results for the RTS methods are omitted,



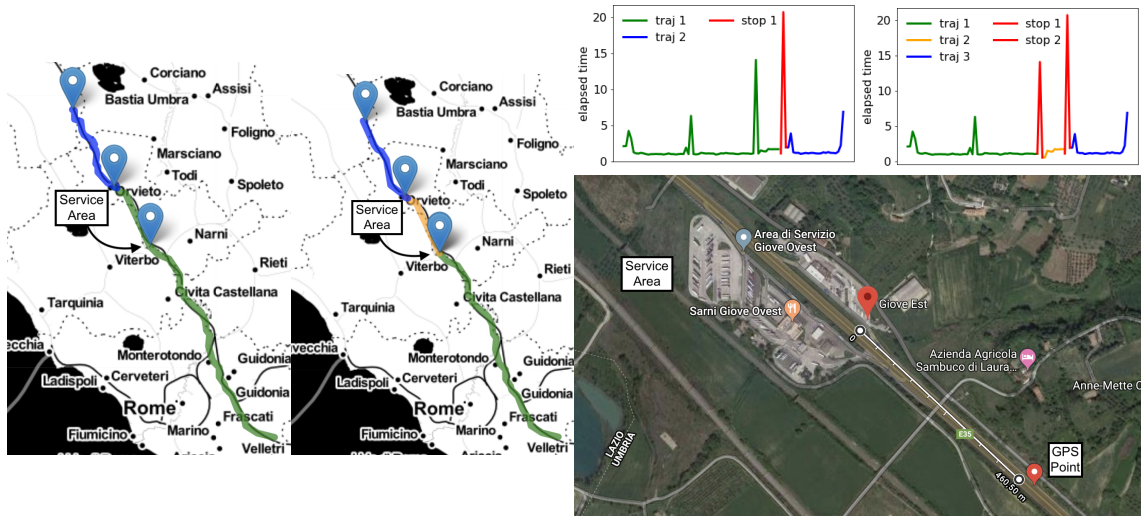


Figure 3.3.7: *Trajectory segmentation returned by  $FTS_{1200}$  (left) and  $ATS$  (right). The user is traveling from South to North. Top: spatial representation showing the trajectory segments. Center: temporal segmentation showing the inter-leaving time between GPS points. Bottom: zoom on the service area highlighted in the top maps where the user probably stops for  $\sim 15$  minutes. Best viewed in color.*

since their plots are very similar to  $FTS$ . Also, the plots for London show exactly the same behavior observed on Rome.

In terms of segments duration,  $ATS$  yields a distribution with a peak around 1200 – 1500 seconds ( $\sim 20 - 25$  minutes). With the  $FTS$  methods the peaks change: for  $FTS_{120}$  the peak is around 500 seconds while for  $FTS_{1200}$  the peak is centered in 1800 seconds. Also in this case, the results on London are very similar and omitted here.

### Case Study

In this section we show qualitatively on a case study the effectiveness of  $ATS$  with respect to  $FTS$ . In Figure 3.3.7 we report the segmentation returned by  $FTS_{1200}$  [307] (left) and by  $ATS$  (right), the user is traveling from south to north.  $FTS_{1200}$  returns two trajectories (green and blue), while  $ATS$  returns three trajectories (green, orange and blue). The second line of plots reports the time gap between consecutive GPS points. The colors match the trajectory segments, while stops are highlighted in red. We observe how  $ATS$  identifies the short stop of less than 15 minutes at the service area similarly to the subsequent longer stop. On the other hand,  $FTS_{1200}$  considers the first stop as part of the green trajectory. The map in the bottom line of Figure 3.3.7 shows the service area which is very close to the GPS points reported on the bottom right corner of the map. This case study highlights how various existing stops under a certain predefined threshold can be missed with a segmentation approach like  $FTS$ , while a more data-driven and self-adaptive method like  $ATS$  is able to take into account specific user behaviors and return more detailed

results.

### Individual and Collective Adaptive Temporal Threshold (acts)

In this section we show the impact and improvements given by the ACTS methods exploiting the collective behavior over ATS. First of all we choose a reference geohash precision looking for a trade-off between the geographical granularity and the number of pseudostops collected in the cells. We opt for a geohash precision of  $h = 6$  corresponding to an area of size  $1.22km \times 0.61km$ . As shown in the next sections, values  $h = 5$  and  $h = 7$  yield very similar results, suggesting that any value of  $h$  around 6 appears appropriate for this kind of data. The set  $G$  of cells obtained this way are used by Algorithm 3 to compute local suggestions for time thresholds by collecting the pseudo stops of all the 2000 users in the dataset under consideration. Then, for the ACTS<sub>LOC</sub> method a distribution of pseudo-stop durations for every cell is created, which will later pass through the Thompson test. Both ACTS strategies require to define the minimum number of points (visits) of a user in a cell that make it significant for them. In order to avoid any manual setting, after a preliminary experimentation we decided to derive it directly from the distribution of pseudo-stops durations of the dataset, fixing it to its 50-th percentile. In our dataset, in particular, that corresponds to  $min\_number = 5$ , meaning that when a user passes in a given cell, if they have at least 5 points inside it, for that cell we can use their own individual time threshold computed by ATS; otherwise, we will use the collective threshold assigned to the cell.

We report in Figures 3.3.8 and 3.3.9 the distributions of the time thresholds selected respectively by ACTS<sub>LOC</sub> and ACTS<sub>WOTC</sub> (Rome dataset on the left and London on the right) for each user (vertical axis represents value frequencies in log-scale). Similarly to Figure 3.3.2, we can observe two peaks in the distributions at about 600 seconds ( $\sim 10$  minutes) and 1200 seconds ( $\sim 20$  minutes) for both cities. Compared to Figure 3.3.2, the two ACTS variants show a lower variability and more focused distributions. ACTS<sub>LOC</sub> and ACTS<sub>WOTC</sub> produce almost identical distributions, however, as will be shown later, their threshold assignments (and therefore the trajectory segmentations they imply) are actually different.

**Impact of acts strategies** In this section we provide a first evaluation of the impact of the selection strategies adopted in the two ACTS variants to assign threshold values to grid cells and, as effect, to individuals over those cells. Figure 3.3.10 shows the spatial distribution of the number of points (and, therefore, of pseudo-stop duration values associated) that fall in the cells of the two observed areas. In both cases, cells are obtained with a geohash precision  $h = 6$ . We notice that in the London dataset (right) the higher number of stops are mainly located along locations with high population density and within the urbanized areas. On the other hand, the Rome data (left) shows high values also along the main roads, and covers an area which is larger than the city itself (southern section of the picture), touching a part of Lazio (outside the city) and a part of Tuscany.

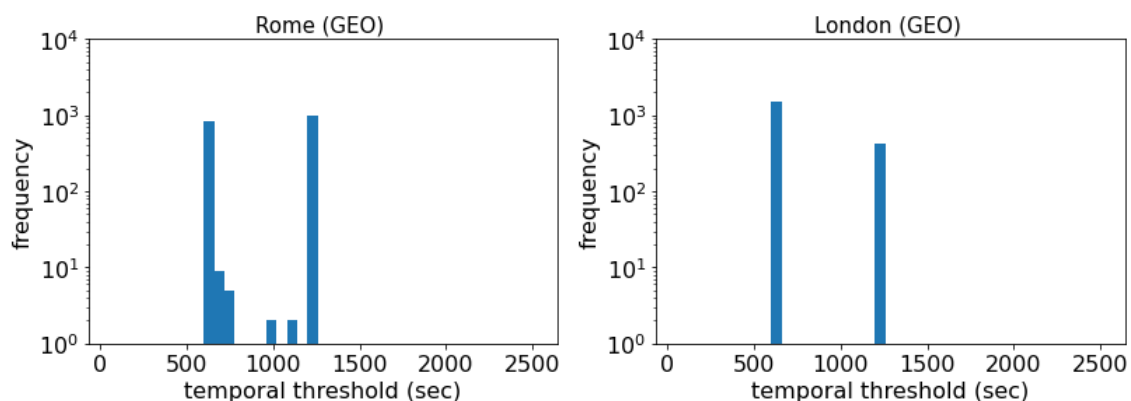


Figure 3.3.8: *Time threshold distributions for the users obtained by  $ACTS_{LOC}$  in Rome and London. Compared to  $ATS$ , the distributions are more concentrated on the two peaks.*

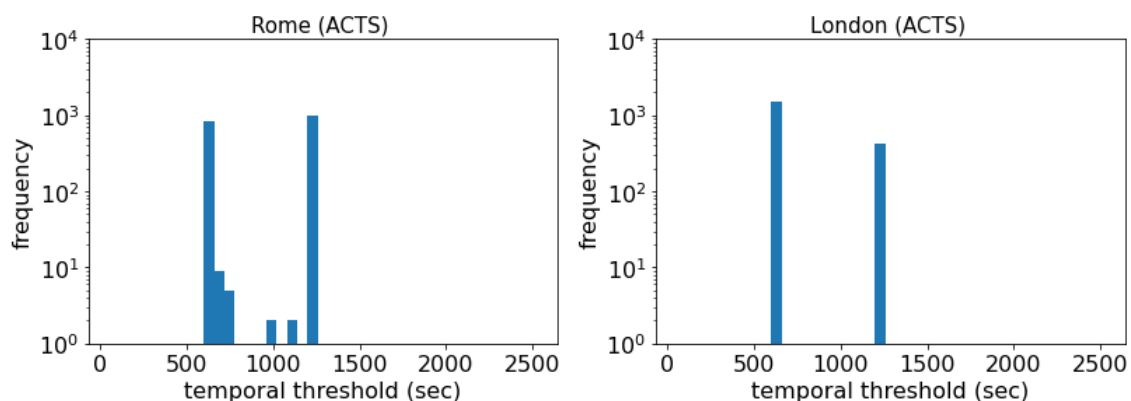


Figure 3.3.9: *Time threshold distributions for the users obtained by  $ACTS_{WOTC}$  in Rome and London. The overall distributions are very similar to  $ACTS_{LOC}$ .*

The plots in Figure 3.3.11 compare the temporal thresholds that  $ATS$  and  $ACTS_{WOTC}$  associate to each user for each cell they visit (remind that the value assigned by  $ATS$  will be the same for all the cells of a user, while  $ACTS_{WOTC}$  yields cell-dependent thresholds that will substitute the  $ATS$  value when the cell is poorly visited by the user). In both cities we can see that the differences, and thus the impact of  $ACTS_{WOTC}$  over  $ATS$ , is significant and approximately symmetric, i.e. sometimes the initial  $ATS$  threshold is increased, some other times it is decreased, with an overall balance between them. The corresponding plots for  $ACTS_{LOC}$  vs.  $ATS$  are very similar to the previous ones, and is therefore not reported here.

As mentioned above, the geohash precision is in principle a parameter that should be chosen by the user. In order to evaluate the sensitivity of the approach over such precision, we show in Figures 3.3.12 and 3.3.13 the same scatter plot replicated with

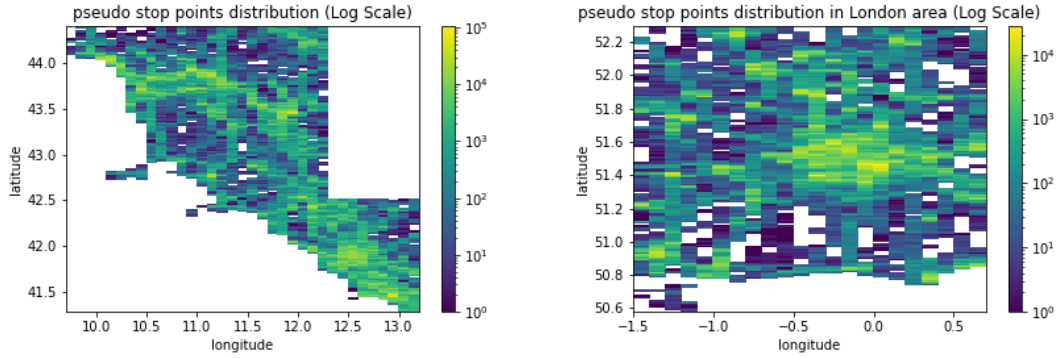


Figure 3.3.10: *Points distribution in Rome and London datasets over the geohash grid ( $h = 6$ ).*

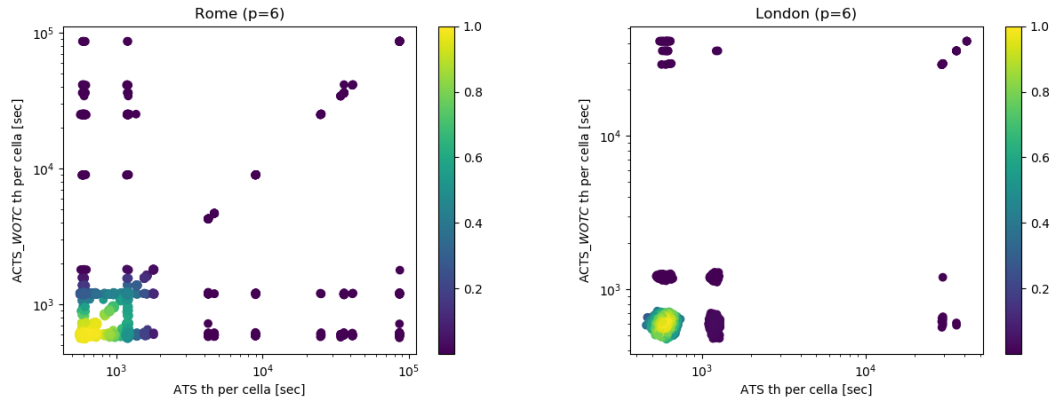


Figure 3.3.11: - *Comparison of  $ACTS_{WOTC}$  vs. ATS thresholds for all user-cell pairs, on Rome (left) and London (right). In both cases the difference appear significant and overall symmetric.*

precision  $h = 5$ , corresponding to cells of twice the area w.r.t. the previous case, and  $h = 7$ , corresponding to cells of half the original area. As we can see, the impact remains virtually the same as  $h = 6$ , suggesting that this is not a critical parameter – although values much smaller or much larger than these are expected to be not effective, since very small ones yield huge cells potentially covering entire cities, and very large ones create cells that are too small to capture significant amounts of points.

**Comparison of Evaluation Measures** In the following we first compare the ACTS methods against ATS, and later we compare their performances with those of the two variants of the competitor considered, namely HEH-D and HEH-O. All evaluations are based on the metrics defined earlier.

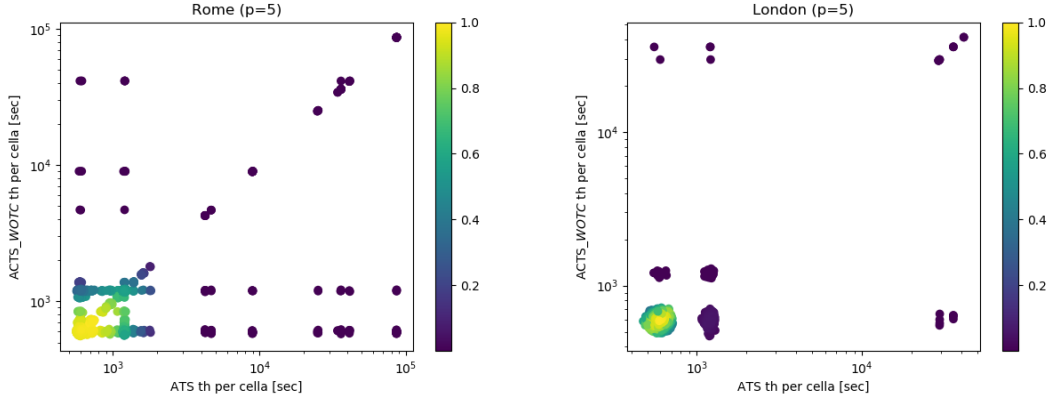


Figure 3.3.12: Comparison of  $ACTS_{WOTC}$  vs. ATS thresholds for all user-cell pairs, lowering the precision value ( $h = 5$ ), on Rome (left) and London (right). In both cases the difference appear significant and overall symmetric.

method	$MF_{25}$	$TP$	$DC$	$ratio_{sr}$	$\#segms (avg \pm std)$
ATS	.9513	.9507	.9876	0.0462	$851.551 \pm 717.173$
$ACTS_{LOC}$	.9587	.9654	.9174	0.0379	$946.743 \pm 785.998$
$ACTS_{WOTC}$	.9514	.951	.9856	0.0459	$857.157 \pm 713.349$
HEH-O	.1560	.1538	.7874	0.0313	$2400757.281 \pm 2760922.811$
HEH-D	.1877	.1308	.8586	0.0511	$2244814.994 \pm 3521517.705$

Table 3.3.3: Evaluation on Rome data. The first three columns show the measures illustrated in Section 10. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of segments.

Comparing our approaches, in Table 3.3.3 we observe that  $ACTS_{LOC}$  improves the performance of ATS in terms of  $MF_{25}$  and  $TP$  for Rome dataset. On the other hand, in Table 3.3.4 we can see that on the London dataset all three approaches are comparable in terms of performance, reaching higher levels of  $DC$  compared to Rome. In terms of sampling ratio (fourth column) the ACTS methods show an improvement against ATS, since their lower value (more pronounced for  $ACTS_{LOC}$ , and a bit marginal for  $ACTS_{WOTC}$ ) means that the former create trajectories with smaller internal time gaps. In addition, by analyzing the number of segments (last column) we can see that values for  $ACTS_{LOC}$  are higher than ATS and those for  $ACTS_{WOTC}$  are comparable, though slightly higher and with a slightly lower standard deviation. These factors, combined with the smaller  $ratio_{sr}$  of ACTS methods with respect to ATS, imply that overall the local and *wisdom of the crowd* mechanisms suggest more changes towards smaller thresholds, therefore leading to more splits.

Figure 3.3.14 reports the  $MF_{25}$  for all our approaches and the FTS baselines as boxplots. For the Rome case we can observe that the distribution of values of  $ACTS_{WOTC}$  is similar to ATS, only slightly more compact, while that of  $ACTS_{LOC}$  has

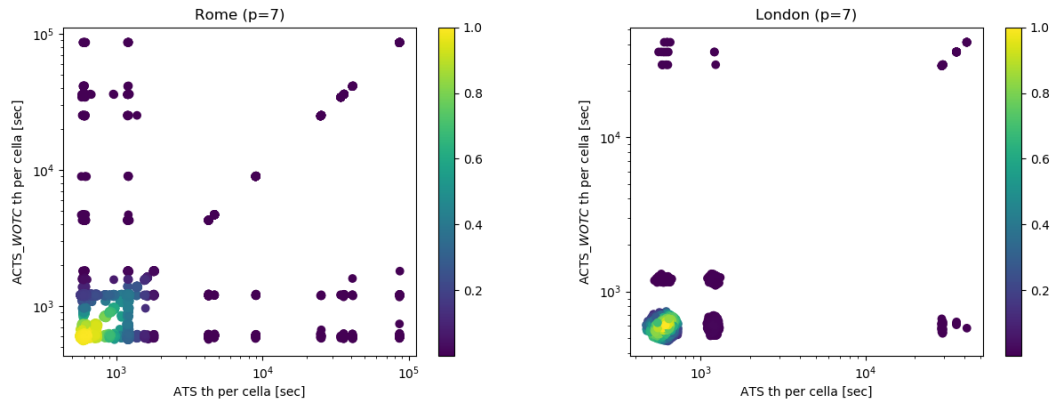


Figure 3.3.13: Comparison of  $ACTS_{WOTC}$  vs.  $ATS$  thresholds for all user-cell pairs, increasing the precision value ( $h = 7$ ), on Rome (left) and London (right). In both cases the difference appear significant and overall symmetric.

method	$MF_{25}$	$TP$	$DC$	$ratio_{sr}$	$\#segms$ (avg $\pm$ std)
ATS	.9547	.9523	.9991	0.0480	433.612 $\pm$ 525.916
$ACTS_{LOC}$	.9538	.9517	.9983	0.0472	477.971 $\pm$ 588.472
$ACTS_{WOTC}$	.9545	.9523	.9991	0.0478	433.652 $\pm$ 525.845
HEH-O	.3660	.3492	0.7513	0.0561	66389.061 $\pm$ 150547.085
HEH-D	.8877	.9140	.8401	0.0459	242882.126 $\pm$ 963253.651

Table 3.3.4: Evaluation on London data. The first three columns show the measures illustrated in Section 10. The fourth one reports the ratio between the average sampling period of non-stop points over that of all points, and the last column is the number of segments.

slightly higher median and a significantly smaller interquartile range. The differences in London are much less visible. In summary, the evaluation measures suggest that the ACTS methods achieve a small but interesting improvement over the basic ATS.

The last two lines of Tables 3.3.3 and 3.3.4 show the measures obtained with the two competitors. We can observe that there is a great discrepancy between them and those obtained with our methods, suggesting that, on our dataset, the clustering-based methods are not able to segment trajectories in an effective way. In particular, both HEH-D and HEH-O produce highly fragmented segments (see their huge number of segments yielded) leading to a medium-low distance coverage and a very low time precision – the only exception being HEH-D on London, which however further shows how its behaviour is unstable. Figure 3.3.15 reports the boxplots showing the distribution of  $MF_{.25}$  for the different approaches. We immediately notice that the scores obtained by HEH-O and HEH-D are significantly worse than the others. In light of the results obtained, we will not discuss these two competitors any further in this work, focusing instead on the behaviour of the other methods.

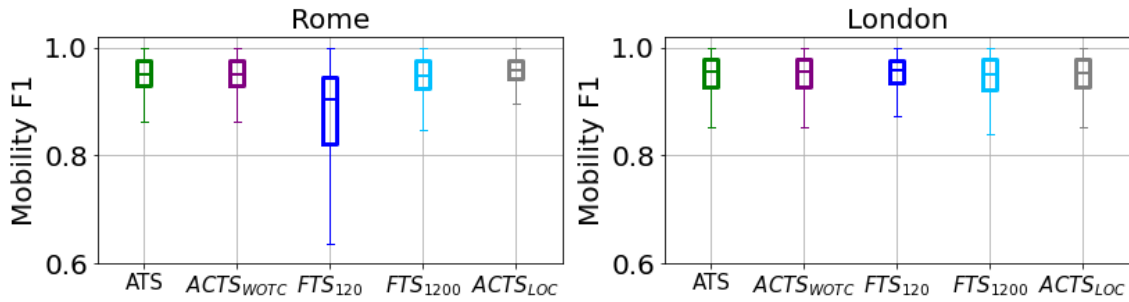


Figure 3.3.14: *Boxplots for  $MF_{.25}$ . On the Rome data  $ACTS_{WOTC}$  yields results similar to  $ATS$ , while  $ACTS_{LOC}$  significantly improves them. On London the differences are less pronounced.*

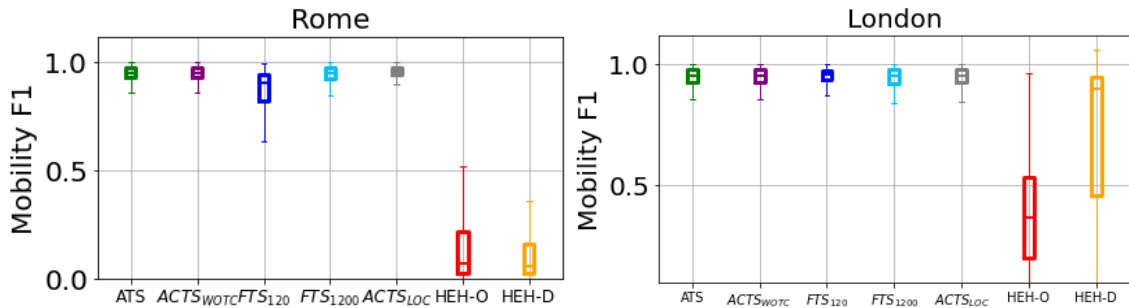


Figure 3.3.15: *Boxplots for  $MF_{.25}$ . In this case it is possible to see the comparison in terms of performance between our approaches and the  $DBSCAN$  and  $OPTICS$  cluster methods. In both cases the performance of the cluster methods are visibly worse than those achievable with  $ATS$  and  $ACTS$ .*

**Comparison of Segmentation Statistics** Similarly to what done in Section 3.3.4 for  $ATS$ , in the following we analyze other statistical indicators on the trajectory segments extracted by the  $ACTS$  methods. In Figure 3.3.16 we report the distributions of the average number of segments per user and points per segment for Rome (top) and London (bottom). The average number of segments per user (first column), highlights that in Rome  $ATS$  and  $ACTS_{WOTC}$  yield similar distributions, while  $ACTS_{LOC}$  generates more users with an high number of segments. In London the distribution is more skewed towards low numbers of segments, again with  $ACTS_{LOC}$  with a peak on higher values. In terms of number of points per segment (right column), we can see that in Rome most segments have between 5 and 15 points, yet  $ACTS_{LOC}$  shows a more concentrated peak on 5-10 points, which is coherent with the previous results (more segments are generated, and consequently they are shorter, on average). Something similar happens in London, now the concentration of values being between 15 and 30 points.



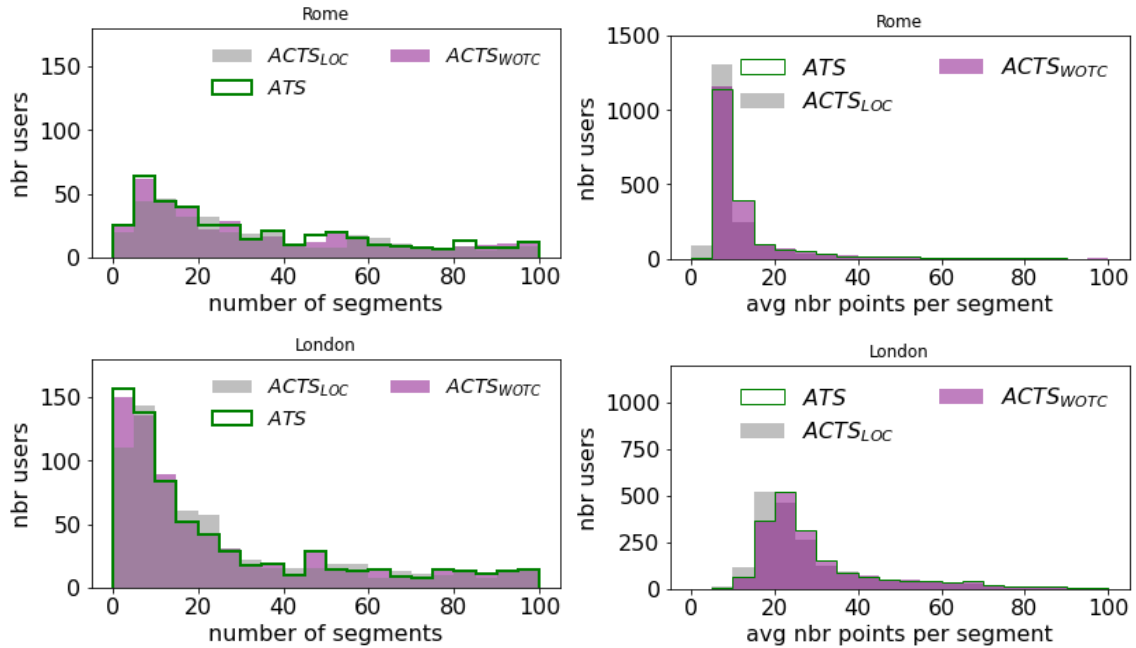


Figure 3.3.16: *Distribution of the number of segments, points and trajectories (from left to right) over Rome (top) and London (bottom).*

**Run times analysis** We present here some performance experiments regarding the scalability of our proposed methods w.r.t. the number of input trajectories (i.e. users) and their duration. In the first experiments, we test how the running time changes by varying the number of users in a range from 200 to 2000 (in steps of 200 and 250) while in the second ones we test it by varying the number of months covered by the data. In particular, in the last case we start from the data of a single month (January) and gradually add the next months one by one, obtaining 12 different datasets. These tests were made on the three ATS/ACTS approaches we proposed, compared with the two methods used as baseline (FTS and RTS). The experimental results for both Rome and London are shown in Figure 3.3.17. As it is possible to notice, the trends of FTS and RTS are linear and very low in both plots, confirming that their simplicity yields very fast executions. As expected, ATS and ACTS have much higher computation times, yet their trends appear to be linear or quasi-linear w.r.t. both the dimensions considered (users and duration), confirming the hypothesis made in Section 4.1.3.

### 3.3.5 Conclusions

In this work we have presented a set of user adaptive methods for solving the trajectory segmentation problem, a very common and useful task in mobility data mining, especially in preprocessing phases. The solutions proposed take into consideration the overall trajectory of the user, identifying an individual cut time threshold (each user can potentially have a different threshold) and also combining the information



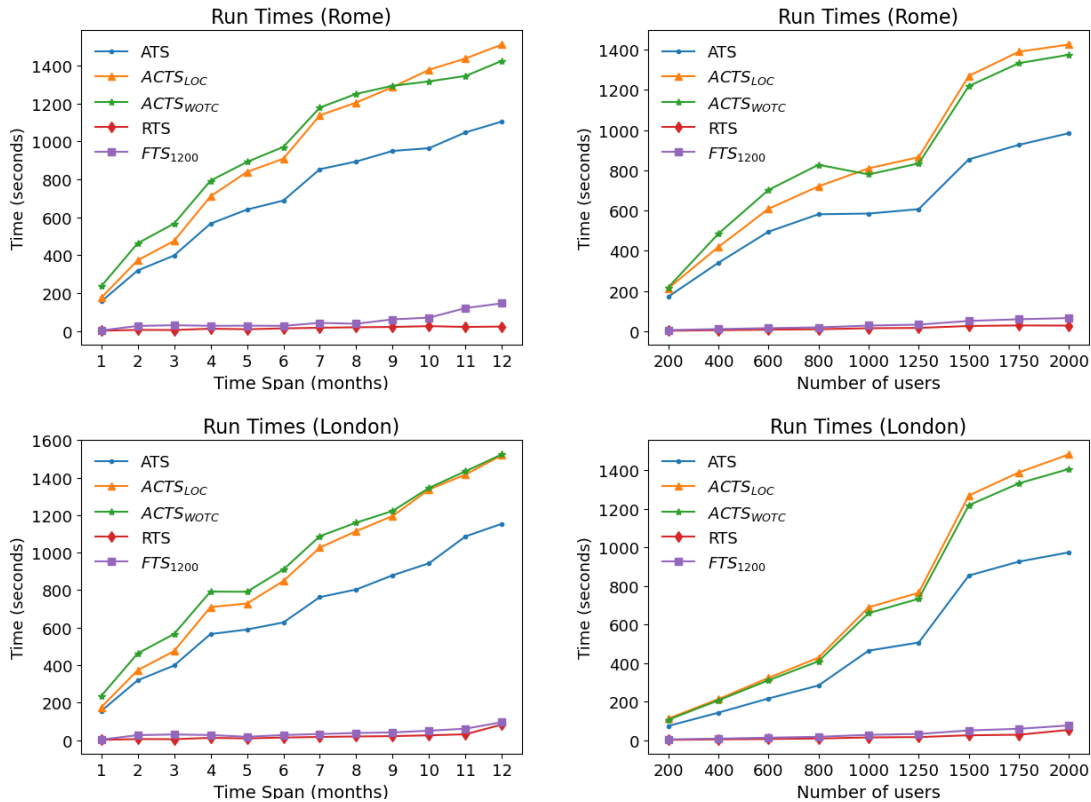


Figure 3.3.17: *Run times experiments in function of the number of users and the data collection period. In both cases the time trend grows quite linearly.*

coming from the different users through the spatial regions they share. This process yields thresholds for trajectory segmentation which are not only user-adaptive, but also location-adaptive, thus taking into account that a stop at different places might require time intervals of different duration to be considered a significant stay and thus a trajectory cut point. The experiments show that the individual and collective adaptive strategies have a significant impact on the thresholds obtained, which lead to a performance improvement in terms of the metrics defined for this purpose. Having a refined segmentation, as those obtained with ATS and the ACTS family, is very important in applications where the individual behaviour is under study.

### 3.4 On the Pursuit of Graph Embedding Strategies for Individual Mobility Networks

As we have seen previously, graphs are a universal language for describing and modeling complex systems. Network data are ubiquitous across different application fields, such as brain networks [330] in brain imaging, protein-protein interaction networks [214] in genetics, social networks [321] in social media and mobility network in mobility science as well. Unlike the regular grid-like Euclidean space data (e.g., images, audio and text), the graph has a nonlinear data structure so it can be used as an effective tool to describe and model the complex structure of network data. Modeling complex systems as graphs helps the characterization of very useful high-order geometric patterns for the networks, which has a great impact on improving the performance of different network data analysis tasks.

Luckily, graph embedding techniques have shown remarkable capacity of converting high-dimensional sparse graphs into low-dimensional, dense and continuous vector spaces where graph structure properties are maximally preserved[52]. The generated nonlinear graph embeddings in the latent space can be used to face different downstream graph analytic tasks (e.g., node classification, link prediction, community detection, visualization, etc.).

The main aim of graph embedding methods is to encode nodes into a latent vector space, i.e., pack every node's properties into a vector with a smaller dimension. Hence, node similarity in the original complex irregular spaces can be easily quantified based on various similarity measures (e.g., dot product and cosine distance) in the embedded vector spaces. Furthermore, the learned latent embeddings can greatly support much faster and more accurate graph analytics as opposed to directly performing such tasks in the high-dimensional complex graph domain.

In literature, many works analyze the mobility demand both from a lower level of trajectories, namely the single trips, and from a collective viewpoint, which aggregates the mobility of a population over a territory, aiming to identify traffic flows characteristics and predicting them [157, 158]. However, when significant data is available on single users or vehicles, a longitudinal analysis can be developed to build a model of the individual's mobility, which can help in several analytical tasks, from understanding mobility patterns [56] to mobility prediction [306], etc.

The mobility demand of an individual can be effectively represented by means of networks and graphs, where nodes correspond to spatial locations and edges represent trips connecting pairs of locations. In particular, we are interested in *Individual Mobility Networks* (IMNs). Some simplified examples are visually illustrated in Figure 3.4.2. By abstracting away the specific spatial position of each location, IMNs also provide a rather intuitive way to compare users belonging to different places, aiming to understand common and discriminating characteristics.

The question that arises, however, is now: *how to effectively (and efficiently) compare IMNs?* Which features, aspects of their structure or combinations of them are significant to characterize an IMN w.r.t. others? Very specific applications and tasks might directly come with an ad hoc answer, yet not providing a general

### 3.4. ON THE PURSUIT OF GRAPH EMBEDDING STRATEGIES FOR INDIVIDUAL MOBILITY

approach to the problem. Similarly, while humans can rationalize some discrimination criteria by visually inspecting IMN examples (e.g. by looking at Figure 3.4.2 one might think to measure the frequency dominance of the node labeled as '0', or to count the number of peripheral connections – i.e. not involving node '0' – in each graph), that might suffer from subjectivity of personal views, and miss some important aspects.

The scientific literature has tackled this type of problems for over a decade, the most relevant and interesting approaches belonging to the *graph embedding* field, i.e. the art of translating graphs into fixed-size vector representations that encapsulate the useful information implicitly contained in graphs. However, the landscape is rather complex, with each method proposing to model various kinds of concepts and undergoing validation on benchmarks that possess specific, often underexplored, characteristics. As a result, comprehending whether our graphs genuinely align with the expected requirements of the provided solution becomes very challenging. The purpose of this work, then, is to study what the current state-of-art in graph embedding can do for the specific case of IMNs, starting from a selection of candidate embedding methods; performing a comparative analysis of IMNs' characteristics to see where they are positioned w.r.t. some reference benchmarks in literature; and finishing with a mixed supervised and unsupervised experimentation aimed to identify promising existing solutions and lasting issues.

This work provides four types of contribution:

- 1) we introduce IMNs and their related embedding problem, characterizing them with respect to other popular graph embedding benchmarks, in particular identifying their key differences. This aspect is often not sufficiently discussed in literature, as the diversity of datasets employed is typically very limited – mostly belonging to social networks or molecule data sources – and the characteristics of datasets are mainly provided to show their size. Understanding how much the data we want to embed fit the validated benchmarks is an under-explored aspect of the problem;
- 2) we discuss several families of embedding methods, some of which are selected for experimental evaluation on IMNs. In particular, we highlight the concepts they want to model, the type of information they handle and how they design their aggregation / propagation within the graph;
- 3) we provide a detailed empirical study of various methods, aiming to understand which approaches seem to extract useful information for IMNs and also what is the intertwined role of different sets of input features and of inferred graph structure information. To this purpose, we design a supervised classification task to obtain objective performance measures, and complement it with a subjective evaluation based on similarity search and visual inspection;
- 4) finally, the work is intended to provide also a prototypical example of the process that analysts need to face when they have to orient in the vast and heterogeneous literature on graph embedding, since research papers often are mostly focused on highlighting the originality of the proposed solutions, while more general guidelines for practitioners are seldom discussed.

### 3.4.1 Graph Embedding: State of Art

Since our main objective is to study the applicability of graph embedding methods to Individual Mobility Networks, we provide here a brief overview of graph embedding literature. We remark that the expression *graph embedding* is often ambiguously adopted to refer both to the strategies that yield an embedding for the whole graph and those that instead assign an embedding to single nodes. We will refer to the first type as *graph-level embedding* and to the second as *node embedding*. These two categories are not mutually exclusive, since several graph-level embeddings are actually derived from a node embedding through aggregation of the output node representations, yet they might have different objectives to optimize.

#### Graph-level Embedding

Embedding of a whole graph is typically useful for querying or learning from several graphs of moderate size, where the final objective is to retrieve or classify/cluster graphs. The embedding can in principle be achieved by simply computing a fixed set of user-defined statistics and aggregates, for instance some common properties studied in network science: number of nodes, edges, closed triangles, the diameter of the graph, etc. However, that would not capture well the detailed structure of graphs. In particular, several methods try to consider properties of both the *local structure*, i.e. the connections between each node and its neighborhood, and the *global structure*, i.e. the connection/reachability among all nodes in the graph.

**Kernel-based Features** Kernel methods derive features for each node from a given neighborhood of other nodes, typically capturing the local properties of the graph around the node. Most solutions in this category are inspired to the basic Weisfeiler-Lehman (WL) algorithm [177], which starts with initial features associated to each node and iteratively updates them with an aggregation of the neighbors' features, repeating the process a given number of times, and finally aggregating the features of nodes into graph-level features, typically through histograms. Properly defining the neighborhood and how values are diffused is fundamental. The LDP method [51] adopts the simplest definition, and is limited to one-step neighbors, computing the distribution of their degrees. Various others follow a wavelet approach, where propagation is performed “jumping” from one node to the others located at exponentially increasing distances, thus capturing also long-range dependencies. For instance, Geometric Scattering [110] computes graph embeddings as statistics of the node values distributions, as obtained at each iteration of the wavelet propagation mechanism, while [319] examines the distributions of node features in subgraphs based on diffusion wavelets.

**Neural embeddings** This family of methods treats the neighborhood of a node as its *context*, following techniques like skip-grams inherited from language models. For instance, Graph2Vec [213] builds subgraphs for each node and applies a doc2vec approach [169] where the subgraph represents the document and the nodes' labels

the words in it. GL2Vec [64] further improves it by also exploiting labels on edges. We remark that these methods need node labels, which provide a common reference vocabulary among different graphs and make their embeddings comparable.

**Random Walk Approaches** These are basically a specific (and very popular) variant of kernel methods, where the neighborhood is identified through several random explorations of the links in the graph starting from the node and performing a given number of steps. Beside the neighborhood, the process assigns a weight or probability to the relation between the explored nodes and the starting one, which are used to propagate information or features values from the former to the latter. The Feather methods [261], for instance, use the weights as probabilities in computing the characteristic function of node features (basically, translating features into complex numbers and then computing their expected values). Finally, Anonymous Walk Embeddings (AWE) [150] renames nodes with their order of visit in the random walk, which highlights the presence of loops. The basic version of AWE, then, simply computes the frequency of each possible anonymous sequence, and associates the frequency distribution to the whole graph.

**Spectral Features** As for random walks, these methods consider transition probabilities between nodes (usually uniform, though external weights might be applied in some cases) and study the information diffusion among nodes, now adopting spectral analysis tools, such as the Laplacian matrix and its eigen-vectors and -values. For instance, NetLSD [308] and IGE [107] adopt a global *heat trace* signature of the graph derived from eigenvalues, achieving isomorphism invariance and adaptivity to scale. FGSD [312] defines a distance function between nodes based on their eigenvectors, computing the overall distribution of distances in the graph, with the possibility to emphasize more either local proximity or global structure.

### Node-level Embedding

Methods in this category aim to embed the individual nodes of a single, usually very large, graph. A graph-level embedding can be in principle derived quite easily by pooling the node embeddings in some way, e.g. through averaging or extracting other characteristics of value distributions. However, in order for it to make sense, the node embedding needs to be consistent across the different graphs we want to embed, so that their representations are comparable. Most works on node embedding do not discuss this aspect in detail, therefore it is often unclear if node aggregation is doable and therefore if the proposed methods also apply to graph-level.

**Proximity preservation** These methods aim to infer node embeddings that capture the relations between nodes and their neighbors, possibly at various distances.

Several approaches rely on neural embeddings (already mentioned above), such as DeepWalk [240] (and its improvement Walklets [241]), which applies the skip-gram approach by defining contexts through random walks. Node2vec [120] does

something similar, with various node sampling strategies to define neighborhoods, improved by Diff2vec [262] through the use of diffusion techniques. GraRep [54] also extends the concept of Deepwalk, considering  $k$ -step neighborhoods, and implementing it through Laplacian matrix factorization.

GLEE [303] applies Laplacian Eigenmaps to directly encode the graph structure (in particular, the proximity) through the geometry of the embedding space, yielding node embeddings from which we can estimate, for instance, the number of common neighbors of two nodes or of short paths.

Various papers study the problem in terms of the Laplacian matrix factorization, such as NetMF [249], which basically provides a reformulation of Deepwalk; BoostNE [178], that learns multiple graph representations of different granularity from coarse to fine; or the simple approach in [34], where the embedding is computed as the  $k$  lowest-frequency Laplacian eigenvectors.

Finally, HOPE [224] introduces the idea of assigning to each node two separate embeddings, namely a *source* and a *target* one, which fits the contexts with oriented graphs, also considering various node proximity measures to capture with the embeddings, such as Katz’s distance, Rooted Page Rank and others.

### Embedding Nodes with Attributes

Most of the works described so far look at the network properties of graphs, not considering additional existing features of its components. In particular, very frequently the nodes of the graph have either categorical features (e.g. the tags of posts in social networks) or numerical ones (frequency of visit of a web page, the length of a document, etc.) that could strongly help the embedding.

MUSAE [259] adopts a neural embedding on top of random walks (thus similar to Deepwalk), where the sequences of nodes (actually, node values, since they are attributed) are built with random walks sampling one node every  $r$  along the path. Different  $r$  values are used, pooling (AE) or simply joining (MUSAE) the corresponding results of each node into a larger embedding. Also inspired by Deepwalk, SINE [339] formulates a probabilistic learning framework that separately models pairs of node-context and node-attribute relationships, where each node learns its representation by considering context nodes and observable attributes of the node. BANE [333] adopts a WL diffusion (thus a kernel method) over the nodes attributes, and aims to obtain embeddings in a (binary) Hamming space. TENE [334] considers the case where nodes have a text annotation, and thus devise a matrix factorization schema to create separate embeddings for network structure and text similarity, later joined and optimized together. An analogous process is followed by FSCNMF [22], which outputs two regularized embeddings of the network corresponding to structure and content, later combined as the final representation.

### Graph Neural Networks

The most common neural network architectures adopted in the graph domain are Graph Convolution Networks (GCN), which work in a very similar way to the

Weisfeiler-Lehman (WL) algorithm, yet with learnable weights driving the aggregation steps. Being a supervised approach, it can be applied only when nodes are assigned to label values. An alternative approach is provided by *self-supervised learning* (SSL) approaches for graphs [186], which extract informative knowledge through ad hoc pretext tasks without relying on existing labels. SSL is often applied as pre-training of a model, and is then followed by a refinement step with conventional GCNs over a labeled dataset. Similarly, *encoder-decoder* methods (for instance [295]) learn an encoding function (which produces the embeddings) and a decoding one that minimizes a loss w.r.t. properties of the graph, typically the adjacency matrix representing the graph connections. ASNE [182] can be seen as a representative example, which separately embeds nodes and their features, then fuse them through NN layers used to estimate edge probabilities between node pairs.

### 3.4.2 Comparative study of IMN properties

As we mentioned in Section ?? IMNs are rather specific networks that show properties slightly different from other graph datasets typically used to validate graph-level embedding methods. Table 3.4.1 compares the averages of six statistics across six popular datasets plus IMNs. The values include size of the graphs ( $|V|$ ), diameter, density (fraction of edges over the theoretical maximum), nodes degree, entropy of node degrees and, finally, a statistics we called *maximal ego coverage*, computed as the fraction of nodes contained in the largest ego-network of the graph. We remark that, while IMNs are directed graphs, the other datasets considered are undirected. Thus, in order to have fair comparisons, for these measures also IMNs have been converted to their undirected version.

From this comparison, it emerges that while the average size and diameter of IMNs are similar to other graph datasets, their degree and density are rather low, thus showing a general sparseness of the graphs. At the same time, the entropy is relatively low and the maximal ego coverage is relatively high – much higher than what we would expect given IMNs’ low density. This suggests the presence of a few highly-connected nodes in IMNs counterbalanced by many low-degree ones. A clearer picture is given in Figure 3.4.1, showing boxplots for the more interesting measures. We can see that in IMNs the density is always very low, with a smaller inter-quartile range, while the maximal ego coverage is significantly higher than the others – excepted three (Deezer, Twitch and IMDB-M) that are however characterized by an extremely small diameter (exactly 2 for the ego-networks Deezer and Twitch, between 1 and 2 for IMDB-M), which makes high ego coverages easily very high. The Reddit graphs are those closer to IMNs characteristics, yet they are significantly smaller and denser.

**Sample IMNs.** The typical structure of IMNs can be seen in the small sample of graphs shown in Figure 3.4.2, where in addition to nodes and edges we also represent their frequency through size and thickness. As the statistics in the previous section suggested, IMNs are mostly characterized by a central, high-degree node (most likely

Dataset	avg  V	avg Diam.	avg Density	avg Deg	avg Deg. Entropy	avg Ego Cov.
DEEZER	23.49	2.00	0.23	5.55	1.64	0.94
TWITCH	29.67	2.00	0.20	5.84	1.93	0.96
GITHUB	113.79	5.86	0.08	4.12	1.51	0.56
Reddit	23.93	4.58	0.12	2.09	0.84	0.71
MUTAG	17.93	8.22	0.14	2.21	1.02	0.18
IMDB-M	13.00	1.47	0.77	10.14	0.48	0.90
IMN	33.96	3.68	0.12	3.26	1.34	0.78

Table 3.4.1: Comparison of statistics of various graph datasets.

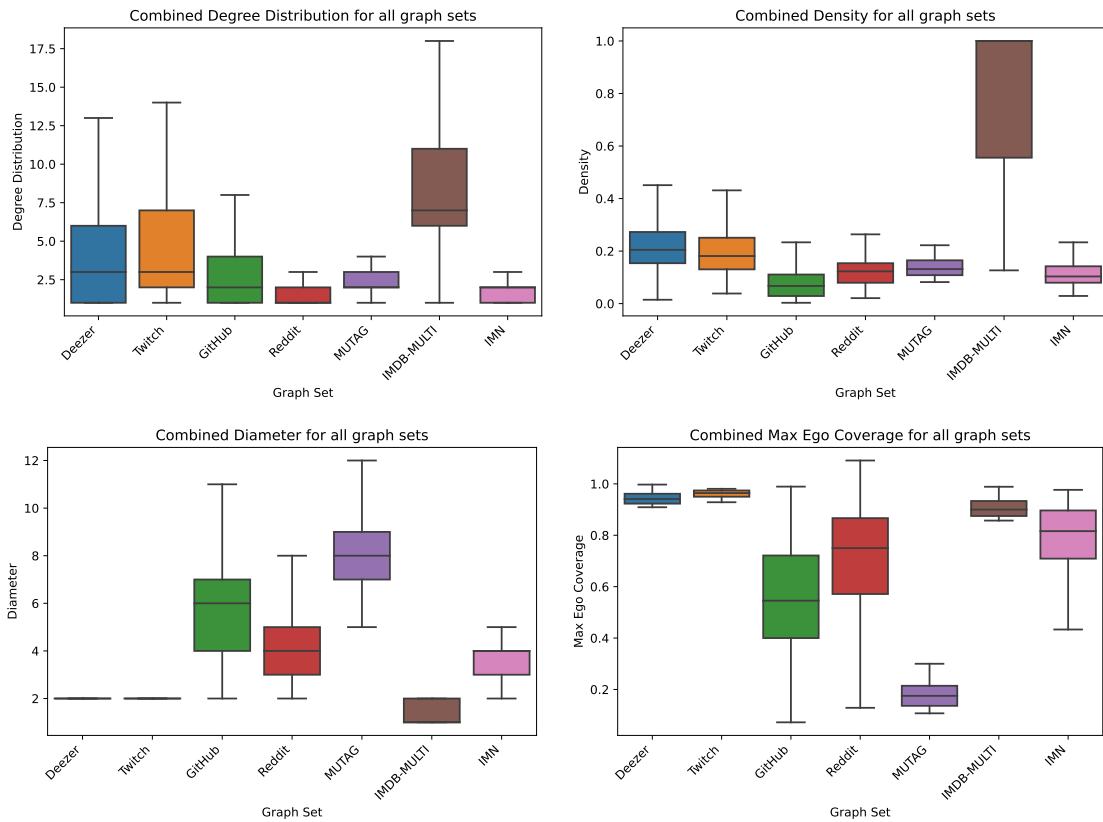
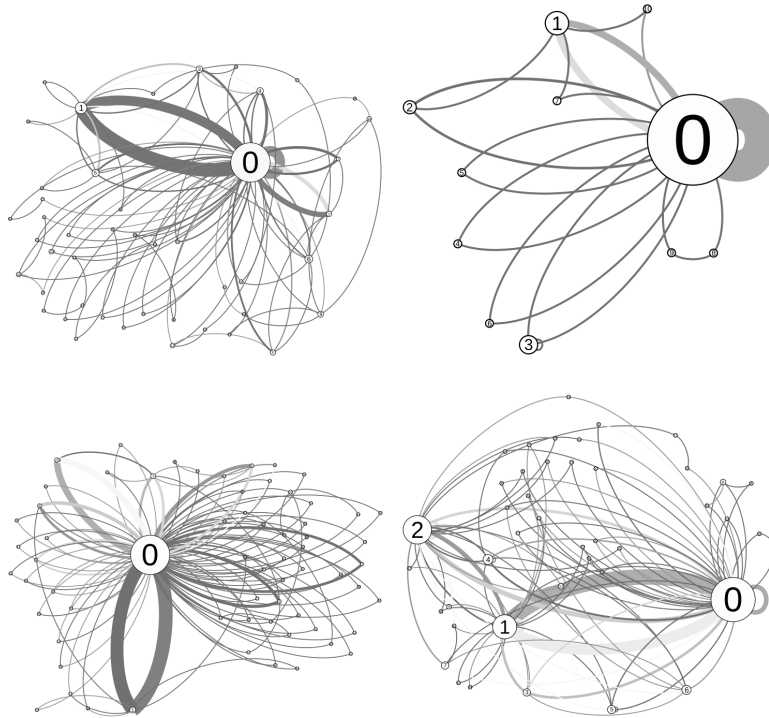


Figure 3.4.1: Distribution of nodes properties in various graph datasets.



Figure 3.4.2: *Sample IMNs.*

corresponding to the home location of users) connected to a large number of other nodes. Often this location is connected to a second one (usually the workplace) with high-frequency edges. The presence of these two strong *poles* drastically reduces the diameter of the graphs, although IMNs generally show a low density of connections.

### 3.4.3 Features, algorithms and validation task

In this section we introduce the three components of our experimentation: the (real) dataset used for evaluation; the embedding methods tested; the downstream tasks and their result evaluation.

#### IMNs and features

Our evaluation of embedding strategies is based on a dataset of 1000 IMNs, each corresponding to a private vehicle moving in the Tuscany region, Italy, over a period of 4 weeks. In particular, each vehicle belongs to one of five provinces, and the corresponding IMN was obtained following the process in [212]. The distribution of vehicles in the five provinces is perfectly balanced, i.e. 200 vehicles per province.

**Node features.** We enrich IMNs with four classes of features:

- Network Structure features: degree, clustering, node centrality, entropy of next location.

- Location usage features: stop frequency, stay duration, all the above stratified by time-of-day and day-of-week.
- Trajectory features: average length of incoming trips, average incoming trip duration, radius of gyration of the vehicle w.r.t. the node location.
- Geospatial features: Points of Interest (POIs). We remark that these features might be strongly location-specific, and thus directly help identifying places and their characteristics. For this reason, we will experiment both with and without this information source. Alternative information belonging to the same wide category might be integrated, such as distance from city center (or center of gravity, the home location, etc.), land usage, etc.

**Edge features.** This kind of information is generally not supported by graph embedding methods. One exception is GL2vec, that builds a dual graph where edges are represented as nodes, and thus their features are used in the embedding generation. For the others, some algorithms allow to use edge weights or probabilities, therefore some types of edge features might be translated into usable values. In our case, edges are associated with trip frequencies, thus we could in principle transform them into transition probabilities and use them in random walks methods, such as AWE and FEATHER-G, by modifying their usual transition choice mechanisms. However, preliminary tests with simple feature translations showed no improvement in the output quality, thus this option was discarded for the time being, and left as potential future direction to explore.

### Embedding algorithms

The embedding is performed adopting several different algorithms, chosen from the state-of-art described in Section 3.4.1 in order to cover the most important general approaches. The selection was also driven by the availability of open source code or libraries, and in most cases we exploited the Karate Club library [260]. We consider both graph-level methods, which thus directly return the embedding we needed; and node-level methods, where a final aggregation step is performed to obtain a graph representation. We briefly list them and provide details about the selected parameters and the adaptations performed (where needed).

**Graph-level methods** include the following:

- LDP[51], a simple **kernel method**. The only parameter is the number of bins, that we set to the default value (32);
- GeoScattering[110], another **kernel method**, requiring as parameters the order and moments adopted in creating the graph spectral descriptors, both set to the default value (4);
- graph2vec[213] is a **neural embeddings** approach. Also in this case, default parameters were used, in particular the output embedding size was set to 128;

### 3.4. ON THE PURSUIT OF GRAPH EMBEDDING STRATEGIES FOR INDIVIDUAL MOBILITY

- GL2Vec[64] is another **neural embeddings** method, similar to graph2vec, and the same parameters were set;
- FEATHER-G[261] has a **random walk** component and an aggregation of sampled values, which require to set, respectively, an order for adjacency matrix (set to default 5) and evaluation points (set to the default 25 values uniformly sampled from 0 to 2.5).
- AWE[150] is also a **random walks**-based method, whose main parameter is the length of generated walks. After testing values between 3 and 8, we chose 5 as the best one;
- GCN-self is a **self-supervised** learning approach [186], based on a standard Graph Convolutional Network to learn predicting a set of classes that is similar yet different from the downstream task. The output of the last convolution layer provides an embedding that is used as input for the final classification task. In our case, the training objective consists in predicting the radius of gyration (RoG) of the mobility of the user, discretized into 4 equal frequency intervals. We remark that a similar measure is provided at the level of single nodes, which however represents a more local view that might miss the overall picture, since the global RoG is computed w.r.t. the overall mobility center, which usually does not correspond to any user’s location.

**Node-level methods** adopted include:

- FEATHER-N is the node-level version of FEATHER-G, described above, thus basically a **random walks** method, the main difference being that it also allows to use **node features**, whereas FEATHER-G adopts a simple node degree. Parameters are set as above;
- MUSAE[259] is a **neural embeddings** method that also uses **node features**. The algorithm employs random walks, for which we set 5 walks per node of length 10, using order 3. Since it makes use of categorical node attributes, the original node features have been discretized into 5 equal-frequency bins.
- TENE[334] also uses **node attributes**, yet adopts a **matrix factorization** schema. Also in this case, categorical node features are used (mostly adopted to represent text labels), thus the same discretization as MUSAE was applied. The technical parameters of the algorithm were set to their default values.
- ASNE[182] is a **Graph Neural Network** (GNN) approach that, as MUSAE and TENE, exploits node labels. Also in this case, we translated original features to discretized values in the same way as above, and the parameters of the algorithm were set to their default values.

The final graph embeddings for node-level methods have been computed through a concatenation of average-, min- and max-pooling.

Among the categories of algorithms described in state-of-art, we are omitting *graph-level spectral features* methods, whose same ideas are basically implemented with different technical tools in other approaches; and *node-level proximity preserving* methods, since aligning the embeddings they produce across different graphs can be very problematic.

### Evaluation approach

We evaluate the quality of embeddings in two ways. First, in order to have objective performance measures, we define a downstream graph classification task, using as target label the reference province of each vehicle/graph. We remark that no node and edge feature adopted has a direct geographical reference, such as spatial coordinates, thus the task is indeed challenging. We test this approach by building a simple logistic classification model on top of the embeddings obtained with each of the embedding methods listed above, measuring the performance with the standard Area Under the ROC (AUC). Second, we simulate a nearest-neighbors query task, where a small number of IMNs are randomly sampled (the queries), and for each of them we select the 10 most similar IMNs in our dataset, based on the cosine similarity computed between the embeddings of the IMNs to compare. We perform this task only on a small selection of methods, and visually compare the returned set of IMNs. Clearly, the outcomes of this task are subjective, and different evaluators might draw different conclusions.

### Baseline and supervised approaches

In addition to the embedding methods listed above, we consider as baseline the usage of raw feature statistics, where, for each node feature, we compute the average value and standard deviation w.r.t. all nodes of a graph.

Finally, while we are not directly interested in supervised graph classification approaches, since the embedding they produce might be too specific for the prediction task, we tested two of them to provide reference performance values. Clearly, our expectation is that they can provide better predictions. The first method considered is a standard GCN (named GCN-PRV, since it is directly built on the target “province”), with three layers of convolution and a mean pooling before a final dense neural network to predict the province (five output neurons, hot-encoding the five provinces). The second method is a simpler dense neural network that takes as input the features of all nodes, stacked according to a node ordering based on nodes’ stop frequency (see Section 3.4.3), padding values for graphs with less nodes. The network, named DNN-sorted-nodes, is composed of three layers.

### 3.4.4 Empirical evaluation

In this section we summarize and comment the experimental results obtained in a downstream classification task, studying the impact of different sets of input fea-

### 3.4. ON THE PURSUIT OF GRAPH EMBEDDING STRATEGIES FOR INDIVIDUAL MOBILITY

tures, and then on an unsupervised nearest-neighbors retrieval task, where the results are visually evaluated through a subjective validation.

#### Classification with standard embedding methods

Table 3.4.2 reports a comparison of results obtained using the different embedding methods discussed above. On top of each embedding a classifier is built through a simple logistic regression. The choice of such simple classifier was intentional, in order to appreciate how well the embeddings could highlight the characteristics that can discriminate among the classes (in our case, the province where the user was moving). The performance measure adopted is the area under the ROC curve (AUC), and, as mentioned in Section 3.4.3, we report it separately on two columns, corresponding to the case where only non-geospatial features are available (“no POI”) and that where all features are used (“w/ POI”). For each column, we highlight the top three results (resp. bold, italic underlined, and just underlined). Embedding methods are grouped based on whether they exploit features or not. Finally, we report results for a self-supervised approach that trains the embedding for a different task (predict the radius of gyration of the user’s mobility), named GCN-RoG; and for the two supervised methods.

Table 3.4.2: *AUC performances of embedding methods on the province classification problem, based on a logistic classifier.*

	Method	AUC	
no features	LDP	0.4997	
	GeoScattering	0.5000	
	Graph2Vec	0.4992	
	GL2Vec	0.4837	
	FEATHER-G	0.5111	
	AWE	0.4693	
		w/out POI	with POI
features	raw features	<i><u>0.5603</u></i>	<b>0.7984</b>
	FEATHER-N	<b>0.5860</b>	<i><u>0.7759</u></i>
	MUSAE	0.5261	0.4940
	TENE	0.4928	0.5316
	ASNE	0.4904	0.5203
	GCN-RoG	<u>0.5558</u>	<u>0.6781</u>
	GCN-Sys	0.5297	0.5787
sup.	GCN-PRV	0.5241	0.9174
	DNN-sorted-nodes	0.6352	0.9388

First of all, results on the top group of the table clearly show that not using features yields very poor results, basically highlighting the fact that the plain struc-

ture information that such methods can derive does not capture the discriminating characteristics of the different areas. Their performance is significantly worse than using only aggregate raw features (see the first line of the “features” block).

Embedding methods that use all features excepted POIs are generally better, yet only Feather-N and self-supervised methods improve over using plain features, suggesting that the aggregation and diffusion processes implemented sometimes can destroy the information content in the original data. Apparently, the specific structure of IMNs makes most embedding methods add noise and hide the useful component of the initial node features. The self-supervised approaches perform relatively well, yet still below plain raw features, and significantly worse than Feather-N. Apparently, having a global objective to train for helps the emergence of generally useful characteristics, yet the task difference makes the final balance negative, as the information lost w.r.t. raw features is larger than that gained through the embeddings.

Examining the results using also POI information, we can see that the three top methods remain the same, yet yielding much higher AUC values. At the same time, not even Feather-N can improve over raw features. The other feature-based methods, instead, either do not improve significantly by adding POI information or even worsen slightly. In general, these results suggest that POI data provide a strong predictive power as they are, and any aggregation, propagation, or convolution simply loses useful information along the process.

### Impact of features

In this section we focus on the best performer among the unsupervised approaches (Feather-N) and investigate the role that different input features play. Figure 3.4.3 shows the AUC obtained selecting different subsets of the available features: network structure measures (NET), trajectory features (TRJ), location usage (USE). In the top plot geospatial features are excluded, while they are used in the bottom one.

Trajectory features taken alone are the best setting, while network structure and location usage have a much weaker impact. While the combination of NET and USE yields much better results than the two taken alone – thus, apparently, they complement well each other – combinations of TRJ with other features yield slightly worse results, suggesting that the added useful information is counterbalanced by the noise they introduce in the simple classification model adopted.

### Supervised approaches

**Graph convolutional network (GCN-PRV).** Following the standard GCN architecture, nodes are initially represented through their features, and then updated through three convolutional layers, i.e. for each node the mean of its neighbors’ features is computed and passed through a linear transformation, then through a non-linear function. The results become the new node representation, and the process is repeated (in our case) three times. The final node representations are then aggregated (in our case, through mean pooling) and fed to a final dense layer with one output node for each class.

### 3.4. ON THE PURSUIT OF GRAPH EMBEDDING STRATEGIES FOR INDIVIDUAL MOBILITY

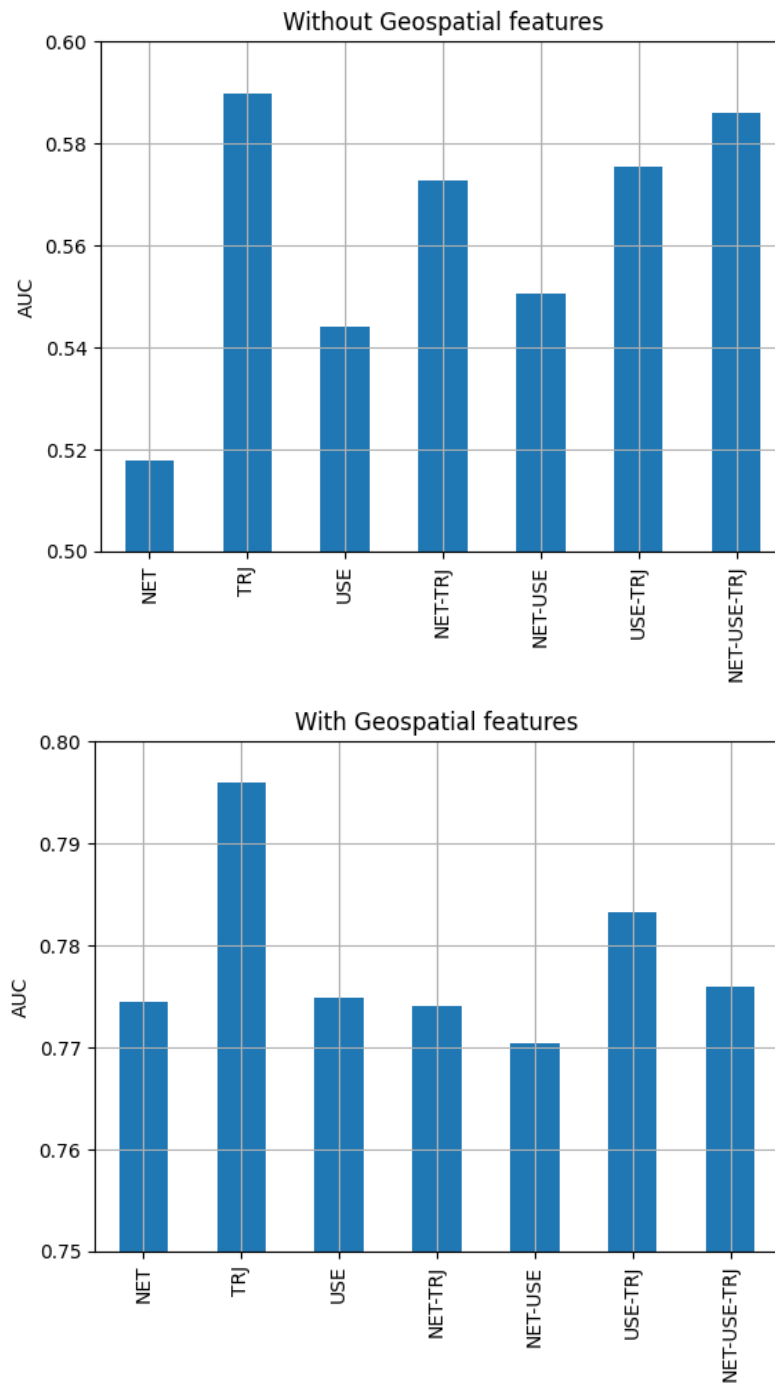


Figure 3.4.3: Impact of features: network measures (NET), mobility (TRJ), location usage (USE). Top plot: without geospatial features; bottom plot: with geospatial features.

The results obtained for GCN-PRV are rather heterogeneous, since with no POI data the performances are slightly worse than the self-supervised approach, while with POI data the AUC increases beyond 90%, largely improving over embedding and raw feature approaches. This suggests that, due to IMNs structure, the GCN mechanism is good at identifying and emphasizing very important basic features (as in the case of POIs) but not to infer useful information from the graph structure and other weaker features.

**Fully connected layers (DNN-sorted-nodes).** This model is a graph-level classification approach utilizing fully connected layers (Dense layers) and a pooling operation. The preprocessing stage involves padding the node features of each graph to ensure consistent input dimensions. The model’s architecture comprises three hidden dense layers, each followed by a dropout layer to prevent overfitting. The final dense layer maps the graph-level features to the desired output dimension, while a softmax activation function is used to derive the predicted class probabilities.

During training, the model optimizes its performance using the Sparse Categorical Crossentropy loss function. The training is conducted on batches of graphs, with the Adam optimizer adjusting the neural network’s weights based on loss and accuracy metrics. Subsequently, the graph-level embeddings are obtained, resulting in final embedding vectors of size equal to the number of classes. This is achieved by predicting the class probabilities for each graph, providing a comprehensive representation of the graphs in the dataset.

The results in Table 3.4.2 show that this less refined approach is actually more robust than the GCN one, improving over all the other methods, also when no POI data is involved. This might suggest that nodes’ ids are important to learn how to treat their features, maybe highlighting more central nodes (home and work locations) and de-emphasizing the others. In order to inspect this hypothesis, Table 3.4.3 shows what happens when we shuffle the order of nodes, independently on each graph, thus losing their identity. However, the results show that the variation is surprisingly small (always less than 0.0105) and practically insignificant, both with and without POI data. This suggests that the most important factor in this prediction task is how different features are combined, and not the node they come from. That is similar to what raw features achieve, in addition to the non-linear combination of features provided by the dense layers that helps to better identify the most discriminant information.

Table 3.4.3: *AUC of Fully Connected architecture*

	Shuffled nodes	AUC (w/o POI)	AUC (w/ POI)
PRV	None	0.6748	0.9682
	All - { Home, Work }	0.6727	0.9671
	All - { Home }	0.6661	0.9677
	All	0.6644	0.9577



While the performances tend to decrease the more we shuffle nodes, the variation is surprisingly small and practically insignificant, both with and without POI data.

### kNN Query Evaluation

In this section, we use the embeddings obtained above to perform a kNN queries over a small set of IMNs and then visually review the results. We will present here the results obtained on two query IMNs, showing the top 10 most similar IMNs returned. For each query IMN, we perform the task both using the embeddings obtained with Feather-N (the best performer in the classification task) and with the self-supervised GCN-RoG, comparing the results. The similarity between two embeddings  $e_1$  and  $e_2$  is computed using the *cosine similarity*  $s(e_1, e_2) = \langle e_1, e_2 \rangle / \|e_1\| \cdot \|e_2\|$ .

An overall comparison of results showed that the two embeddings generally return very different sets of outputs, on average having an overlap smaller than 10%. In the following, we consider a query IMN where the overlap is relatively high (two IMNs over ten) and then one where the overlap is nil.

Figure 3.4.4 shows the first query IMN (on top) and the 10 top results returned, ranked by similarity. For the sake of readability, the graphs report the frequency of edges (numerically and as thickness) while nodes only show the frequency rank ('0' is the most frequent one, and so on). The overlaps between the two answer sets are highlighted. A comparison of results shows that Feather-N returns IMNs of similar size and complexity of the query. Also, the query contains a central *open triangle*, namely the most frequent node is strongly connected to the second and third most frequent ones, which are not (significantly) mutually connected. This same pattern seems to emerge in most of the Feather-N answers. On the contrary, GCN-RoG output seems to have fewer open triangles and also tends to include more crowded IMNs and star-like shapes than the query. From this perspective, Feather-N seems to return intuitively better answers.

Figure 3.4.5 presents the results with the second query, which show similar characteristics to the previous case: GCN-RoG tends to include much larger and densely connected answers, where the central node is more predominant than in the query, whereas Feather-N's appear more balanced, except very few cases presenting a star-like shape.

### 3.4.5 Summary and Conclusions

Our exploration started from realizing how much our data, namely IMNs, are different from typical benchmarks used in the graph embedding literature: a semantic difference, since most benchmarks deal with human interactions data or physically connected elements of molecules, whereas IMNs are about movements, thus making recurring concepts in embedding literature like information propagation and bindings not perfectly fit; and a statistical difference, since the empirical exploration of IMNs' properties resulted to be different from many of the others.

The review of existing graph embedding methods highlighted the existence of a limited set of fundamental approaches, plus several variants and improvements.

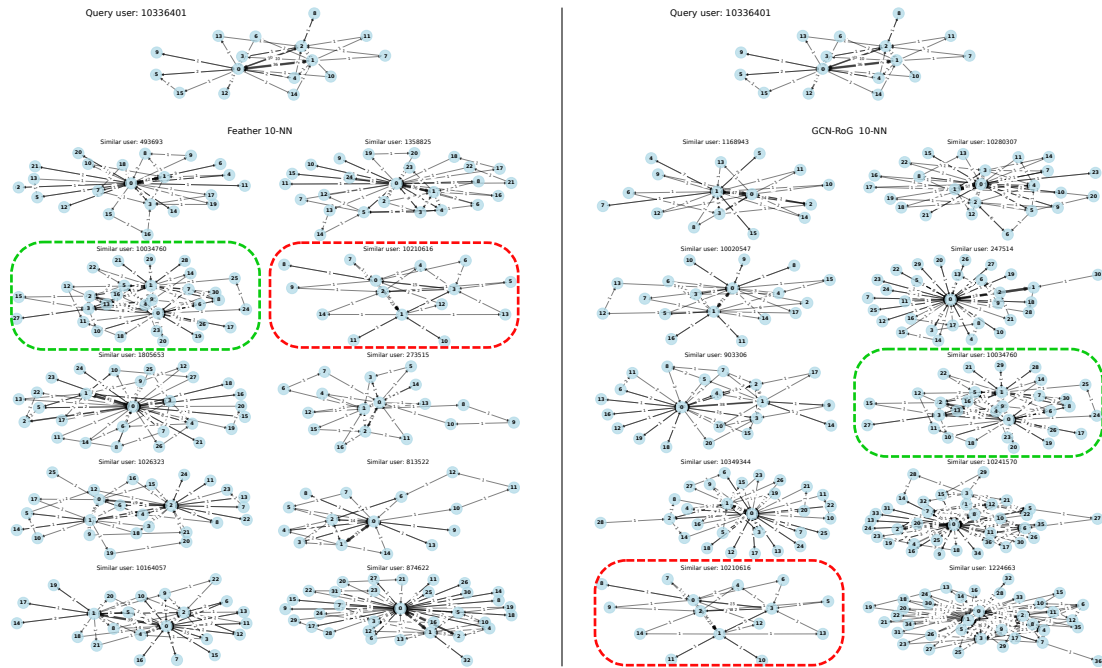


Figure 3.4.4: Query test 1: Top 10 most similar IMNs to a sample user, using Feather-N (left) and GCN-RoG (right) embeddings.

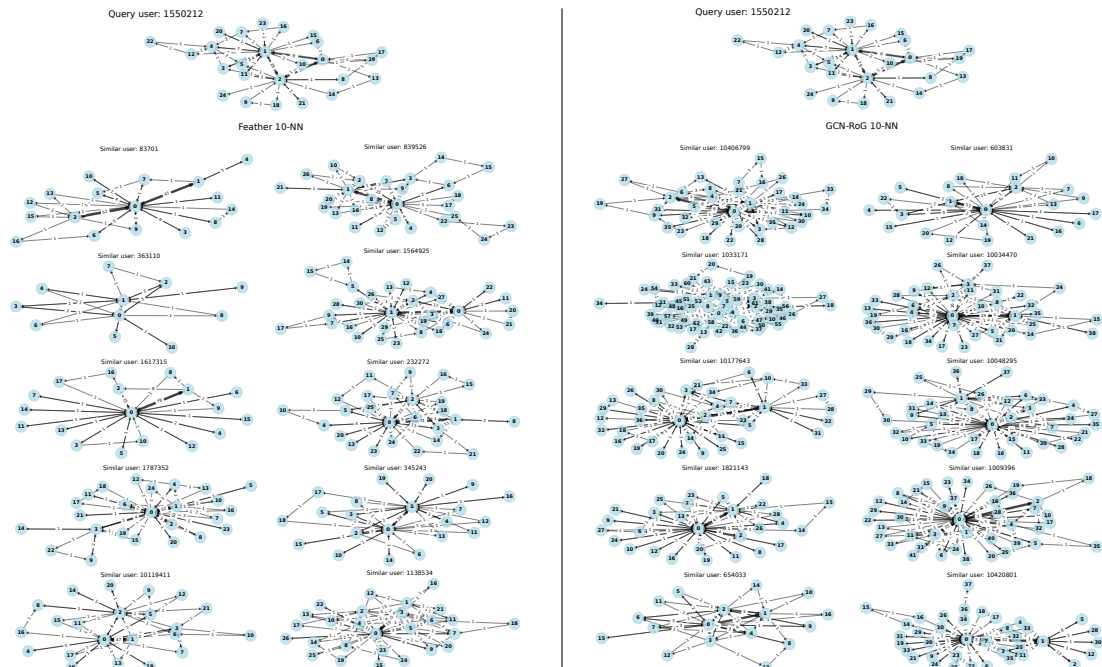


Figure 3.4.5: Query test 2: Top 10 most similar IMNs to a sample user, using Feather-N (left) and GCN-RoG (right) embeddings.

### 3.4. ON THE PURSUIT OF GRAPH EMBEDDING STRATEGIES FOR INDIVIDUAL MOBILITY

Many of them are general purpose, yet their validity is typically assessed on prediction tasks (which are not our primary objective) over a limited set of application domains. That makes it difficult to identify the promising methods to choose, for which reason we selected approaches from the most representative algorithms in the literature that could be applied to graph-level embedding. This lack of theoretical foundations in the selection of an algorithm is an important gap that can only partially be filled by purely empirical tests.

Empirical evaluations highlighted how the (rich) node features of IMNs are fundamental to achieving acceptable results, yet also suggesting that most methods are not able to handle them properly. Among the methods tested, we could identify only one – a simple adaptation of Feather-N to work at the graph level – that seems to perform better than no-embedding input features, and yet with no large margins. We performed a simple visual check of the embeddings produced through kNN queries by looking at the exterior graph properties (nodes, connections, and weights), which showed results close to what we expected intuitively.

Overall, the claim of this work is that current graph embedding literature provides several appealing approaches and, yet, very poor support to the development of applications, leaving the analyst alone in orienting within the vast literature, guided almost exclusively by empirical explorations. Our exploration aims to provide a first example of how to develop an analytical process in this context.

The natural evolution of this work goes in two directions. The first one is specific for IMNs and includes the development of embedding methods ad hoc for them, aimed to better exploit the abundant features at node and edge level and to cope with their particular graph structure. The second one is more theoretical and aims to classify existing methods based on what kind of information is actually built from the input graph structure and features, abstracting away from formalization and computational aspects, which often cover very similar concepts behind different covers.



## Act II: Individual Mobility Models at Work

Realistic human mobility models have great potential benefits to societies. Traces generated by human mobility models can be used in epidemics, urban planning, transportation systems, and disaster response. The first step to apply mobility models to realistic applications is always to have a strong mathematical abstraction of human behavior. Among many mobility solutions, the Individual Mobility Network (IMNs), described in Sec.??, fully meet this request. As we have already seen, a IMN of a user is a directed graph defined by a set of nodes and a set of edges. Nodes represent locations and edges represent movements between locations. Having this kind of abstract tool allow us to exploit it in specific application, too. In this chapter we present some examples of IMNs employment for specific task. First of all we want to introduce an analysis of the potential risks involved in using mobility data without the right attention. Even if it looks as anonymized data, it is easy to extract sensitive information about individuals from them. Taking this into account, the following sections will present esamples of IMNs adoption for two specific tasks: electrica vehicles simulations and car crash prediction. In both IMNs play a key role in achieving the aiming results .

### 4.1 Electrical Vehicles

Electric mobility is one of the main advocated solutions for making urban environments ecologically more sustainable, improving the quality of life of citizens [196]. Despite the quick development of the Electric Vehicle (EV) market and the strong commitment of car makers, various social barriers need still to be overcome to complete the transition of mobility towards electric [143]. Indeed, most users are very little familiar with what driving an EV really means and what it might change in their daily life if they replace their fuel-based vehicle with and electric one. This lack of knowledge causes several worries to the average potential user, even though its many advantages for the environment are clear.

**What makes EV mobility different.** From the viewpoint of the end user, one of the biggest differences between a fuel-powered vehicle and a battery-powered vehicle lies in the reduced autonomy: while high-profile EV models have performances similar to fossil-fuel cars, average EVs have a range in the order of 200 km, which makes the need for recharging more frequent. This fact can often induce in the user the so called “range anxiety” [100], which might be reduced by gaining experience on range management and building trust in range estimation systems. In addition, the time required to fill a fuel tank is usually just a few minutes, while a stop to recharge the battery of an EV can take much longer times, up to some hours, depending on the capacity of the battery and the type of recharger [286]. This requires a more careful planning of trips and recharges. Finally, in most countries the recharge infrastructures are currently much less developed than fossil fuel ones, thus arising further concerns about the capability for a user to satisfy their mobility needs without introducing significant deviations from original travel plans. On the positive side, different from conventional fuel, electric energy is an utility available in any building, and in several cases that makes EV recharging possible at home or at workplaces.

The objective of our study is to propose a process that, through a mix of mobility data analytics, ad hoc trip planning and simulation, is able to analyze the current fuel-based mobility of a user and quantitatively describe the impact of switching to EVs on their mobility lifestyle. We emphasize that our aim is to reproduce the study over large sets of users and long periods of time, thus the process needs to be scalable and completely automatic. As we aim to reproduce the study over large sets of users and long periods of time, the process needs to be scalable and completely automatic. The final result is not only a set of general indicators over the whole population under study, but also insights about how the switch to EVs affected the mobility of single users. Our solution adds a novel perspective to existing literature on the topic, the latter being mostly focused on the infrastructural issues, namely how to organize the energy distribution (e.g. where to place recharge stations) and how it will impact the current power grid, or on abstract path optimization problems, trying to minimize the battery consumption of trips or the overall time. Only a small portion of works try to quantitatively study how much the current mobility actually fits the constraints imposed by EVs, and in most cases that is done at a general level, e.g. counting the trips that might fit a given range [219] (with or without recharges) or studying the mobility characteristics of a territory to evaluate the sustainability of EV usage [82].

**The simulation framework.** In order to achieve our objectives, we developed several components. First, we set up an enriched road network by integrating the basic OpenStreetMap network with elevation information and estimate battery consumption for each road segment, obtained through a mathematical estimation model, and with the availability of recharge stations at each node. Then, we developed a simulation framework that takes as input the sequence of trips (origins and destinations) performed by a single user, and returns a simulated travel plan that

mimicks the original one adapting it to EV requirements. The framework has two main components. The first one is a fast heuristics for determining the best path to reach a destination starting with a given battery level, also performing intermediate stops (and associated deviation from the path) to recharge, when needed. The second one is a set of mechanisms for simulating passive recharges (i.e. performed while the vehicle was parked at a destination) at key places, such as the individual home or work, or nearby recharge stations. The concepts of home and work are automatically inferred from the input sequence of trips, based on the Individual Mobility Networks [257] (IMN).

**Brief summary of results.** The framework was used to analyze a large pool of users over a significant period of time, providing a deep analysis of results both at the collective level and at the individual one. Various scenarios were considered, depending on if recharging at home and/or at work is allowed. The results suggest that the impact of trip deviations and recharge stops are generally negligible compared to the original mobility, with an increase of time spent not larger than 1.23% and an increase of distances travelled less than 1% in the worst scenarios (recharges are possible only at public recharge stations), which significantly drop when recharging at home and work is possible. Also, emergency situations where no suitable path can be discovered are rather rare. Also, worst case individuals suffer from increases that do not exceed 4% both in terms of distance and time. Finally, through an IMN-based summary visualization of the history of a sample of individual users, it was possible to appreciate how the recharges are distributed on the different trips, and how recharging at home and work keeps the battery level always rather high (which is good to reduce “range anxiety” issues) and removes several recharges during trips in favour of frequent short recharges at home/work. In summary, the novel contributions of this work are the following:

- we develop a simulation framework for EVs based on a set of (real) individual trips, that mirrors them according to EV constraints and battery recharge opportunities, including availability of recharge at home and at work;
- we define and implement a fast heuristics to compute the best path from an origin to a destination, taking into account the battery constraints and, where needed, computing a deviation to reach a recharge station;
- we perform an experimentation over a large dataset of real users in the Tuscany region, Italy, analyzing the impact of EVs on their mobility both at a collective and at the individual level;
- finally, we select a sample of users and explore the impact of EVs on their mobility through a network (IMN) representation.

### 4.1.1 Related Works

Electric vehicles (EVs) are experiencing a rise in popularity over the past few years as the technology has matured and costs have declined, and support for clean trans-

portation has promoted awareness, increased charging opportunities, and facilitated EV adoption. Suitably, a vast body of literature has been produced exploring various facets of EVs and their role in transportation and energy systems.

### **Studies on issues and opportunities of EVs**

**Market and stakeholders studies.** Several papers and reports perform surveys to see how EVs fit individual needs [171, 41, 136], mostly capturing the feelings of people or general statistics, thus not providing ways to profile the individual electricifiability of users in objective terms. Vehicle price, fuel cost, driving range, battery replacement cost, charging time and maintenance cost are among the significant attributes considered in consumer choice modelling [86]. Range anxiety, refueling availability, and vehicle efficiency influence EVs' purchase. On the same direction, [70] propose psychologically founded methods of assessing the acceptance of EVs in everyday use, identifying four key elements for user evaluation of EVs: mobility aspects, human-machine interaction (HMI), traffic and safety implications and acceptance. Results show that a great part of daily mobility needs are satisfied, and ecological aspects play an important role.

### **Simulation-based EV studies.**

The work in [219] investigated the charging behaviour of EV drivers by simulating EVs travelling and charging at public chargers. The results show that more than 5% of the trips would require recharging at a public charger for different driving range and charging assumptions. The location of the charging stations is directly related to the impact of driving behaviour in urban road transport networks. In [111] a general corridor model is used to propose the optimal location of EV charging stations, while the authors of [179] propose a multi-period optimization model to expand the charging network. Similar studies for an urban environment can be found in [138] and [57]. A spatial-temporal demand coverage location approach is used in [309] to address the location problem of electric taxi charging stations. Some traffic simulator platforms offer EV support and give to the user the ability to run traffic simulations with all or partially electrified vehicle fleets. Such a simulation can be found in [74], where EVs are simulated in highway networks with on-line charging. Another example is found in [21], where a spatial-temporal model is build based on a Poisson arrival location model (PALM) for EVs charging at public stations on the highway. Our approach extends the aim of previous works by providing a framework to derive quantitative feedbacks on potential issues introduced by EVs for individual users, and their impact on users' mobility, exploiting large scale mobility traces.

**EVs impact on the grid.** The charging behaviour of EV drivers at public and domestic chargers affects the electricity grid. The charging method is of particular interest, as it defines the magnitude of this impact [69]. In [273], the authors evaluate different charging strategies through performance indicators, and show that three characteristics are essential to develop a sustainable charging strategy: the maximum charging power, the duration of a full recharge, and the shape of the charging curve.



All three characteristics impact directly the electricity grid. Most works predict the electricity demand from the EVs charging without considering historical data. They are mainly based on the traffic patterns, the charging and the battery characteristics of the EVs. However, in [39],[155] the concept of forecasting the charging demand is introduced. In [39], a model for Short-Term Load Forecasting for EV charging was implemented using neural network. In [331] an artificial intelligence EV load forecasting technique is introduced using Support Vector Machine. Charging events for one year were created using national statistical data, to cover the lack of real historical data of EV's charging events. After the model was trained, the SVM model provided a forecast for the day-ahead EV demand. Another perspective to the problem is given in [89], which analyzes the use and usability problems that are faced by EV drivers who rely on workplace recharge facilities, providing a case study in the UK. Finally, the work in [243] studies the impact on the grid from the viewpoint of commercial fleets, through an ad hoc data collection and considering various power sources, such as the public grid and renewable sources.

### Simulating the EV mobility

In order to simulate the mobility EV vehicles in a realistic and effective way, various factors should be taken into consideration.

**Route planning.** A key part of simulation is the planning of single trips, namely the most likely path that a user/vehicle will follow to move from an origin location to a destination. There are different approaches to route planning focused on EVs: the challenge of finding the optimal solution to save energy consumption, minimize wasted time and travel the shortest feasible path is computationally hard, thus requiring a strong effort from the transportation engineering community. A detailed overview about recent advances in algorithms for route planning in transportation networks can be found in [26]. Classic route planning approaches apply Dijkstra's algorithm to a graph representation of the mobility network [83]. Speedup techniques have been also proposed, with different benefits in terms of preprocessing time and space, query speed, and simplicity. An optimal solution to the Constrained Shortest Path problem is proposed in [300]: authors propose a general labelling algorithm to take into account a set of constraints while searching for the shortest path. Such an approach relies on Dynamic programming, implementing a bidirectional strategy to optimize the search process. Constrained Shortest Path (CSP) formulations try to find the most energy efficient route without exceeding a certain driving time or finding the fastest route that does not violate battery constraints [135]. For example in [29] authors extend this problem respecting battery constraints, minimize overall trip time, including time spent at the charging stations. The solutions proposed include all types of stations: battery swapping stations, regular charging stations with various charging powers and superchargers. In [288] the problem of finding an optimal routing and recharging policy is explored using a grid network with uncertain charging station availability. Similarly, one can consider some trade-offs between driving time and energy consumption: in [30] a set of routes for EVs mobility is computed by Pareto optimisation considering how to save energy driving

road segments at different speeds. In literature many works focus their attention in finding the most energy efficient path for battery powered electric cars [19]. In [289] the problem of finding a minimum cost path when the vehicle must recharge along the way is modeled as a dynamic problem. For electric vehicles, most papers have focused the attention on the integration of battery capacity constraints and negative edge weights (a result of recuperation) into classical single-criterion route planning algorithms optimizing energy consumption [266, 31]. However, such paths may have disproportionate detours: driving slower saves energy at the cost of greatly longer travel time. Finally, a few works introduce heuristics in the shortest-path and trip planning problem, e.g. [284], where the authors designed an approximation scheme to compute the energy efficient shortest route for EV drivers. In our work we propose a greedy routing strategy that favours travel times (expected to be the priority criterion for average users) and plans recharge stops at stations in such a way to minimize the combined cost of deviating the path and the recharge time.

**Global optimization of EV mobility.** The several trips that make up a user’s daily mobility clearly influence each other, as battery recharges might be anticipated or postponed in various ways, in principle calling for an overall optimization based on the knowledge of a long-term plan of movements. State-of-the-art routing software provides a set of tools based on operational research algorithms to solve Constrained Vehicle Routing problem. One of the most popular tool is OR-Tools [242]. However, such tools do not provide the possibility to analyze and solve the problem with the level of details we need: as an instance, Vehicle Routing algorithm in Or-tools requires to pre-compute the shortest path among all the possible destinations. It would not be possible to evaluate the routing of every new user/vehicle from our dataset without a huge overhead in terms of data preprocessing. Compared to the solutions for route/global EV simulation mentioned above, our work provides fast-yet-accurate and reasonable heuristics, based on greedy single-step optimization, that reflect typical human behaviours and realistic situations. Since our aim is to achieve large-scale simulations, computational costs are critical.

**Battery charge-discharge.** The authors of [289] consider that battery charging times are nonlinear using a particular cost function which takes this aspect into consideration. Indeed, while nearly linear for low state of charge, the charging rate decreases when arriving the battery limit. In [288] this aspect is modeled by matching a linear with an exponential function for high state of charge. At the same time, practical battery maintenance guidelines (e.g. [166]) suggest to refrain from reaching such charge limits, thus linear charging can be assumed when operating within the devised charge ranges. In our work we will adopt such a simplified model, although our framework can easily accommodate more complex ones.

**Mobility data analysis for EVs.** Similar to our objectives, some studies on EVs make use of mobility data, typically to understand the usual mobility needs of the population, and then check how well they fit the main characteristics of EV mobility. E.g., [337] analyzes both traditional ICE vehicles and one EV to check if typical range and mobility needs are compatible with the EV infrastructure in an extreme environment. In [82] the general urban mobility behaviour in two Italian

areas is studied, characterising trips length, average speed and parking duration distributions, which are then used to quantify the urban fleet share suitable to be converted to battery electric vehicles.

Compared to these works, our approach aims to analyze EVs' impact at a finer granularity, analyzing the single vehicle's mobility, providing more precise estimates and allowing a better understanding of the general phenomenon and of potential issues. Moreover, we aim at exploiting the IMNs processing tools and models, in order to infer basic semantic information that allows a more reliable simulation and a more insightful analysis of the results.

### 4.1.2 Problem definition

A simulation framework for EVs should aim to compute a mobility schedule that satisfies the (real) mobility demand  $S_u$  of a user  $u$  respecting the battery constraints of EVs and trying to minimize the overall cost that the user would experience in doing that. In particular, battery constraints require that the simulation identifies when and where recharges should take place. Also, the cost of a schedule might be defined in various alternative ways, such as the total time spent (probably the most natural choice, which will be adopted in our experiments), the amount of charge consumed, the overall distance travelled (considering that reaching a recharge station might require significant detours), and so on.

#### Optimal EV schedule

We start introducing the concept of EV schedules, basically consisting in a sequence of stops that cover those strictly required by the original mobility demand of the user, potentially adding new ones for recharge purposes, and that satisfy the basic requirements enforced by battery-powered vehicles.

**Definition 4.1.1** ((Valid) EV-schedule). An *EV-schedule* is defined as a tuple  $(S, c, r, b_1, b^*)$  composed of: a sequence  $S = \langle s_1, \dots, s_n \rangle$  of stop locations; a function  $c : \{2, \dots, n\} \rightarrow \mathcal{R}$  defining the amount  $c(i)$  of battery consumed for traveling from  $s_{i-1}$  to  $s_i$  ( $1 < i \leq n$ ); a function  $r : \{1, \dots, n\} \rightarrow \mathcal{R}^+$  assigning the amount of battery recharged at each stop  $s_i$ ; the initial battery level  $b_1 \in \mathcal{R}^+$ ; and a maximum battery capacity  $b^* \in \mathcal{R}^+$ .

EV-schedule  $(S, c, r, b_1, b^*)$  is said to be *valid* if the following holds:

1.  $\forall 1 \leq i \leq n : arrival\_batt(i) \geq 0$
2.  $\forall 1 \leq i \leq n : arrival\_batt(i) + r(i) \leq b^*$

where  $arrival\_batt(i) = b_1 + \sum_{j=1}^{i-1} r(j) - \sum_{j=2}^i c(j)$ .

Constraints (1) and (2) above express that the battery level at stops never exits its working limits, namely it never exceeds its maximum capacity and it is never negative. We also observe that functions  $c$  and  $r$  basically define, respectively, the charge and discharge operations applied during the simulation.

**Definition 4.1.2** (Compatible EV schedule). Given the sequence  $S_u$  of stops performed by a user  $u$  and a set  $R$  of recharge-enabled locations, we say that the EV-schedule  $\xi = (S, c, r, b_1, b^*)$  is *compatible with  $S_u$  and  $R$*  (or simply *compatible*, when clear from the context) if:

1.  $\xi$  is a valid schedule;
2.  $S_u \sqsubseteq S$ ;
3.  $\forall i. r(i) > 0 \iff s_i \in R$

with  $\sqsubseteq$  denoting the subsequence relation.

Finally, we introduce the optimization problem in a general way:

**Definition 4.1.3** (Optimal EV schedule). An *Optimal EV schedule* for user  $u$  is an EV schedule  $\xi = (S, c, r, b_1, b^*)$  compatible with her sequence of stops  $S_u$ , such that it minimizes the total cost  $C(\xi) = \sum \tau(i) + \sum \sigma(i)$ , where  $\tau(i)$  represents the cost of performing the trip between locations  $s_{i-1}$  and  $s_i$ , and  $\sigma(i)$  represents the cost of stopping/recharging at location  $s_i$ .

### Instantiating of the general problem definition

The definitions provided in the previous section can be instantiated into several different ways, depending on the factors that are considered more important to weigh for a specific application or context. In the following, we provide an instance of the general framework that we consider reasonable for the aim of this work, namely evaluating the potential impact of EV-based mobility over individual users' mobility. The instance is given by defining the parameters and functions involved in the definitions above. We first give some preliminary definitions.

**Definition 4.1.4** (EV-Map, path and bestPath). An *EV-map* is a road network  $G = (N, E)$  composed of nodes (i.e. road intersections)  $N$  and edges (i.e. roads)  $E$ , where each edge  $e \in E$  is a pair  $e = (n_1, n_2)$  of nodes and is associated with three attributes: length  $e.len$ , traversal time  $e.time$ , and battery consumption  $e.battery$ .

Given two nodes  $n_o, n_d \in N$ , a *path* from  $n_o$  to  $n_d$  is a sequence of connected nodes  $\langle n_1, \dots, n_k \rangle$  such that  $n_1 = n_o$ ,  $n_k = n_d$  and  $\forall 2 \leq i \leq k. (n_{i-1}, n_i) \in E$ .

Finally, the function  $bestPath_G(n_o, n_d)$  returns the path that minimizes the total travel time, i.e.  $\sum_i e_i.time$ , where  $e_i = (n_{i-1}, n_i)$ .

Then, we instantiate the general problem by defining the four key functions  $c(i)$ ,  $r(i)$ ,  $\tau(i)$  and  $\sigma(i)$  as follows.

**Definition 4.1.5** (EV Time Minimization Problem). Given a user  $u$  with their sequence of stops  $S_u$ , an initial battery level  $b_1$  and a maximum battery level  $b^*$ , the *EV time minimization problem* consists in finding an optimal EV schedule for  $u$  under the following definitions:

- Battery consumption of trips is computed as  $c(i) = arrival\_batt(i-1) - ch_i^k$ , where  $bestPath_G(s_{i-1}, s_i) = \langle n_1, \dots, n_k \rangle$ ,  $ch_i^1 = arrival\_batt(i-1)$  and  $\forall j > 1. ch_i^j = \min\{b^*, ch_i^{j-1} - (n_{j-1}, n_j).battery\}$ . This means that we choose the fastest path between two locations. Also, since roads can have a negative battery consumption (recharging when traveling downhill) we ensure that the battery level never exceeds the cap  $b^*$ .
- Recharge amount  $r(i)$  is defined in two different cases, depending on whether the visited location was already in the original schedule:

$$r^*(i) = \begin{cases} staytime(i) * power(s_i) & \text{if } s_i \in S_u \\ \infty & \text{otherwise} \end{cases}$$

$$r(i) = \min\{r^*(i), b^* - arrival\_batt(i)\}$$

where  $staytime(i)$  represents the duration of stop  $s_i$  in the original schedule and  $power(s_i)$  is the speed of the recharger available in  $s_i$ . In summary, recharges in the original stop locations last exactly the stop duration, whereas in other stops it lasts as much as needed to fill the battery up to the cap. Also, we assume that the recharge speed remains constant for a given station, regardless of the current battery level, which is an approximation.

- Trip cost  $\tau(i) = \sum_{j=1}^k e_j.time$ , for  $bestPath_G(s_{i-1}, s_i) = \langle e_1, \dots, e_k \rangle$ . Thus travel time is our cost for moving between locations, not considering other parameters at this stage, as for instance the battery consumption.
- Recharge cost  $\sigma(i)$  is computed as:

$$\sigma(i) = \begin{cases} 0 & \text{if } s_i \in S_u \\ r(i)/power(i) & \text{otherwise} \end{cases}$$

thus counting only the time spent recharging in the new stops introduced in the EV-schedule, and zero in the other cases (what we call also passive recharges). Recharge time is assumed to be linear in the amount of energy required, while queue waiting times at stations are not considered

Our final objective is then to evaluate the cost of an optimal EV schedule under time minimization as compared to that of its original, internal combustion engine-based one.

### The four simulation scenarios

A fundamental aspect that affects EV mobility is the availability of recharge options. Part of them are determined by the public infrastructures on the territory, which are basically the same for every user. Others depend on the user's status, which therefore might in principle condition their capability of safely replace the current ICE vehicle with an electric one. We consider the following four basic settings, covering a range of different recharge opportunities.

**Home Scenario.** The user can not only recharge on public stations when needed (thus making deviations for the actual trip and wasting time while waiting for the recharge), but also at home every time they stop there for at least a specific minimum duration, our default threshold being 20 minutes.

**Work Scenario.** Similarly to the Home Scenario, the user recharges at their work location every time they stop there for at least a minimum duration, the default threshold being 20 minutes also in this case.

**Home And Work Scenario.** Both Home and Work options are available. Clearly, this is the best scenario for the user, since there are more opportunities of recharging without spending time for reaching stations and waiting.

**Public Station Scenario.** The user can only recharge the battery at public stations. This represents the minimal scenario, and the need for recharging at public stations is expected to increase as compared to the other scenarios. We remark that, according to Definition 4.1.5, if the stop at a recharge station was already in the original schedule (i.e. the user was visiting that area for purposes unrelated to the station itself) then the recharge is not considered as a cost, since it is not subtracting free time to the user. In this case the recharge lasts only for the duration of the original stay. Since it involves a small overhead for the user (namely walking to/from the station to leave the vehicle and later getting it back in advance if the battery is full<sup>1</sup>), we allow this recharge option only for stays of at least 1 hour. In addition to the scenarios above, we define a situation that can happen when the EV range limitation makes it impossible to travel a leg of the schedule:

**Emergency Situations.** When the initial battery at a starting location is not sufficient to reach the destination nor any charging station, the user runs into an emergency situation. These cases will be counted separately, and will be a critical measure of the usability of EVs. In terms of simulation, we assume that in case of emergencies the vehicle is rescued and transported to destination, where it can continue the schedule with a fully-recharged battery. According to literature, the share of users belonging to each scenario is quite variable from country to country, yet typically showing a majority in the Home and Home-and-Work ones. From EU estimates [216] it follows that 33% can recharge at home, 11% at work, 44% in both, and just 12% only at public stations; also, these statistics fit with information available for Italy (e.g. [53] cites 80% of EV recharges performed at home) and other countries (e.g. Canada [247]).

### 4.1.3 Simulation framework

#### Setting the stage: battery charging/discharging on the map

**Battery consumption model** Our estimation of battery consumption for each trip of the user is based on a instantaneous consumption model introduced in [310] and recommended in [11] as a good trade-off between realistic simulation and efficient

---

<sup>1</sup>This assumption stems from the fact that many EV station operators apply surcharges if the vehicle is left unattended after completing the recharge.

computability. The model considers all the physical forces to which the vehicle is constantly exposed in order to estimate the amount of electric power needed to reach a certain speed, including in particular: Rolling resistance, based on tire characteristics, mass of vehicle and driver, and slope of the road; Aerodynamic resistance, depending on front surface of the vehicle, air density (in our case fixed to 20°C) and vehicle speed; Horizontal component of gravity, also depending on the road slope; Inertia of the vehicle, mainly depending on the acceleration of the vehicle. The model also considers the efficiency of the vehicle, such as engine and gear efficiency. Also, the energy consumption due to onboard electronics (lights, air conditioning, radio, etc.) is estimated.

Electric Vehicles have regenerative braking that can recover a fraction of the energy lost during decelerations and use it to recharge the battery.

In our simulations we will consider a medium class car model, with associated parameters. In particular, we will fix the maximum battery capacity to 40 kWh, which represents a lower-end setting. More details, including all the physical parameters adopted, are given in Appendix ??.

The final variation in the capacity  $SoC$  of the battery at time step  $t$  will be

$$SoC[t] = SoC[t - 1] - \Delta Soc - \delta_{self\_disch}$$

where  $\Delta Soc$  is the overall balance of energy consumption and  $\delta_{self\_disch}$  represents fixed self-discharge losses.

**Estimating elevation and speed** Our approach starts with building a road network containing some key information needed to find shortest routes and evaluate their battery consumption. We obtain a first version of the network through OpenStreetMap, which provides the map of the region of interest in the form of a graph. Then, we computed the maximum travel speed on each road, the altitude of the nodes and subsequently, the slopes of the respective edges. All this additional information has been inserted into the nodes and edges of the graph in order to obtain a structure containing the necessary information.

**Speed.** Regarding to the speed on the edges, the data provided by OpenStreetMap can be scarce, inaccurate and often missing. Indeed, only 10% of the road segments in our area of study had a maximum speed value specified. To remedy the lack of data, we exploited the Highway attribute, which is present on each edge and represents the type of road, including 'alley', 'crossing', 'emergency bay', 'living street', 'motorway', 'motorway link', 'primary', 'primary link', 'residential', etc. Each type was then associated to the most common speed limit adopted in Italy. Figure 4.1.1 depicts a speed-based colored map of Tuscany, comparing the original available information (left) and that obtained through reconstruction (right). The speed values are distributed according to a chromatic scale that goes from red for the lowest speeds to green for the highest speeds. The predominant color is yellow, representing the speed of 50km/h, which is consistent with the initial distribution of maximum speeds.



Figure 4.1.1: *Maximum speed in Tuscany (left) using the attribute MaxSpeed and (right) inferring it from the Highway attribute*

**Slopes.** To calculate the slope of each edge in the network, we started from the altitude of all the nodes. The Shuttle Radar Topography Mission (SRTM) data was used to obtain the elevation information (Figure 4.1.2). Then, the difference in height was calculated between the extreme nodes of each edge (Figure 4.1.3). In the calculation of the difference in height it was noticed that some were not gradual but they have jumps of several meters. This happens because the altitude data are calculated on the ground and therefore do not consider infrastructures such as tunnels, bridges and overpasses. To overcome this problem it was decided to consider the slope on those sections equal to zero. Another problem was the lack of altitude values for some nodes. In these cases, it was decided to assign the average of the values of the neighboring nodes.

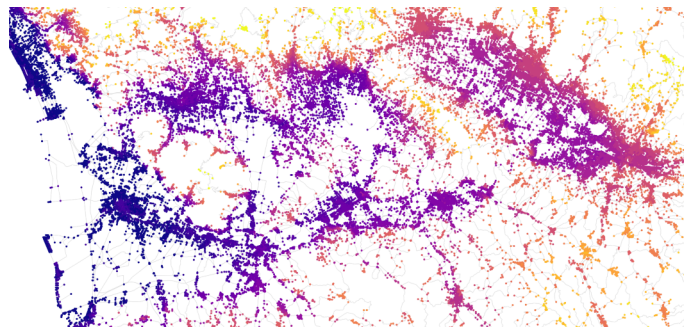


Figure 4.1.2: *Nodes altitudes on a portion of Tuscany (blue=low, yellow=high)*

**Estimating road-level discharge** In computing the battery consumption on one edge (road), the speed and the slope on the edge were assumed constant and the



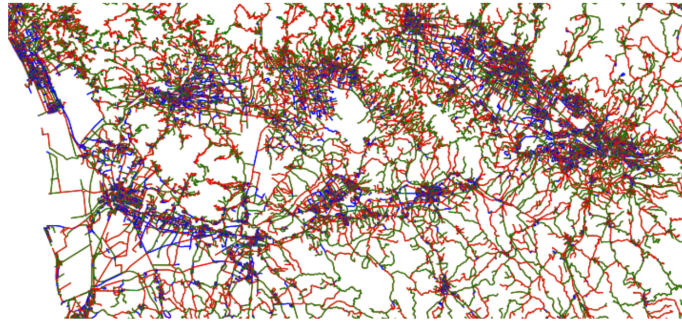


Figure 4.1.3: *Slopes of roads on a portion of Tuscany (blue=flat, red=steep)*

maximum speed limit for that edge was chosen as speed. Finally, the consumption by the vehicle on the edge was calculated, based on the battery consumption model described in the previous section.

**Integrating public recharge stations** Finally, we extracted from the public repository OpenChargeMap<sup>2</sup> the list of recharge stations available in the geographical area of our interest, and associated each of them with the closest node of our road network. Such nodes were labelled as public recharge locations, also describing the maximum recharge power available.

### EV-compliant best path computation

**Best path heuristics** As described in Section 4.1.2, the selected path followed to reach a destination is simply the fastest one, when the battery constraints allow that. When that is not possible, an intermediate stop at a charging station is performed, selecting the one that minimizes the overall time (travel for the new path plus recharge time). In complex situations, especially with long trips, one stop might be insufficient, thus requiring a more complex, multi-stop optimization that might greatly increase computation times. Our approach to the problem is a greedy solution that identifies the first recharge stop assuming that one stop is sufficient to reach the destination; then, if that is not the case, we repeat the same process to reach the destination from the current station (now starting with a full battery), thus greedily identifying the next stop, as above.

**Paths precomputation** Our simulation technique relies heavily on shortest-path computations. To improve the simulation performance, we perform offline shortest path precomputations to collect some useful information. However, considering the quadratic space requirements of storing the shortest paths of all node pairs in the road network graph, we limit the precomputations to/from charging station nodes and store only the aggregate information about paths.

We precompute the shortest paths between the charging stations and all other nodes

---

<sup>2</sup><https://openchargemap.org>

in the road network by using Dijkstra’s algorithm [83] adopting the edge traversal times as weights.

We denote with  $\mathcal{P}(n_i, n_j)$  the precomputed values for the shortest path between  $n_i$  and  $n_j$ . For each pair of so-called nodes, we store aggregate information about the shortest path’s total traversal time  $\mathcal{P}_{tot}^{time}(n_i, n_j)$ , spatial length  $\mathcal{P}_{tot}^{length}(n_i, n_j)$ , and consumption  $\mathcal{P}_{tot}^{cons}(n_i, n_j)$ . Storing only the sum of consumption values for path edges can not provide enough information to reason about the final battery level of the vehicle after traversing path edges. Note that the recharge during downhill edges may take the battery level beyond the maximum capacity. For instance, suppose the sequence of edge consumptions (and recharges) for a path is  $\langle -2, 3, -2, -1, 5 \rangle$ , and the maximum battery capacity is 7. The path’s total consumption is 3, but starting with an initial charge of 7 does not necessarily mean that the vehicle will have a battery level of  $7 - 3 = 4$  at the end of the path. That is because the first downhill recharge amount of  $-2$  in the sequence can not increase the battery level as the battery level is already the maximum capacity.

To account for this dynamic behavior, we store the maximum charge  $\mathcal{P}_{max}^{charge}(n_i, n_j)$  which indicates an upper bound for the battery level of the vehicle that guarantees the battery will not go beyond the capacity considering all the downhill recharges along the path. For the example above, this upper bound is 5. If the vehicle starts with a charge of 5, the ultimate battery level at the end of the path is exactly  $5 - 3 = 2$ . Moreover, we store minimum charge  $\mathcal{P}_{min}^{charge}(n_i, n_j)$ , which indicates the minimum charge needed for the vehicle to reach from  $n_i$  to  $n_j$ . This value is particularly important to filter the reachable charging stations from  $n_i$  based on the current battery level  $arrival\_batt(n_i)$  when the vehicle needs a recharge.

Algorithm `findMinMaxCharge` summarizes the computation of the so-called minimum and maximum initial battery levels for a sequence of path consumptions  $consSeq$  and the battery capacity of  $C$ . In line 1, we first compute the sequence of prefix sums of  $consSeq$ . The prefix sum indicates the vehicle’s battery levels, starting with a battery level of zero from the first edge of the path.

---

**Function** `findMinMaxCharge`

---

**Input** : consumption sequence  $consSeq$ , and battery capacity  $C$   
**Output:** ( $minInitCharge$ ,  $maxInitCharge$ )

- 1  $prefixSumSeq \leftarrow$  compute prefix sum of  $consSeq$
- 2  $minInitCharge \leftarrow \max(prefixSumSeq)$
- 3  $maxInitCharge \leftarrow C$
- 4 **if**  $\min(prefixSumSeq) < 0$  **then**
- 5 |  $maxInitCharge \leftarrow C + \min(prefixSumSeq)$
- 6 **return** ( $minInitCharge$ ,  $maxInitCharge$ )

---

The maximum prefix sum of  $consSeq$  specifies the minimum amount of charge

needed for the vehicle to follow the path successfully; obviously, this maximum value should not exceed  $C$  as in that case, the path would be unfeasible for the vehicle with even a full battery (line 2).

On the other hand, if the minimum of this prefix sum sequence denoted by  $ps_{min}$  is negative, then the path has a subsequence that charges the battery. Thus, if the vehicle arrives at that edge with a charge of  $C$ , no recharge will happen as the battery is already full. Thus, in that case, the upper bound of the battery level to avoid exceeding the battery limit is  $C + ps_{min}$  (line 4). However, if  $ps_{min}$  is positive, we can infer that the upper bound would be the battery's capacity (line 3).

For the example above, the prefix sum sequence is  $\langle -2, 1, -1, -2, 3 \rangle$  and the minimum and maximum initial charges are 3 and  $7 + (-2) = 5$ , respectively. We formalize the above discussion in the following, where we also specify how to exactly compute the final battery level:

*Theorem 1.* Given a path having *prefixSumSeq*, *minInitCharge* and *maxInitCharge* as defined in Function *findMinMaxCharge*, we have that:

- if the initial charge  $c$  is such that  $c < minInitCharge$ , then the path is not doable by the vehicle;
- if  $minInitCharge \leq c \leq maxInitCharge$ , then the path is doable, and the final charge is  $c - prefixSumSeq$ ;
- if  $c \geq minInitCharge$  and  $c > maxInitCharge$ , then the path is doable, and the final charge is  $maxInitCharge - prefixSumSeq$ .

*Proof.* If  $minInitCharge \leq c \leq maxInitCharge$ , then the battery level  $c(t)$  remains within the interval  $[0, C]$  throughout the path with no battery overflows nor empty battery issues, thus the final level will simply be  $c(t_{final}) = c - prefixSumSeq$ . Assuming that  $c < maxInitCharge$  (and thus there are no overflows), if  $c < minInitCharge$  there is a time  $t^*$  in the path where the battery goes below zero ( $c(t^*) < 0$ ), and thus the path is not doable. It is easy to see that if  $c > maxInitCharge$  then the battery levels  $c'(t)$  obtained at each time  $t$  are such that  $\forall t : c'(t) \leq c(t)$ , and thus also in this case  $c < maxInitCharge \Rightarrow c(t^*) < 0$  and the path is not doable. Finally, if  $c > maxInitCharge$  there is at least a time  $t^\top$  where  $c(t^\top) > C$  and the battery level is capped to  $C$  and thus the battery levels from this point on will be exactly the same we would obtain (with no overflows) when  $c = maxInitCharge$ , hence  $c(t_{final}) = maxInitCharge - prefixSumSeq$  by applying the formula of the first case.  $\square$

The precomputation of the shortest paths is justified mainly due to the fact that the road network or the charging stations in the network do not change frequently, and the precomputation can be done offline when significant changes happen to the network.

### User history EV-simulation

Algorithm 4 summarizes our proposed procedure for the overall simulation of EV users' mobility. The procedure is based on a set of general simulation parameters that are listed in Table findMinMaxCharge and are all passed to Algorithm 4. In addition, the simulation algorithm receives the road network graph, the user's individual mobility network, and the sequence of the user's trajectories sorted chronologically. Also, three boolean parameters indicate the possibility of using home and work chargers and whether moderate discomfort in reaching destinations is allowed so the user can leave the vehicle in a public charging station instead of the exact destination.

Parameter	Description	Default
$k_1$	Weight of the first leg of the path (origin→recharger)	0.4
$k_c$	Weight for time to fully recharge the battery at the station	0.2
$k_2$	Weight of the second leg of path (recharger→destination)	0.4
$ms$	Min stay duration threshold for charging	20 mins
$mdms$	Min stay duration threshold for moderate discomfort	60 mins
$maxRecharges$	Maximum number of recharges allowed during a single trip	3
$b^*$	Battery capacity	40 kWh
$b_1$	Initial charge	$b^*$
$\mathcal{P}$	precomputed data	N/A

Table 4.1.1: *General Simulation Parameters*

The simulation starts with obtaining the assumed home and work locations of the user, and their corresponding nodes on the road network (lines 3-4). Then the origin and destination for every trip in user's history  $\mathcal{T}$  is used to simulate the sequence of consecutive locations that the user intends to visit. Hence, the graph nodes  $orig$ ,  $dest$  that correspond to the origin and destination of the trip are obtained (line 6). The duration of user's stay at  $orig$  which is basically the time they spent from the end of the previous trip until the start of the current trip is computed in line 7.

If the home and work recharges are allowed, and the user stays there for at least a predefined amount of time  $ms$ , a home or work recharge is done before the next trip starting from  $orig$  (lines 8-11). Moreover, if *moderateDiscomfort* is enabled and the user intends to stay for a long enough time, the option of recharging at a public station close to the  $orig$  is also considered (lines 12-13).

Next, in lines 16-21, the algorithm tries a shortest path from  $orig$  to  $dest$  with the weight criteria of edge travel times. If the shortest path is feasible given its sequence of edge consumptions, then *reachedDestination* is set to *True* and both  $orig$  and  $dest$  are inserted into the simulated path sequence  $\mathcal{S}$ . Otherwise, depending on the current battery level  $cc$ , a list of reachable charging stations is fetched from the precomputed information  $\mathcal{P}$  (line 23). Note that, there might be no reachable charging stations if the battery level of the vehicle is too low, in which case, the current trip simulation is interrupted (lines 24-25).

Then, the best charging station *recharger* is selected among the possible options using the heuristics described in section 4.1.3 (line 27). Considering the discussion in section 4.1.3, if the current charge of the vehicle is lower than  $\mathcal{P}_{max}^{charge}(orig, recharger)$ , then the vehicle will consume the energy equal to the sum of path's edge consumptions. Otherwise, the same computation will be performed,

**Algorithm 4:** (EV Strategy)

---

**Input** : Road network graph  $G$ , user's IMN  $\mathcal{IMN}$ , user's trajectories  $\mathcal{T}$ ,  
Comfort Parameters:  $homeCharge$ ,  $workCharge$ ,  
 $moderateDiscomfort$ ,  
General Parameters:  $k_1$ ,  $k_2$ ,  $k_c$ ,  $ms$ ,  $mdms$ ,  $maxRecharges$ ,  $b_1$ ,  $b^*$ ,  
 $\mathcal{P}$  (Table findMinMaxCharge)  
**Output:** Simulated Path  $\mathcal{S}$

```

1  $\mathcal{S} \leftarrow \langle \rangle$ 
2  $cc \leftarrow b_1$  //  $cc$  is the current charge
3  $homeNode \leftarrow G.getNode(\mathcal{IMN}.homeLoc)$ 
4  $workNode \leftarrow G.getNode(\mathcal{IMN}.workLoc)$ 
5 foreach  $t \in \mathcal{T}$  do
6    $orig, dest \leftarrow G.getNode(t.s), G.getNode(t.e)$ 
7    $stayDur \leftarrow$  stay duration at  $orig$  before starting  $t$ 
8   if  $homeCharge$  and  $orig$  is  $homeNode$  and  $stayDur \geq ms$  then
9      $cc \leftarrow$  recharge at home
10  else if  $workCharge$  and  $orig$  is  $workNode$  and  $stayDur \geq ms$  then
11     $cc \leftarrow$  recharge at work
12  else if  $moderateDiscomfort$  and  $stayDur \geq mdms$  then
13     $cc \leftarrow$  recharge at the fastest charging stations for the stay duration
14     $reachedDestination \leftarrow False$ ;  $rechargeCount \leftarrow 0$ 
15    while not  $reachedDestination$  and  $rechargeCount \leq maxRecharges$ 
16      do
17         $p \leftarrow SP(G, orig, dest, weight = 'traveltime')$ 
18         $consSeq \leftarrow$  consumptions of  $p$ 's edges
19         $(isFeasible, cc) \leftarrow isFeasible(consSeq, batCap, cc)$ 
20        if  $isFeasible$  then
21           $reachedDestination \leftarrow True$ ;  $\mathcal{S}.insert(orig)$ ;  $\mathcal{S}.insert(dest)$ 
22          break
23        else
24           $rcs \leftarrow \{charging\ stations\ cs \mid cc \geq \mathcal{P}_{min}^{charge}(orig, cs)\}$ 
25          if  $rcs = \emptyset$  then
26            break
27           $recharger \leftarrow \arg\min_{s \in rcs} k_1 \cdot \mathcal{P}_{tot}^{time}(orig, s) + k_2 \cdot \mathcal{P}_{tot}^{time}(s, dest) +$ 
28             $+ k_c \cdot recharge\_time(s)$ 
29          if  $cc < \mathcal{P}_{max}^{charge}(orig, recharger)$  then
30             $cc \leftarrow cc - \mathcal{P}_{tot}^{cons}(orig, recharger)$ 
31          else
32             $cc \leftarrow \mathcal{P}_{max}^{charge}(orig, recharger) - \mathcal{P}_{tot}^{cons}(orig, recharger)$ 
33             $cc \leftarrow b^*$  (full recharge and collect recharge time statistics)
34             $rechargeCount \leftarrow rechargeCount + 1$ 
35             $\mathcal{S}.insert(orig)$ 
36             $orig \leftarrow recharger$ 
37          if not  $reachedDestination$  then
38            EMERGENCY condition
39             $cc \leftarrow b^*$ 
39 return  $\mathcal{S}$ 

```

---

---

**Function** isFeasible

---

**Input** : consumption sequence  $consSeq$ , battery capacity  $C$ , and charge  $b$ **Output:** (isFeasible, remainingCharge)

```

1  $finalCharge \leftarrow b$ 
2 foreach  $cons$  in  $consSeq$  do
3    $newCharge \leftarrow finalCharge - cons$ 
4   if  $newCharge < 0$  then
5     return (False, None)
6   else if  $newCharge > C$  then
7      $finalCharge \leftarrow C$ 
8   else
9      $finalCharge \leftarrow newCharge$ 
10 return (True,  $finalCharge$ )
```

---

yet capping the initial battery level at  $\mathcal{P}_{max}^{charge}(orig, recharger)$  (lines 28-31). Then, a recharge is done at  $recharge$  (line 32), the number of recharges for the current origin/destination pair is incremented (line 33), the node  $orig$  is added to  $\mathcal{S}$  and  $recharger$  is set as the origin node for the next trial in reaching the destination (lines 34-35). This process repeats until either the vehicle reaches the destination or the number of recharges exceeds the  $maxRecharges$  threshold. In case it was not possible to reach the destination, the algorithm raises an emergency condition and continues with simulating the next trip (lines 36-38).

#### 4.1.4 Experiments

##### Setting up the stage

In this section we describe the dataset used and the general setting of the experiments carried out. Also, some properties of the dataset are explored, to give the reader a better understanding of the application context. **Dataset** Our experiments are based on a dataset of real GPS traces of 1000 private vehicles moving in the Tuscany region in Italy, and spanning 2 months, namely March and April 2017. The vehicles, in particular, are residents from (i.e. their main location belongs to) five provinces of the region: Arezzo, Firenze, Lucca, Pisa, Pistoia, although their trips can span all the region. The raw data is segmented by identifying stops as points where a vehicle remains virtually in the same place (namely, within a distance of 50 meters) for at least 20 minutes. Each trip is then represented by its pair of origin and destination points, filtering out those that are shorter than 1 km (typically representing cases where the vehicle is simply parked in a different slot in the same area). This results in a total of 176'300 trips. In order to make the results of our simulation perfectly comparable with the real mobility data, the trip between each origin and destination pair is reconstructed through a fastest path heuristics – the same used in the simulation, yet with no battery constraints –, storing its length and duration. This operation reduces the impact of imperfections in the road network (missing edges,

wrong directions, incorrect speed, etc.) over the comparison process. Each origin and destination point is snapped to the closest node in the road network, and we use the shortest path function provided by the OSMnx library [45] with cost defined as the traversal time of each edge.

**Home and Work** In order to implement the simulation scenarios it is essential to identify the locations representing home and those representing the work places. We do that following the approach described in [257, 212, 126], which infers a graph structure of the user’s mobility named *Individual Mobility Network* (IMN), and identifies visited locations and their frequency. The most frequently visited location is then selected as home and the second most frequent one as work place of the user.

**Dataset statistics** As shown in Figure 4.1.4, most users have more than 100 trips in the observations period. Also, the average number of trips per day ranges from 2 to 8 for the large majority of users. This suggests that the movement history of the users analyzed is significant.

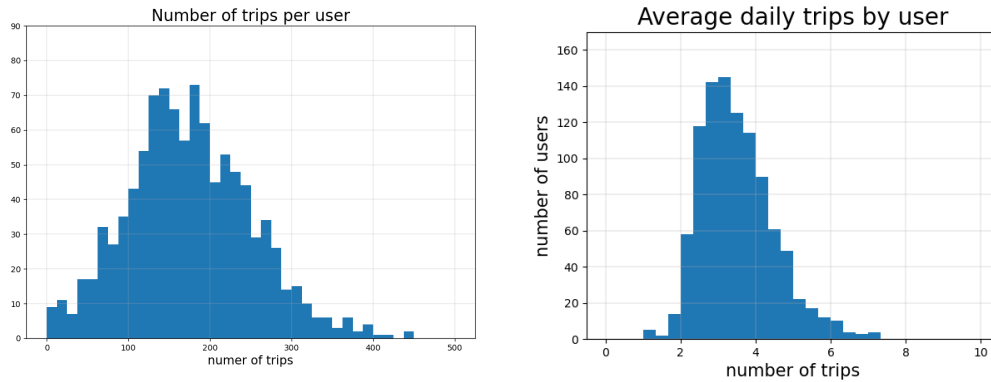


Figure 4.1.4: *Distribution of the number of trips involved in the experiments: (left) trips per user; (right) trips per day of each user.*

The length of the trips is studied in Figure 4.1.5(left), and shows a log-normal shape, representing the fact that there are many trips of medium-short length (less than 10km, in most cases), as well as a significant number of moderately long ones (between 10km and 40km) and a small fraction of long trips (longer than 40km up to 300km), which appears to be coherent with the extension of the geographical area considered. Figure 4.1.5(right) compares duration and length of the trips, showing a linear relation, as expected, with a variability that grows with the trip length. Longer trips, belonging to the tail beyond 100km, appear to have a more stable average speed (around 100 km/h), probably due to the fact that they are mostly performed along high-speed roads.

The experiments make use of two sources of geographical data: OpenStreetMap, for the road network of Tuscany, which is composed of 138792 nodes (intersections) and 305804 edges (road segments); and OpenChargeMap, for the catalogue of recharge stations available on the territory and their power, for a total of 354 stations. Figure 4.1.6 shows the distribution of recharge power and the geographical

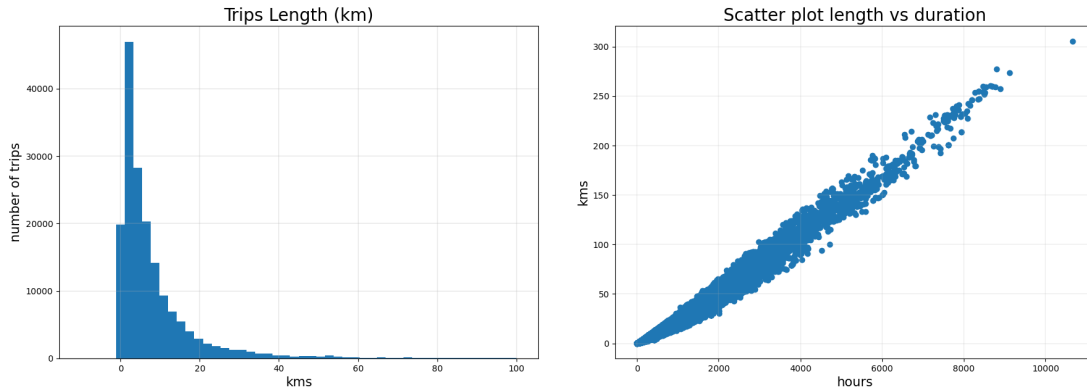


Figure 4.1.5: *Distribution of trip lengths (left) and duration (right).*

disposition of stations. Recharge stations are grouped by power (expressed in kW) and power category, following the current standard classification (see, e.g., [184]): *slow* if  $power < 7kW$ , *fast* if  $7 \leq power < 25kW$ , *rapid* if  $25 \leq power < 100kW$ , *ultra-rapid* if  $power > 100kW$ . The most common ones are fast rechargers, especially those with a power of 22 kW, although there is a good number of rapid ones, and a few ultra-rapid. The large majority of stations are on the Northern part of Tuscany, in particular along the line connecting Florence and Pisa.

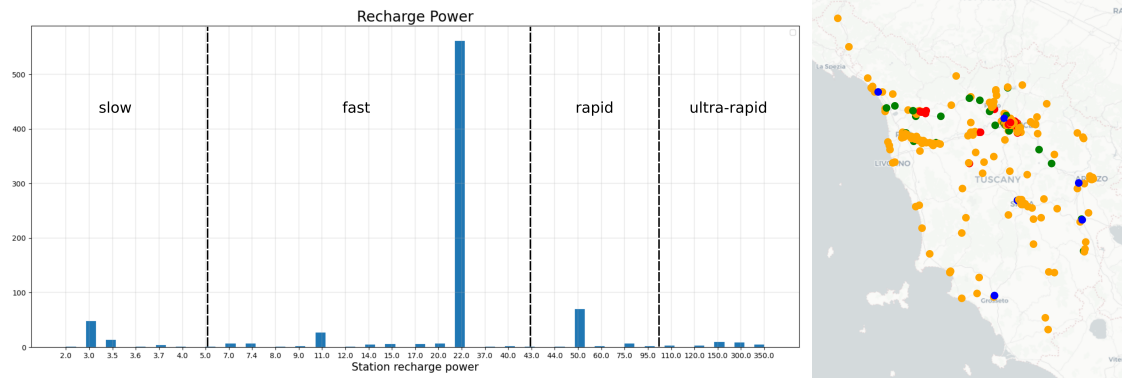


Figure 4.1.6: *(left) Number of stations by maximum recharge power provided (in KW); (right) Geographical distribution of stations (red=slow, orange=fast, green=rapid, blue=ultra-rapid).*

## Runtime evaluation

In this section we evaluate the scalability of the proposed approach w.r.t. the input dataset size (namely the number of trips involved in the simulation) and its complexity (namely the length of trips). We consider the two extreme scenarios: public-only recharges and recharging also at home and work.



Figure 4.1.7 shows the growth of runtimes for increasing sizes of the input dataset. In particular, each data sample is composed of the first  $N$  trips in chronological order, and the ticks in the plot at every 20k trips also correspond approximately to one more week of data. Runtimes grow linearly in the input size for both scenarios, as we could expect given that each trip is only loosely dependent on the previous ones. Also, the public-only scenario has slightly higher runtimes due to its higher chances of deviations to recharge stations, which add complexity to the process and increase runtimes. We remark that these time measures do not consider the fixed cost of the pre-computation phase described in Section 4.1.3, which depends only on the geographical area of interest and not on the mobility data. In our experiments that added an amortized cost equivalent to  $\sim 221$  milliseconds per trip, assuming to amortize it over a single simulation run (i.e. a single scenario with a single set of parameters). While still significant, it easily becomes negligible when multiple runs are needed or larger datasets are employed.

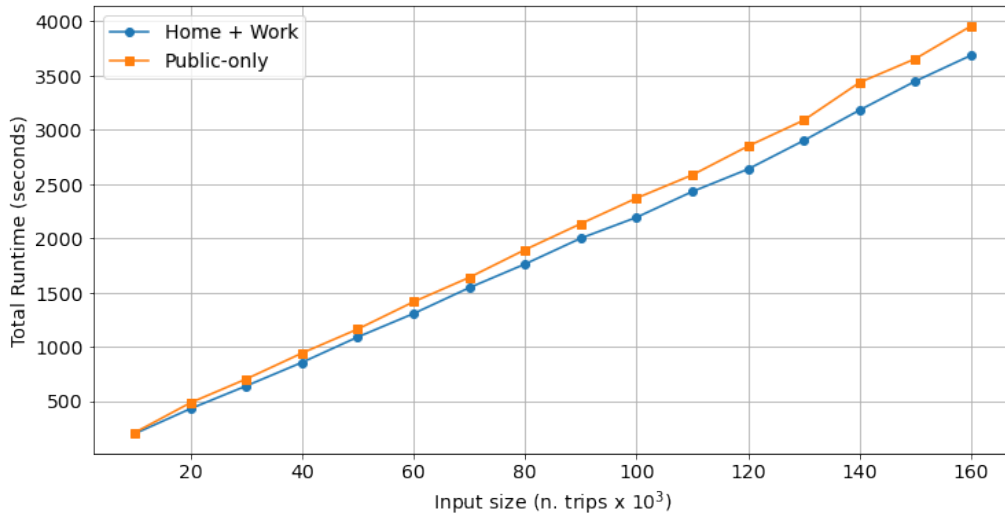


Figure 4.1.7: *Simulation runtime for different input sizes (seconds vs. n. of trips).*

Simulating short trips is expected to be less expensive than longer ones, since the latter usually involve a more complex shortest path computation and there also higher chances that an additional stop at a recharge station needs to be planned. The plots in Figure 4.1.8 analyze the runtime distribution of single trips grouped by trip length, for the two scenarios (home and work on the top plot, and public-only on the bottom). In both cases, we can see that runtimes grow approx. linearly with the trip length, although trips above 130 km are not frequent enough (see gray line in the plot) to draw clear conclusions from them. Also, the vast majority of trips (more exactly, 96.9%) are less than 30 km, for which the single trip cost is virtually always less than 50 milliseconds. The main difference between the two scenarios is the variability of runtimes, which appears to be slightly higher for the public-only one.

As mentioned in the related works, most competing approaches available in lit-

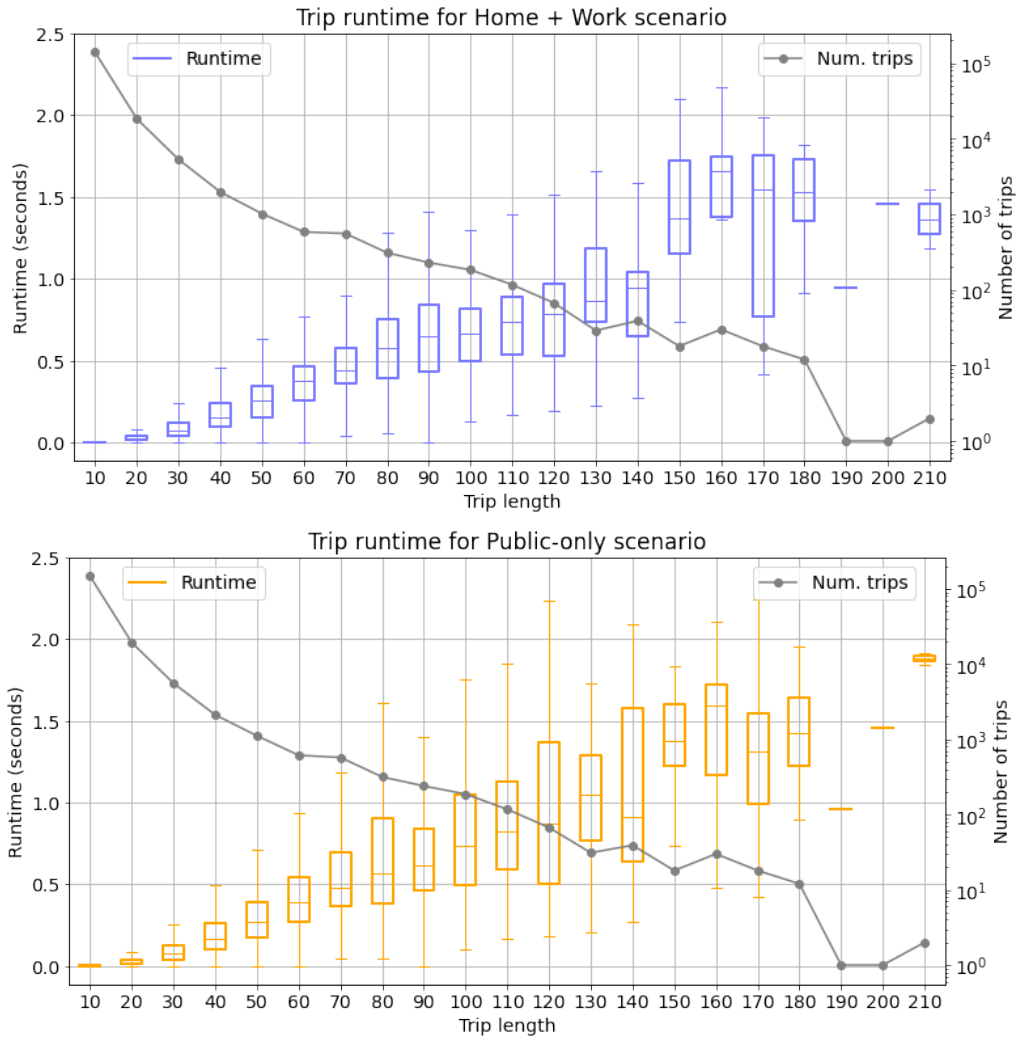


Figure 4.1.8: Average runtime for trips, divided by trip length, for the Home and Work scenario (top) and the Public-only one (bottom). The black line represents the number of trips in each group (log-scale).

erature aim to precisely optimize the single trip, yet requiring much higher computational costs – indeed, the optimal routing algorithm behind them (e.g. [300]) is known to be  $\mathcal{NP}$ -hard, and thus hardly applicable to medium-sized setting as the one we are considering here, which requires to simulate over 170k trips over a road network composed by over 300k edges. Our heuristics, instead, results to be efficient enough to run each experiment in around 1.2 hours on a single commodity machine.

### Simulation results

**Overall impact of EV on individual mobility** The core of the experiments consists in a comparison of the original trips against the simulated ones on the four scenarios (charge at public stations only; public and home; public and work

Table 4.1.2: Overall impact of EVs on trips. We focus on the comparison between the average length and the average duration of real and simulated trips.

Scenario	Avg. length (km)			Avg. duration (s)			Emergencies	Recharges
	REAL	EV	DELTA	REAL	EV	DELTA		
public-only	9.323	9.415	+0.98 %	7'9"	8'26"	+18.11 %	1270 (0.72%)	2.26 %
work		9.358	+0.37 %		7'36"	+6.29 %	328 (0.19%)	0.95 %
home		9.337	+0.15 %		7'18"	+2.06 %	94 (0.05%)	0.41 %
home + work		9.334	+0.12 %		7'16"	+1.79 %	60 (0.03%)	0.34 %
sample mix	avg	9.347	+0.26 %		7'27"	+4.34 %	246 (0.14%)	0.66 %
	std	±0.00	±0.02 %		±0'1"	±0.22 %	±15 (±0.01%)	±0.02 %

place; public and home + work) over the whole dataset. In addition, we created 100 random *sample mixes* of the four scenarios, associating each user to one of them by following the representative distribution statistics for (some) EU countries provided in [216], according to which 77% of users can recharge at home, 55% at work and 12% in none of them. The results are summarized in Table 4.1.2. First, we can see that the average lengths of the trips in the four scenarios are very similar to the original ones, with an increase of less than 1%, signifying that deviations for recharging are on average modest. In terms of trip duration, the worst-case scenario yields increments that are moderate in absolute terms (+1'17") and yet, given the typical short lengths of trips, are significant in relative terms, reaching a 18.11%. This percentage very quickly drops to moderate levels when recharge-at-work is introduced, and to modest ones with recharge-at-home. In general, we observe that the highest increases are observed when only public stations are available for recharging, which are strongly reduced by recharges at home, and slightly less by recharges at work. When both options are available, their synergy actually produces slight improvements.

The simulations yield a 0.75% of emergencies in the public-only scenario, which is relatively large. We believe this to be an overestimate of real user issues, mainly caused by the insufficient distribution of recharge stations in Tuscany (currently covering only larger cities and main ways), aggravated by the incompleteness of OpenChargeMap (we estimate it is missing 30% stations). With the growth of the EV infrastructures, we expect that these factors will be greatly alleviated in the near future. Introducing other recharge options drastically reduces emergencies down to 0.03% for the home + work case. Similar results are obtained for the percentage of trips with recharges at stations. Finally, we observe that the representative sample mix achieves rather low values, that are between the recharge-at-work and the recharge-at-home scenarios. Overall, the results show that by applying the simple charge management heuristics considered in this paper, the majority of trips incurs into minor deviations from the original ones. Considering the sparseness of the current recharge infrastructures available in the area of study, that provides positive feedbacks for individual users about the feasibility of switching to an EV without changing any aspect of their mobility habits.

These results can be seen from the perspective of the single individual, in order to understand if the overall moderate average impact measures shown actually hides some portion of users that are largely affected. Figure 4.1.9 represents the distribution of the duration increase (left) and distance increase (right) measured

Table 4.1.3: *Temporal variations of EVs impact for the four scenarios. The two months period is split in four shorter periods of two weeks each ( $t_1$ ,  $t_2$ ,  $t_3$  and  $t_4$ ) in order to see how the percentages change inside the selected time.*

	Increment Length %				Increment Time %				% Recharge Trips			
	$t_1$	$t_2$	$t_3$	$t_4$	$t_1$	$t_2$	$t_3$	$t_4$	$t_1$	$t_2$	$t_3$	$t_4$
public-only	0.68	0.99	1.04	1.19	12.24	18.82	19.5	20.8	1.61	2.25	2.3	2.77
work	0.3	0.36	0.35	0.42	4.77	6.67	5.89	7.64	0.7	0.94	0.9	1.23
home	0.08	0.18	0.13	0.23	1.49	1.79	1.82	3.12	0.31	0.37	0.36	0.59
home/work	0.05	0.12	0.1	0.19	1.33	1.52	1.59	2.73	0.26	0.3	0.3	0.5

aggregating times and distances by user. As we can see, the figure not only confirms that introducing home/work as recharge opportunities the impact is reduced (the peak around a 0 increase grows significantly), but also that virtually no user suffers increases above 4% in any of the scenarios.

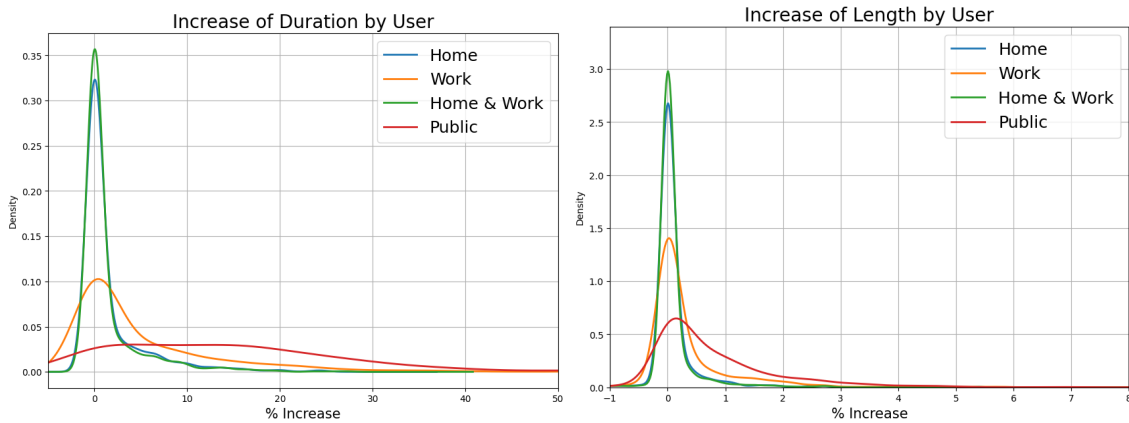


Figure 4.1.9: *Distribution of trip lengths and duration aggregated by user for the 4 scenarios.*

**Temporal stability of results** In order to evaluate if the results obtained are time dependent, we provide in Table 4.1.3 aggregates over four consecutive bi-weeks for the four scenarios. We can observe that there is indeed some variability, and also a general slight increase on all measures considered, yet always remaining at low levels and well below 1%. The increase can be justified by considering that some vehicles travel relatively little, and thus it is unlikely that they will need recharging during the first bi-weeks, concentrating recharges (and, consequently, deviations) later in the period.

**Spatial stability of results** Since our dataset spans a significantly large area, and the recharge infrastructures are not homogeneously distributed, we try to understand if trips in different provinces suffer from deviations of different intensity. Table 4.1.4 summarizes the results. Here we can see that, indeed, different areas show a different impact level. In particular, Pisa benefits from a larger number and better distribution of recharge stations at least in relation to its size, and thus shows a significantly smaller impact than the others. Firenze and Lucca have similar values that are much larger than Pisa, most likely because of the large extension of

Table 4.1.4: *Geographical variations of EVs impact. Each province is associated to the trips that start from it.*

	Increment Length %					Increment Time %					% Recharge Trips				
	Arezzo	Florence	Lucca	Pisa	Pistoia	Arezzo	Florence	Lucca	Pisa	Pistoia	Arezzo	Florence	Lucca	Pisa	Pistoia
<b>public-only</b>	1.0	1.13	0.96	0.55	0.93	14.3	22.2	17.0	13.6	19.6	1.5	2.59	1.96	2.04	1.9
<b>work</b>	0.21	0.4	0.38	0.14	0.26	3.35	7.21	4.81	3.99	5.76	0.47	1.05	0.76	0.67	0.65
<b>home</b>	0.03	0.1	0.09	0.04	0.03	0.42	0.92	0.87	0.96	1.0	0.1	0.28	0.21	0.24	0.15
<b>home/work</b>	0.02	0.06	0.05	-0.01	0.01	0.27	0.71	0.62	0.52	0.5	0.07	0.21	0.16	0.13	0.08

Firenze, resulting in a lowered density of stations, and the limited number of stations in Lucca, only partially balanced by the proximity to Pisa and its infrastructures. Finally, Arezzo and Pistoia are less covered by stations and also slightly peripheral to the other big cities. As final remark, we notice that the impact of the single provinces are smaller than the aggregates shown in Table 4.1.2. This is due to the fact that trips originated outside provinces are not included here, and they indeed tend to be longer and traversing less populated (also in terms of stations) areas.

**Impact of heuristics’ parameters** When a direct trip to a given destination is not possible because of a battery charge shortage, the proposed heuristics builds a path passing through a reachable recharge station, which is chosen by considering the travel time to reach the station, the travel time to reach the destination from the station, and the recharge time. The weight associated to each component is defined, respectively, by parameters  $k_1$ ,  $k_2$  and  $k_{charge}$ . In this paragraph we discuss the effects of different choices of parameters, in terms of performances and also in terms of station usage.

Since stations can have very different recharge speeds, it can happen that the heuristics chooses to perform large deviations (and thus spend more time to travel) in order to reach a fast recharger and thus spend less time recharging. For this reason, high values of  $k_{charge}$  are expected to increase the usage of highest speed stations. Figure 4.1.10 shows the distribution of recharges on the different station types, grouped by power/speed, for different values of  $k_{charge}$ . As we can see, when the weight of recharge time is null, recharges are strongly concentrated on relatively slow stations, namely those labeled as “fast”, which are the most popular on the territory. With  $k_{charge} = 0.2$  the distribution immediately changes, and the peak is now on the “rapid” group. Further increasing  $k_{charge}$  has little effect on the slow/fast group, whereas the rapid one slightly decreases in favour of the “ultra-rapid”. In all cases, the slowest stations in the “fast” category and those in the “slow” one have a marginal role, since they are not common enough nor convenient in terms of recharge time.

Tables 4.1.5 describes how the impact of EVs changes when varying the travel time parameters ( $k_1$  and  $k_2$ ). Increasing  $k_2$  has no clear effect on the length and duration of trips, while it apparently leads to a slight reduction on the number of recharges required. This might be motivated by the fact that high values of  $k_2$  promote recharges which are closer to the final destination, which is thus reached with more charge in the battery for the following trips. A low  $k_2$ , instead, would

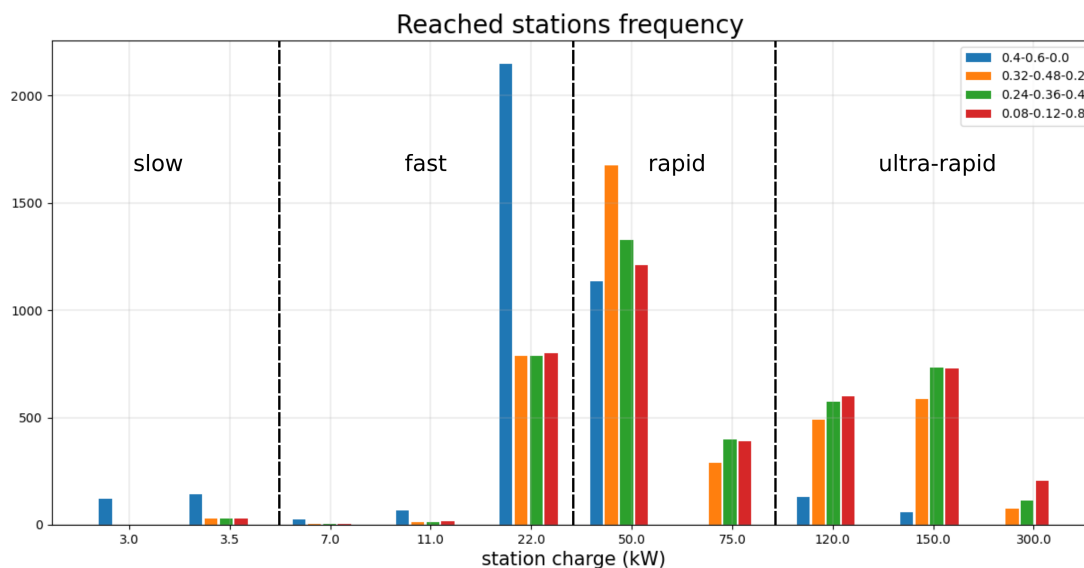


Figure 4.1.10: Usage frequency of stations by power by increasing values of  $k_{charge}$  (weight of recharge time in the path selection algorithm).

Table 4.1.5: Effects of varying  $k_1$  and  $k_2$  on EVs impact.  $k_{charge}$  is fixed to 0.2.

Parameters		% Increment	% Increment	% Trips
$k_1$	$k_2$	Length	Time	with Recharge
0.1	0.7	0.89	19.25	2.19
0.2	0.6	0.93	18.65	2.22
0.3	0.5	0.96	18.31	2.24

favor early stops at stations along the trips, resulting in lower battery levels at the destination.

**Validation of results** In order to test whether our solution provides results coherent with other existing EV-related services, we compared it against the popular online EV-based trip planner ABRP (<https://abetterrouteplanner.com/>) through a small-scale experiment covering 20 users over one day. Results show that the trips generated by ABRP have a similar length (average difference around 2%) and a significant, yet stable increase in driving time (35.3%) and recharging time (24.6%) – which can be attributed to ABRP referring to real-time, and thus traffic-affected, road status information. This small comparison suggests that our results, expressed as relative increase/decrease of times, are overall coherent with ABRP. More details are provided in [? ].

## Case studies

In this section we closely examine the impact of EVs on two sample users, each under two scenarios: charge at home and work, and charge only at public stations. User A is characterized by a moderate number of recharges performed (12 in the

worst case), while user B had a higher number (28 in the worst case). Figures 4.1.11 and 4.1.12 show the results respectively for user A and user B, adopting a double visual representation of the EV-based mobility for each scenario.

The IMN representation in Figure 4.1.11(top-left) shows that in the public-only scenario user A performed several recharges along different trips of their history (see the red edges), plus rather frequent passive recharges (i.e. while stopping at a trip destination) in two locations (see the small red nodes). The temporal charge plot on the bottom-left also confirms that recharges at stations (in red) are approx. uniformly distributed in time, while passive recharges (in green) are less frequent. The corresponding home + work scenario is shown on the right plots. As we can see, recharges are now much more concentrated on the home and work location, significantly reducing the recharges at stations (especially for trips starting/ending at home or at work) and also strongly reducing the other passive recharges. The temporal graph confirms this, and also shows that the battery level is generally kept much higher than in the previous scenario. For this user, the overhead in adopting an EV seems to be moderate, and reduced almost to zero when recharging at home and work is possible.

User B, shown in Figure 4.1.12, starts from a much more complex situation, as they require rather frequent recharges at public stations in the public-only scenario and does not benefit of passive recharges. Moving to the home and work scenario, recharges at home reduce significantly the usage of public stations, yet there is no dominant work place, and the overall result is that also in this scenario a significant number of stops at public stations is needed, especially in the central period of time. User B is not only energy-hungry, but their mobility distribution also makes the effect of passive recharges less incisive. Overall, this appears to be a user that might require more effort in the transition to EVs.

#### 4.1.5 Notes on privacy and ethical issues

In this paragraph is presented a brief analysis of the potential privacy risks that could be hidden in this work. The simulation framework for EVs described above should aim to compute a mobility schedule that satisfies the (real) mobility demand of a user respecting the battery constraints of EVs and trying to minimize the overall cost that the user would experience in doing that. All the simulation is based on extracting some information from real data and then creating alternative trips that take into account the EV needs. Starting from the individual real trajectories the only points of departure and arrival for each trip are extracted and the identification of the places most frequented by each user (probably home and work). This involves many risks related to the possibility of matching this data and tracing the identity of the user. Simulating the rest of the journey instead, there is no knowledge about the trip, the speed or any driving characteristics of the user. Above all, nothing is known about stops or detours made before arriving at the point of arrival. Compared to the many hidden risks behind the individual mobility network (see details in sec 4.4), for this research the identification risks are minimized since the full use of GPS

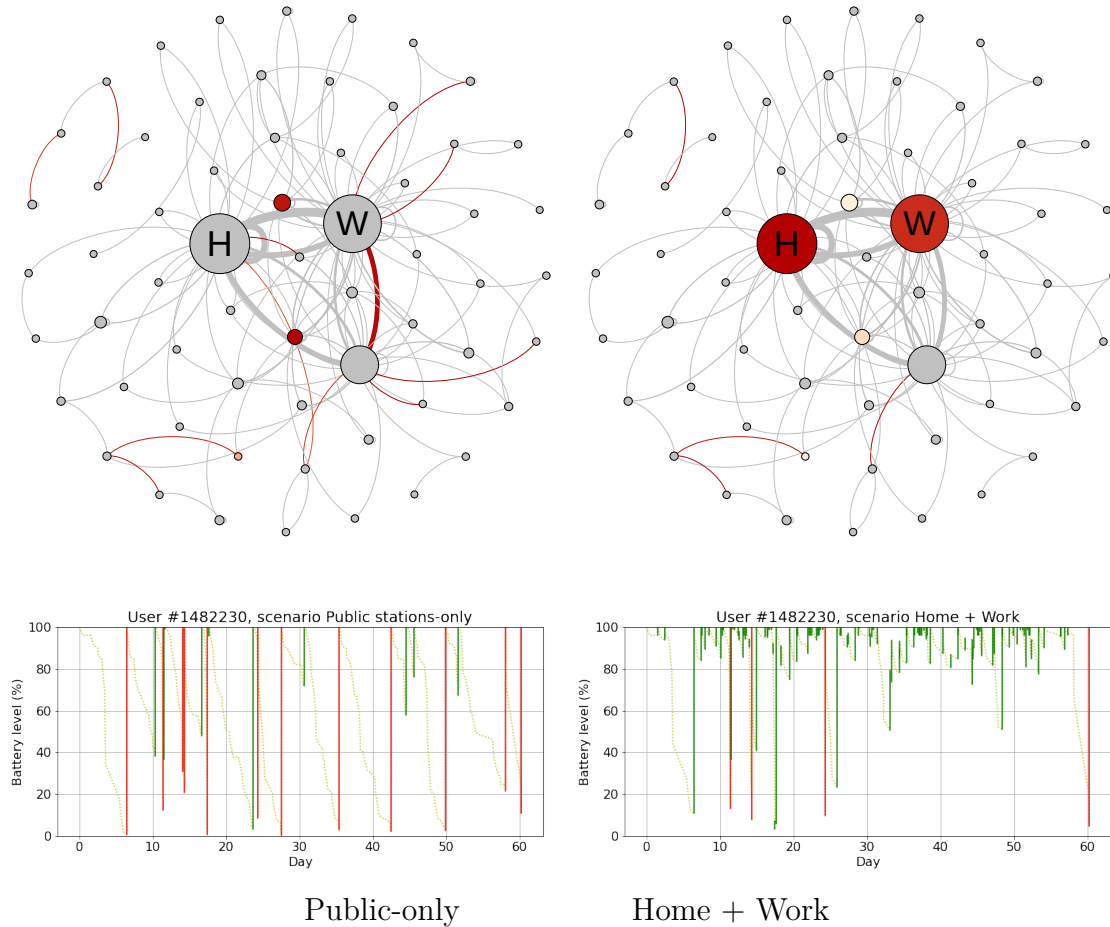


Figure 4.1.11: **Use case A:** *IMNs (top) and temporal graph of charge (bottom). Left: Home + Work scenario; right: Public-only. Size and width in IMNs represent frequency of stop/trip, darkness of red represents frequency of recharge. In the temporal graph, passive charges are green, those at stations are red.*

data it is outside the scope of the work.

Therefore there are not other obvious risks of privacy attacks than those already mentioned in 4.4.

#### 4.1.6 Conclusions

In this work we proposed a methodology to analyze the process of switching the current private mobility from fossil fueled cars to their corresponding electric version. Our process, which combines mobility data analytics, ad-hoc trip planning, and simulation, allows for a quantitative analysis of the current fuel-based mobility of a user and the potential impact of transitioning to EVs.

Through the use of our approach, we can analyze the impact of EVs on the mobility routine of a real-world user, considering the potential challenges posed by limited driving range and charging infrastructure availability.



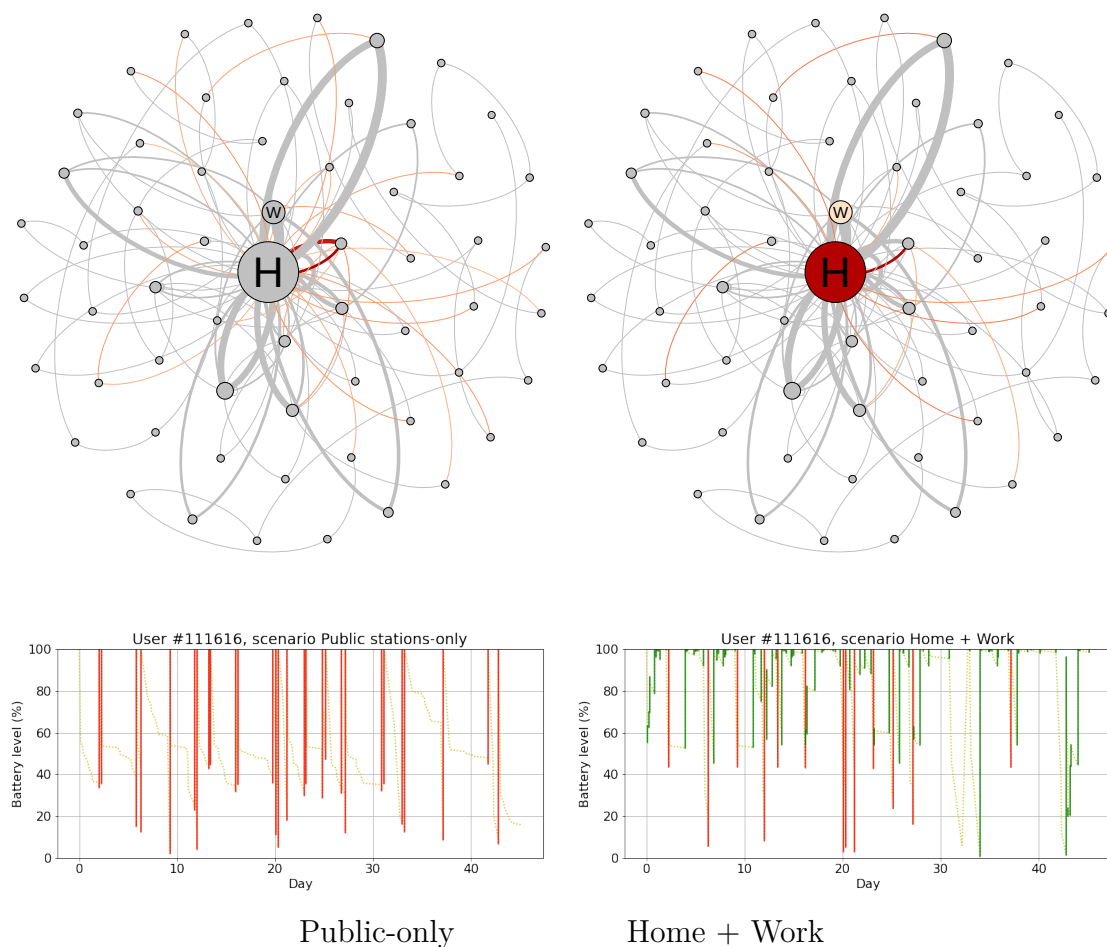


Figure 4.1.12: *Use case B: same layout as Figure 4.1.11.*

Overall, our proposed process provides a valuable tool for individual users to understand how much their mobility fits the EVs' requirements, as well as for decision makers to make informed choices about designing charger networks to meet the specific needs of a given region. By considering the user's specific mobility needs and habits, we can provide personalized insights and recommendations to help guide their decision-making process. **Use case results and application.** The results obtained over an Italian region shows how the electrification process is expected to generate only moderate issues at the collective level (mainly, marginal increases in distance traveled and overall moderate time spent at recharge stations), and yet individual users can expect slightly different impacts in they travel & refuel habits. We envision that these results (and the tool in general) can help various actors of the mobility scene: decision makers in better planning the charging infrastructures by simulating the impact of installing new stations or improving their speed; car makers to support the design of models that better fit users' needs; and the single users, that can better understand their personal fitness to EV under different conditions and car models (e.g. choosing their personal best trade-off between battery capacity and cost).

**Limitations and open problems.** Though a ready-to-use tool, the proposed approach is still amenable to improvements in several directions that we aim to explore: integrating/estimating waiting times at stations, as well as a more complete map of charge stations; considering the variability of power provided by chargers as effect of the time-variable energy grid load; studying the effect of different battery capacities; studying the impact of EVs on user costs and environmental factors; finally, devising processes and setting up experiments to achieve a stronger validation of results, and a better calibration of the tool.

## 4.2 Comparative cities studies through City Indicators

Classifying a geographical territory into semantic categories is one of the most common tasks in research areas such as urban geography, urban planning and mobility data analytics. Characterizing human mobility is a key component of this process, and it is well known that mobility often does not work the same way across different regions. A movement pattern in a mountainous countryside may have other implications than the same pattern has in the suburbs of a large town. The movement trajectories in a planned city with rectangular streets and strict zoning laws might be completely different than the ones in a town that has grown organically without any clear structure. Therefore, any kind of property that was learned in a particular area, in general cannot simply be assumed to hold in another one.

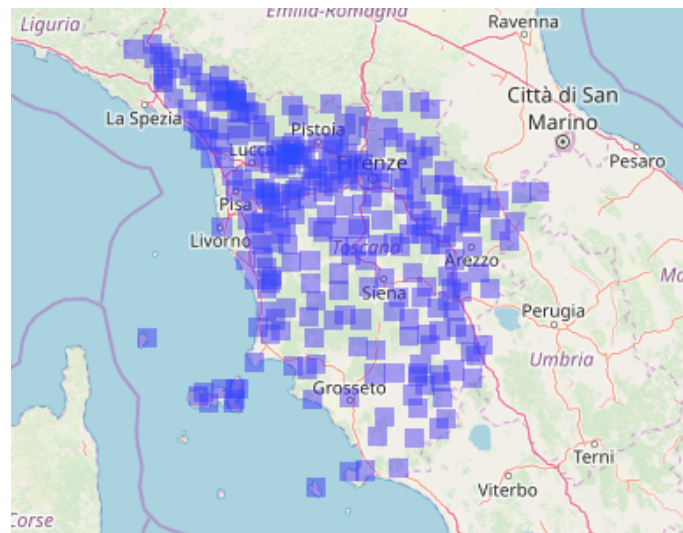


Figure 4.2.1: *The areas of study: 10×10km squares centered on each municipality in Tuscany.*

### 4.2.1 Local City Indicators

Here we introduce the local city indicators designed individually for each municipality. They are grouped in spatial concentration measures, flows measure, individual mobility and street network.

#### Spatial Concentration

Spatial concentration is one of the most important aspects in the description of urban regions and answer the question *how the density of people and activities vary across the area?* This question was traditionally focused on people's residency and workplace, since that was the only available data, mostly coming from census or

government records. More recent research is profiting from the availability of more detailed data from mobile phones, vehicle trackers and satellite imaging [154, 306, 15]. Spatial concentration is used in a vast range of different fields [106, 275, 116, 127, 126]. In this work, the concept of spatial concentration is focused on the overall amount of mobility, undifferentiated by types of activity. The question of interest is: *are the activities concentrated in cluster-like centers of high density or are they spread-out across the map?*

In the following, we present three approaches to answer this question: *spatial entropy*, *Moran's measure*, and the *average nearest neighbor distance*. The first two approaches can only be calculated after the geographical space has been partitioned into a set of disjoint areas. In this work, we do that adopting an equally-spaced grid, and divide the  $100\text{km}^2$  region representing each area using different resolutions, including a grid of  $10 \times 10$  (i.e. each cell is a square of side 1 km),  $20 \times 20$  and  $50 \times 50$  cells.

## Entropy

It can be used to measure how equally activities are distributed across the grid. Let  $X$  be a discrete random variable modeling the positions of an individual ending up in  $n$  different fields [28]. The entropy is defined as [274]:

$$E(X) = - \sum_{i=1}^n P(x_i) \log P(x_i)$$

where  $\{x_1, \dots, x_n\}$  are the possible values of  $X$  and  $P(x_i)$  is the probability of  $X$  being in state  $i$ . For maximum entropy ( $\log(n)$ ) there is an equal amount of activity in all fields; for minimum entropy (0) all the activity is amassed in a single field. In order to compare entropy scores of different-sized grids, the measure must be normalized by dividing it by the expected entropy of a uniform distribution, i.e.,  $\log(n)$ .

## Moran's I

It overcomes the entropy weakness by considering how the fields are positioned in space: spatial autocorrelation [258] that represents the degree to which the fields' values are correlated to the value of neighboring fields. For spatial autocorrelation, the nearness between all pairs of fields must be defined with a so-called *weight matrix*  $w$ , where  $w_{ij}$  is the nearness between nodes  $i$  and  $j$ . A simple form of weight matrix is an adjacency matrix, with the value 1 if fields are adjacent, 0 otherwise. An important difference to the entropy is that spatial autocorrelation has two directions. A high autocorrelation indicates that values of the same magnitude are prone to be next to each other, while a low autocorrelation means that similar values are less likely to be near each other than under random positioning. Somewhere in between lies a value of autocorrelation in which the population of the fields is how one would expect it to be under a random distribution with no spatial autocorrelation. The

most famous autocorrelation measures is *Moran's I* [209]:

$$I(X) = \frac{N \sum_i \sum_j w_{ij} (x_i - \bar{x})(x_j - \bar{x})}{W \sum_i (x_i - \bar{x})^2}$$

where  $N$  is the number of fields,  $x$  is the amount of activity or population,  $\bar{x}$  is the average field value, and  $W$  is the sum of all the weights. The minimum and maximum values of *Moran's I* depend on the weight matrix. We highlight that the absence of autocorrelation is given at *Moran's I* equals to  $-1/(N-1)$ , that tends to zero in grids with an high amounts of fields.

### Nearest Neighbor Distance

The *Average Nearest Neighbor Distance* (ANND) is not dependent on a grid and its parameters. For every point, the distance to its nearest neighbor is calculated. The mean of those values is the *ANND* :

$$ANND = \frac{\sum_i \min(d_i)}{N}$$

where  $d_i$  is a vector containing the distances of point  $i$  to all the other points, and  $N$  is the amount of points. The lower the *ANND*, the higher is the average spatial concentration in the areas surrounding the points. We highlight that this definition bears a similar weakness as the entropy. The expected *ANND* under assumption of a uniform distribution of points across the area is the *Mean Random Nearest Neighbor Distance* (*MRNND*)  $MRNND = 0.5\sqrt{A/N}$ , where  $A$  is the surface of the area and  $N$  the amount of points. By dividing the *ANND* by *MRNND* we obtain the *Nearest Neighbor Index* (*NNI*) which is comparable among samples with different sizes and areas. A *NNI* smaller than 1 indicates a higher spatial concentration than in a random case, whilst value above 1 shows that the points are spread out across the map more than one expects in a random scenario.

### 4.2.2 Flows in a Grid Network

In order to capture the information about flows in urban regions, the data can be transformed into a directed weighted graph that represents the flow of the people's trajectories:

- a set of nodes  $V$  representing places that are origins and destinations of trajectories,
- a set of edges  $E$  representing the directed connections between the nodes,
- a weight function  $w : E \rightarrow \mathbb{R}$  that maps each edge to a weight, which indicates the amount of trajectories that occur along the edge.

The map is split into fields of a grid and all origins and destinations of the trajectories in the area are assigned to the field in which they lie. The network is created by assigning every node to a cell, and to each edge the weight the amount of flows occurring along the edge. The weight function  $w$  is equivalent to an origin destination matrix. The network allows us to gain knowledge about the structure of a region by looking at the properties of the resulting network described in the following.

### Node Degrees

A basic property of the network is the distribution of its degrees. Degree is hereby defined as the total traffic (sum of in- and out-flow) of a grid field. This measure is sometimes also referred to as node-flux [265].

### Louvain Modularity

An interesting quality of networks is the degree to which nodes can be partitioned into groups, such that the connectivity is high within those groups, and low in between. In the context of urban regions, the corresponding question is: can the city be split into areas that are relatively autonomous and have only low interaction between them? In network science, *modularity* measures this property for a given partitioning: a graph partitioning separates the graph's nodes into non-overlapping communities. Modularity shows the difference between the relative amount of inner-community links and the expected relative amount under random linking in a non-directed weighted graph [25]. The modularity goes from  $-1$  to  $+1$ , where  $0$  marks the value expected in a network where all possible edges have the same expected weight. We highlight that the direction of traffic flow is not important here. Thus, the grid networks in this work are transformed into non-directed networks before the modularity is calculated. Modularity does not describe a network on its own, but a network along with its partition. In order to quantify how well an urban region is separable into different sub-areas we adopt the *Louvain Algorithm* [44] that does not guarantee an optimal solution but it performs well empirically.

### Interaction Models

The flow network allows us to test how well the empirical data aligns with two established models that describe human interaction in space. The *Gravitation Model* [10] idea is that the traffic flow from place  $i$  to place  $j$  depends on the origin population  $m_i$  and the destination population  $n_j$ . Highly populated places, attract flow towards them. The classic model predicts the traffic flow from  $i$  to  $j$  which have a distance of  $r$  as  $G_{ij} = Am_i^\alpha n_j^\beta / r^\gamma$ , where  $A$  is a normalization factor, and  $\alpha, \beta, \gamma$  are the model's parameters. They can be optimized by multiple regression when fitting data to the model. In this work we adopt a simpler model [204] with  $\alpha = \beta = 1$ . The *Radiation Model* [278] updates  $G_{ij}$  by introducing  $s_{ij}$  that is the population within a circle around place  $i$ , with a radius of its distance to place  $j$ , minus  $m_i$  and  $n_j$ . The intuition is that outgoing trips are being attracted by nearby populations [204]. It

predicts the flow  $T_{ij}$  as  $\frac{T_i}{1-\frac{m_i}{M}} \frac{m_i n_j}{(m_i + s_{ij})(m_i + n_j + s_{ij})}$  where  $T_i$  is the sum of outflows from  $i$ , and  $M = \sum_i m_i$  is the total sample population.

### 4.2.3 Individual Mobility

Here we consider the mobility at level of individual users. From this perspective, urban regions can be described by aggregated values of their inhabitants' mobility, therefore a set of statistics are calculated for each individual from their trajectories:

- Average distance and duration per trip
- Average driving distance and duration per day
- Average amount of trips per day

As we see in Sec.3.2 individuals' mobility data can be transformed into a Individual Mobility Network, which describes the individual mobility of a user through a graph representation of her locations and movements, grasping the relevant properties and removing unnecessary details.

From an *IMN*, we can describe the individuals travel behavior with the following indicators:

- *Size of the network*: number of nodes and edges.
- *Temporal-uncorrelated entropy*: measure how equally the different places of the *IMN* are visited.
- *Radius of gyration* [232]: approximates the average distance of an individual from its center of mass [118].
- *Regularity of trajectories*: percentage of trips that are driven more often than a certain threshold per time [129, 126].
- *Modularity*: the *Louvain Algorithm* [44] applied to the *IMN*.

### 4.2.4 Roads and Traffic

#### Static Road Network

This section focuses on the road network modeled as a directed graph ( $G = (E, V)$ ), where  $V$  is a set of nodes representing roads intersections,  $E$  is the set of directed edges which model the the road segments, and  $l : E \rightarrow \mathbb{R}$  maps each edge to its length in meters. Some *basic statistics* of the road network can be calculated:

1. amount of edges and nodes/node density
2. amount of intersections/intersection density
3. average node degree/average intersection degree

## 4. total length of edges/mean edge length

In addition, since nodes in any network can be evaluated w.r.t. their centrality, we evaluate the *road network's closeness centrality* in terms of the length of the shortest path to any given node. The average of those path lengths is a node's average farness from other nodes. The reciprocal of this value is a node's closeness centrality  $C(x) = \frac{1}{\sum_y d(y,x)}$ , where  $x$  and  $y$  are nodes and the  $d$  returns the length of the shortest path between its arguments. As distance function we consider the length as the summed road lengths of the edges of the shortest path [245].

### Traffic in the Road Network

To investigate how traffic is distributed in a road network one must *map match* the sequences of GPS locations that represent the trajectories to nodes and edges in the road network. There is a variety of algorithms that handle this problem, such as hidden Markov models [217]. In the case study of this work, a simpler algorithm was implemented due to the high reliability of the data. It independently maps every point of a trajectory to a node in the road network. The nodes are then connected and build a path that describes the individual's trajectory.

Given a map matching, it is possible to create a function that reveals the fraction of total traffic that flows through a given percentage of the most dense roads. For this purpose, all edges are sorted by their traffic flow in a non-ascending order. Cumulative traffic, measured as  $\#cars \times meters$ , is calculated for the end of every edge by multiplying the edge length with the amount of traffic flow and adding the result to the previous amount of cumulative traffic. The intermediary values within edges can be calculated by linear interpolation. For any given percentage of roads, the percentage of traffic in those roads is calculated by dividing the cumulative traffic until that point by the total amount of traffic.

### 4.2.5 Global City Indicators

In this section we introduce the *global* city indicators designed to compare two cities. To compare and cluster cities in groups, we need some quantitative features. Therefore, we have to define some metrics describing a city with respect to traffic. A possible approach is to exploit again a *network* structure where each city (in our case study, 276 municipalities in Tuscany) is a node, and edges are drawn based on the trajectories between them. Starting from the trajectories we infer descriptive attributes from two perspectives: (i) graph measures from the complete network of cities; (ii) graph measures from the ego-network of each city.

### 4.2.6 Complete Network of Cities

We can derive a set of global indicators through a *network of cities* as described in the following. Given the trajectories on the territory, we can derive an *Origin-Destination Matrix* (OD), which measures the number of trips that starts from city



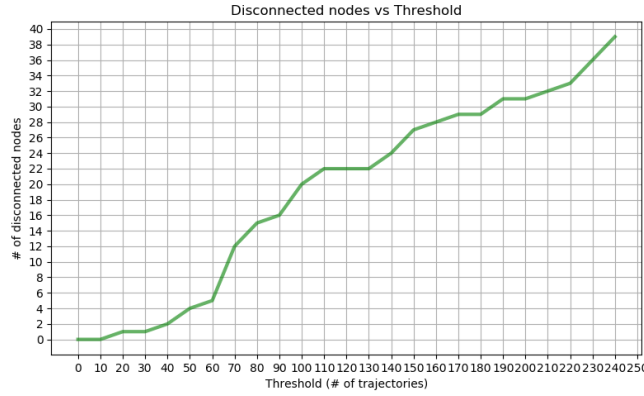


Figure 4.2.2: Disconnected nodes vs. flow threshold.

$A$  and ends in city  $B$  for each pair  $(A, B)$ . Since connections established through very few trajectories might be not significant, a threshold is needed to establish if an edge should be drawn. In our case study, after empirical evaluation, we fixed this threshold to 110 trajectories by analyzing the results yielded by different values through Figure 4.2.2. The plot shows the number of disconnected nodes corresponding to a selected threshold. The fraction of “isolated” cities grows as the threshold increases, but there is a little *plateau* between 110 and 130, which led to our choice.

With the selected threshold, the final graph consists of 276 nodes (corresponding to municipalities), 22 of which are disconnected from the *giant component*.

The properties related to each node of the network constitute the first set of attributes to be considered for clustering:

- **Self-loops:** # trajectories starting and ending in that node.
- **In/Out degree:** fraction of nodes its incoming/outgoing edges are connected to.
- **Closeness:** the closeness centrality of a node  $u$  is the reciprocal of the average shortest path distance (see Section 4.2.4).
- **Betweenness:** the betweenness of a node  $v$  is the sum of the fraction of all-pairs shortest paths that pass through  $v$ .
- **Clustering coefficient:** the local clustering coefficient  $C_i$  for a vertex  $v_i$  is given by the proportion of links between the vertices within its neighborhood divided by the number of links that could possibly exist between them.
- **Radius of Gyration:** the radius of gyration of a city  $c$  is defined as  $r_g(c) = \sqrt{\frac{1}{N} \sum_i w_i (r_i - r_{cm})^2}$ , where  $N$  is the total number of travels from  $c$ ,  $w_i$  is the number of travels from  $c$  to  $i$ ,  $r_i$  is the pair of coordinates of location  $i$  and  $r_{cm}$  is the center of mass (i.e., the average position) of the visited cities starting from  $c$ .

- **Random Entropy:** the random entropy captures the degree of predictability of the destination starting from a city  $i$  if each location is visited with equal probability  $S_{ran} = \log_2 M$ , where  $M$  is the number of distinct cities visited starting from city  $i$ ;
- **Uncorrelated Entropy:** the temporal-uncorrelated entropy is the historical probability that a location  $j$  was visited starting from a city  $i$ , characterizing the heterogeneity its of visitation patterns  $S_{unc} = -\sum_j^N p_j \log p_j$  where  $p_j$  is  $i$ 's probability of visiting location  $j$ . We can also normalize the uncorrelated entropy by dividing it by  $\log_2 N$ .

### 4.2.7 Ego-Networks

In Social Network Analysis, it is usual to refer to *Ego Networks* as social networks made of an individual (called *ego*) along with all the social links he has with other users (called *alters*)[75, 18]. Several fundamental properties of social relationships can be characterized by studying them. Adapting the terms to the present context, we can obtain an ego network for each city, where the ego is the city itself and the alters are its neighbors. The additional set of attributes obtained consists of:

- **Number of nodes** of the ego network.
- **Number of edges** of the ego network.
- **Average clustering coefficient:** the clustering coefficient is the average  $C = \frac{1}{n} \sum_{v \in G} c_v$ , where  $n$  is the nbr. of nodes in  $G$  and  $c_v$  is the clustering coefficient of each node;
- **Diameter:** is the longest shortest path of the ego network.
- **Assortativity:** is measured as the Pearson correlation coefficient of degree between pairs of linked nodes. It measures the preference for a network's nodes to attach to others that are similar in some way.

### Case study: transfer-compliant geographical locations

The huge amount of urban data generated by smartphones, vehicles, and infrastructures (e.g., traffic cameras, air quality monitoring stations) opens up new opportunities to learn about city dynamics from a variety of perspectives and facilitates various smart city applications for traffic monitoring, public safety, urban planning, etc. – all contributing to what is called *urban computing*.

However, there are some questions that remains still almost unexplored: what if the administration of a city wanted to predict the impact of an event on the urban mobility without having historical data on it? Is it possible to infer some useful insights exploiting the experience gained by other municipalities? Can knowledge be transferred from any city or are there some constraints? How can you compare two cities, for example in terms of urban mobility? Lately there have been different

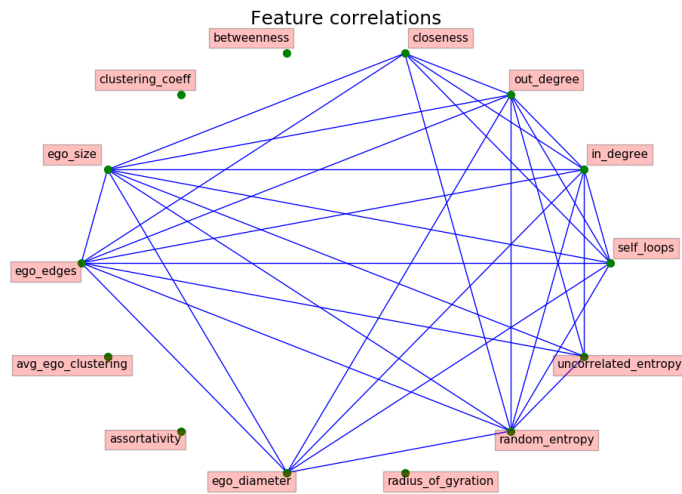


Figure 4.2.3: *Network of correlations for the first set of attributes (total graph).*

attempts to overcome the data scarcity issue in “new” urban contexts. All these studies have in common the application of *Transfer Learning*, a very broad family of approaches which focuses on developing methods to transfer knowledge learned in one or more “*source tasks*”, and use it to improve learning in a related “*target task*”. This section studies these questions in the context of Machine Learning (ML) and big data analytics for mobility data. In particular, our goal is to verify the feasibility of a *model transfer*, i.e., a ML model is trained in the source domain and then transferred to the target domain, in the prediction of urban traffic, exploiting the city indicators developed in the previous sections.

The basic idea is that cities that are similar can be represented by the same model more easily than very different cities. For instance, a highly populated city with heavy traffic and users that frequently make long trips is expected to have mobility dynamics very different from small, country-side cities with low traffic. The approach proposed in this section is developed in three steps: first, using a similarity measure between cities based on the indicators presented in Sections ?? and 4.2.5, cities are clustered into similarity groups; next, for each city a traffic prediction task is defined, which is approached through a standard machine learning solution (XGBoost regression [65]); finally, the prediction model of a city is applied to make predictions in each of the others, aiming to test whether cities in the same cluster show a better transferability of their models.

## 4.2.8 City Clustering

In this step, the city indicators built in the previous sections are first preprocessed and filtered, and then used to cluster cities.

**Preprocessing and Feature Selection** Since the range of different indicators varies widely, we applied a form of normalization to make them homogeneous. We

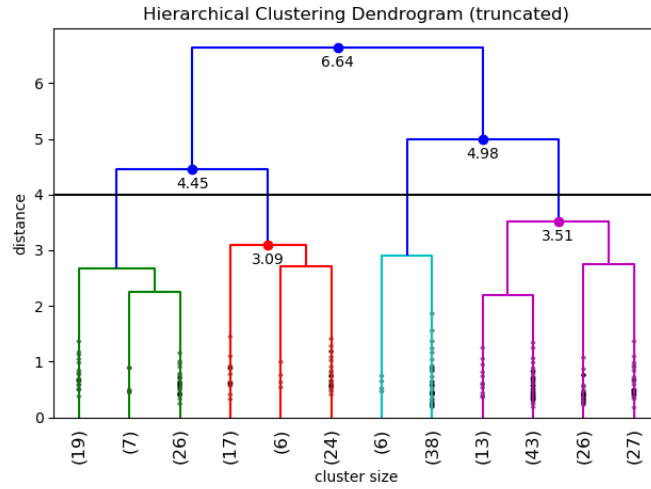


Figure 4.2.4: Dendrogram and selected clusters.

cluster id	# of cities	% of cities
0	22	8.0
1	53	19.2
2	47	17.0
3	110	39.9
4	44	15.9

Table 4.2.1: Cluster Population

adopted the *min-max scaling*, where feature are re-scaled in the interval  $[0, 1]$ . Then, we performed a study of correlation on each set of features (local and global) to eliminate unnecessary ones. To efficiently filter them, we adopted a network-based correlations finder, where the features are interpreted as nodes of a graph, and a link is drawn between two features if they are highly correlated. As evaluation metrics, the standard *Pearson's Correlation Coefficient* is used [237].

Considering each couple of features  $(i, j)$ , an edge is drawn if  $\rho_{i,j} > 0.65$ . The result obtained on the global features is shown as example in Figure 4.2.3. The removal of features is an iterative process that removes the node (feature) with the highest degree (thus is correlated to the highest number of non-filtered features) and repeats until the average degree of the network is 0. The remaining nodes are the features which are preserved. This preprocessing step is applied to global and local indicators separately, and then on the set of survived features of both categories. Applying the procedure to our case study, the initial set of indicators, composed of a total of 178 measures, was reduced to 21 features.

**Hierarchical Clustering** The city clustering step has been realized through a Hierarchical agglomerative clustering schema, adopting Ward's linkage criterion, which at each step of aggregation aims to minimize the total within-cluster variance.

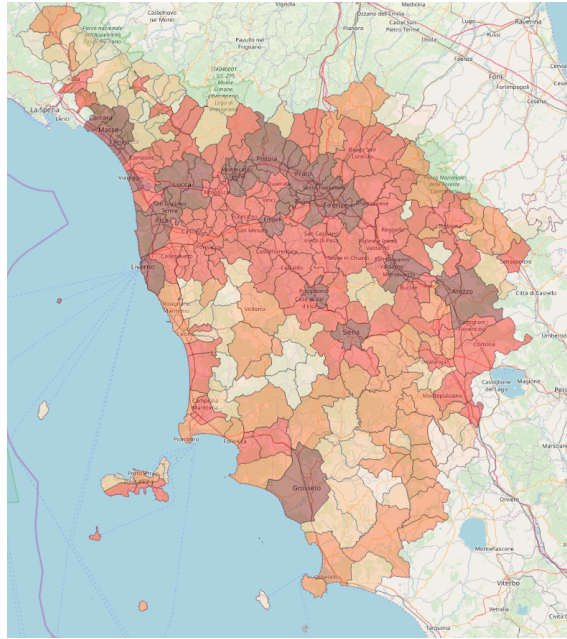


Figure 4.2.5: *Map of clustered municipalities*

In our case study, a small fraction of cities resulted to be disconnected from all the others in terms of flows, thus making them outliers w.r.t. the global features (e.g., the assortativity measure is null). Therefore, we decided to put them in a separate cluster, and apply the hierarchical clustering on the remaining ones. The results of applying the clustering to our dataset is shown in Figure 4.2.4 as a dendrogram of the hierarchical clusters found (notice that the dendrogram is truncated, in order to show only the last 12 aggregations. Based on the gaps between splits/merge points in the dendrogram, the aggregation is stopped at distance 4.0, yielding four clusters. To these, we add another cluster (id 0) containing the isolated cities. A summary of clusters' size is in Table 4.2.1.

An analysis of the properties of each cluster reveals that they may be distinguished based on the kind of traffic flows they involve. Also, clusters are depicted on the map in Figure 4.2.5. Cluster 0 was named *Disconnected*, since it is composed by the nodes not connected in the inter-city flows network. These municipalities also have a low entropy and low Moran's I score, meaning a not significant pattern of traffic, and most of them are located at the boundary of Tuscany and in the country-side areas, where there is a lower concentration of roads. Cluster 1, named *Self Sufficient*, is characterized by high entropy, high modularity and high fraction of regular trips, yet a low radius of gyration and low diameter of the associated ego networks. Also, they are mostly far from the highways that cross the region. Cluster 2, called *Visited Sites*, have a very low entropy (almost as low as those in the disconnected group), low modularity and the lowest fraction of regular trips, and yet a relatively high betweenness. Cluster 3 was named *Drive Through*, as these cities are crossed by a great flow of traffic, which is however basically coming from outside or going outside. Indeed, they have high values for entropy and low values

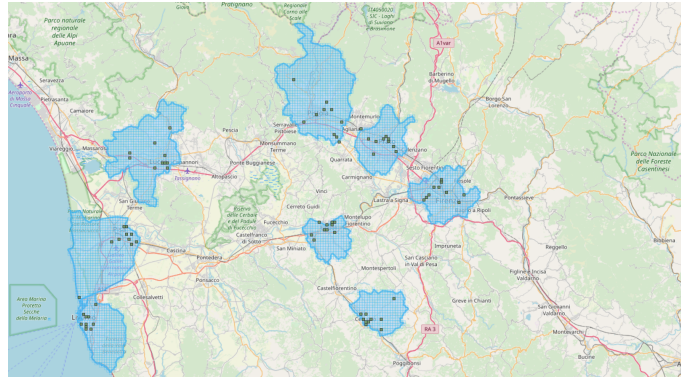


Figure 4.2.6: *Selected cells for some municipalities.*

for Moran’s I, the highest number of nodes regularly visited from users and a large ego network radius. This cluster is the most populated, comprising almost 40% of the dataset. Finally, cluster 4 was called *Hubs*, since it comprises all the biggest cities, encompassing most of the busiest roads in Tuscany. Municipalities are pretty similar to those belonging to cluster 3, excepted that they have a large Moran’s I, which reflects the presence of specific patterns within the city.

### Traffic Forecasting in City Grids

Urban traffic prediction is a discipline that aims to exploit ML models to capture hidden traffic characteristics from substantial historical mobility data, making then use of trained models to predict traffic conditions in the future [187]. However, there is a main problem to face: is it possible to extract specific traffic patterns that reflect the peculiarities of a city structure?

\*A Grid to Split the City Following one of the most used approach in traffic prediction problems [187], we divide every geographical area corresponding to municipalities in adjacent squared cells having side of 0.5 km, and our predictive objective is to forecast the traffic flow that crosses a given cell. In our case study we select a subset of representative cells and, in order to avoid the possible issues emerging when a random or top-frequency subset is selected, we adopt a mixed approach, randomly selecting 5 cells among those having a traffic volume above the 90<sup>th</sup> percentile over the municipality, and other 5 cells among those having a traffic volume between the 80<sup>th</sup> and the 90<sup>th</sup> percentiles.

### Time Series Preprocessing

Based on the trajectories that cross the representative cells identified above, we compute a time series for each cell with a 1-hour sampling rate, by counting the number of vehicles that crossed the cell within each hour of each day. A first operation performed was to compute a *moving-average smoothing* of the time series, since a preliminary test with the *Augmented Dickey-Fuller* test (ADF) [102] reveals that they are not stationary, i.e. it could not be rejected the null hypothesis that

a *unit root* is present in the time series sample (ADF=-2.38 against a critical 90% threshold at -2.57). On the contrary, after smoothing, the null hypothesis is rejected with a very large confidence (ADF=-5.57 against a 99% threshold at -3.43, p-value =  $2 \cdot 10^{-5}$ ).

### Predictive Features

Similarly to what done by several time series forecasting solutions [91], we base our predictions for the next value of the time series on more recent observations of the same time series. In particular, we adopt as basic features the 24 most recent lagged values, i.e., the observations of the last 24 hours. We remark that in this simplified approach we do not include features about other time series in the same municipality, as done in more complex solutions that exploit the spatial autocorrelation of this kind of phenomena.

Another important property that can be encoded is related to the weekday; at this regard, we introduce the boolean feature *is\_weekend* that is true if the weekday is Saturday or Sunday and false otherwise, since we expect to see different behaviors in the weekends. Finally, we can encode information about a weekday by inserting the average traffic volume at that day.

Having a total of 26 new features, we can now try to forecast the smoothed time series.

### Predictive Model

As regressive model, we selected the popular and effective algorithm *XGBoost* [65]. XGBoost has proved to be highly reliable in regression tasks, providing in general a good accuracy of predictions and remarkable speed of execution, yielding good results in term of robustness with its default settings, which simplifies our task. XGBoost adopts a Boosting procedure, i.e., is a ML ensemble meta-algorithm for primarily reducing bias and variance in supervised learning, where a set of weak learners is turned into a single strong learner.

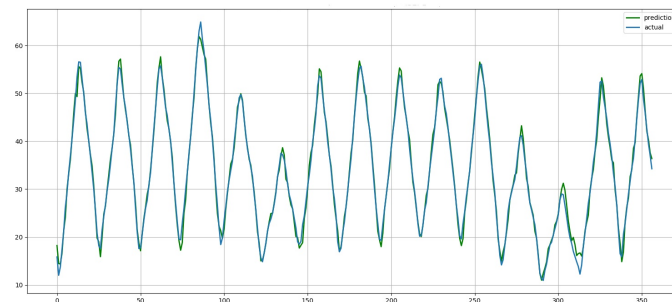


Figure 4.2.7: *XGBoost* traffic forecasting on Florence (green) against real values (blue).

In Figure 4.2.7 we can see an example of XGBoost predictions exploiting the



features previously introduced over the municipality of Florence, which shows results very close to the real values. The model performance is evaluated through the standard *Normalized Root Mean Squared Error*, defined as  $\text{RMSE} = \sqrt{\frac{\sum_{t=1}^T (\hat{y}_t - y_t)^2}{T}}$ , having predicted values  $\hat{y}_t$  for times  $t$  of a regression's dependent variable  $y_t$ , with variables observed over  $T$  times. RMSE is always non-negative, the lower is the value the better are the predictions. Since RMSE is scale-dependent, we adopt the Normalized RMSE (NRMSE), computed as:  $\text{NRMSE} = \frac{\text{RMSE}}{\sigma}$ , where  $\sigma$  is the standard deviation of the observed values.

Empirical evaluation shows that the most important feature is the value of traffic 1 hour before, as expected, while the previous hours have all a comparable influence. Instead, it is apparently almost irrelevant to know if a day is a week-end day or not.

### 4.2.9 Testing Model Transferability

In this section we study the transferability of the predictive models built above, and its relation with the similarity groups found through clustering. The hypothesis we want to test is that the similarity based on our city indicators is indeed useful to identify groups of areas such that any model built from an area in the cluster is usable in other areas within the same cluster. The first step is to split the traffic time series of each city in training and test sets. In this way it is possible to obtain a matrix of prediction scores where on the rows there are the cities in which the model is trained and in the columns those where the model is tested. The algorithm implemented iteratively trains a model on each city, tests it against all the cities and fills the score matrix with the corresponding NRMSE score obtained. To enable a more meaningful comparison, NRMSE scores are *log*-transformed to reduce the skewness.

The final result is visually shown Figure 4.2.8 which shows the transfer scores by sorting the cities based on their cluster belonging. Keeping in mind that the squares around the diagonal represent training and testing on cities of the same cluster, while the other rectangles depict training and testing on different clusters, we can observe:

1. the transfer is far better between cities of the same cluster (the NRMSE values are lower);
2. it is worth noting that also cluster 0, that we built up artificially behave exactly as the others;
3. the matrix is not symmetric: training on city  $A$  and testing on  $B$  is different from training on  $B$  and testing on  $A$ .

The trend noticed in Figure 4.2.8 can be better identified by computing the average error among the clusters, i.e., considering all the possible *source* areas in each cluster (where the models are built) and all the possible *target* areas in each other cluster (where the model is tested), including the case source = target. This is shown in





Figure 4.2.8: *Transfer scores matrix with cluster separation.*

Figure 4.2.9, where each bar corresponds to one of the *rectangles* outlined in red in Figure 4.2.8. We observe that the lowest mean values are always those corresponding to central squares, where the source and the target cities are from the same cluster.

## Conclusions

In this work we have defined a large array of local and global city indicators, we have calculated them on a real case study, and we have proved that they can be successfully exploited in a task of mobility transfer learning. In particular, we have clustered municipalities based on the mobility behavior described by the city indicators. Then, we have assessed the transferability of a machine learning model for traffic forecasting. Experimental results show that models trained on a municipality perform markedly better when tested on other municipalities belonging to the same cluster, and thus more similar (according to the city indicators) to the first one.

As future work, it would be interesting to extend the set of features used to describe a city, for example including census and cartographic data or some indicators related to economy, industry level and information about the most florid commercial

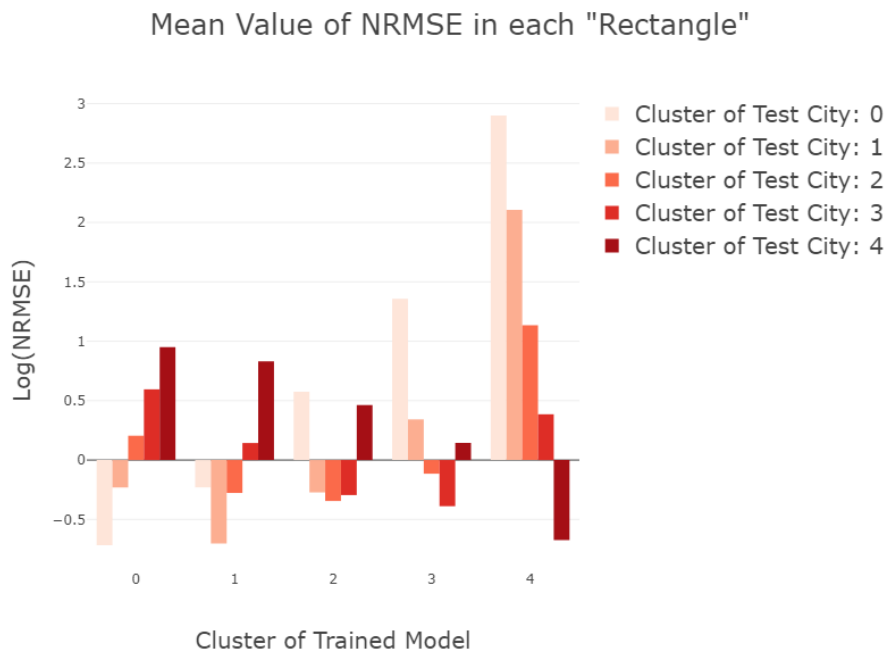


Figure 4.2.9: *NRMSE* mean values for all train-test pairs.

activities in each area. All these extra properties would also help to interpret the results of clustering, to identify patterns of similarity and eventually to supervise with some kind of feedback the allocation of a city to a determinate group. More models should be analyzed and compared to evaluate which is the most effective. Finally, the approach presented here works on a city-to-city transfer, namely the model of a single city is used to make prediction on the destination city. That assumes that there exists at least one origin city that is similar enough to perform the transfer. Alternatively, all the data and known city models can be exploited to achieve better prediction on the target city.

## 4.3 Car Crash Prediction

Collecting and processing mobility data is a fundamental task of car telematics and (modern) car insurance companies. Their main objective in doing that is typically to provide to end-users services like pay-as-you-drive contracts, anti-theft control, and prompt emergency rescue in case of accidents [191]. One of their foremost priorities, however, is to adapt policy pricing to customers in the best way, which mainly consists in finding a trade-off between profit and competitiveness. In this context, risk assessment is probably the most critical problem addressed. The risk from the company perspective can involve several aspects, yet the most impactful one is the customer's risk of having accidents in the future [322] since high-risk ones are likely to cause the company a loss (paying the costs of her accidents), while low-risk ones are more likely to provide a plain profit. In this context, since the car insurance markets are quickly expanding also towards new (for the market) geographical areas, there is the need to establish services in areas where very little or no prior knowledge at all is available, making the risk assessment task even more challenging.

Along the lines mentioned above, our research pursues two distinct objectives.

First, develop a methodology for predicting the customer's risk score: given a car insurance customer, provide a risk score relative to the long-term future, e.g., the next month or the next year. Since this estimate is expected to depend both on how the customer drives and on the conditions of the surrounding environment [172, 20, 77], we adopt an approach based on the computation of individual driving features, describing how much the user drives and how much dynamically, also related to the general characteristics of mobility in the places that the user visits. Since the raw mobility data collected by car telematics and car insurance companies is typically limited to positions and events of the vehicle [191], with no vision of what happens around it, our approach elaborates the data to infer higher-level knowledge, such as driving behaviours (frequent accelerations, average speed, etc.), individual mobility demand (detecting frequent trips, travel times during the day, etc.), habit changes, etc. [126]. That is achieved, in particular, by exploiting the Individual Mobility Networks (IMNs), described in detail in Sec3.2, that are a network-based representation that integrates important locations, movements, and their temporal dimension in a succinct way. Therefore, the proposed approach takes into account several different aspects: *individual* components of the driving behavior including those that can be derived from IMNs, elements considering the *collective* mobility of other users, and static *contextual* information such as road categories and the presence of points of interest. In this context, it is also important to identify possible risk mitigation strategies, namely to identify the characteristics of a driver labeled as risky that determine her risk score, since they could provide to the user indications of how to lower down her risk score, with benefits for her (in terms of safety and insurance costs) and the insurance company (in terms of costs for accidents).

The second objective is to enable the geographical transfer of crash prediction

models, i.e. to make the customer’s risk score prediction system usable and effective also on areas where historical data about crashes is unavailable or too limited. Given an area where we want to assess the customers’ risk scores and yet there is not a local training dataset to learn from, we derive a prediction model through techniques for *geographical transfer learning* which exploit the models and data available in other areas, in particular those similar to the one analyzed [211]. We define an array of geographical transfer learning strategies based on the data and the models available in certain areas that can be applied to target areas individually or as an ensemble. In particular, we rely on a set of city indicators (explained in detailed in Sec 4.2) that can be retrieved for every area to evaluate the similarity between two or more areas. The measures considered covers a wide spectrum of features, thus providing a multi-perspective description of area. They include a set of spatial concentration indexes of human activities; network features of intra-city traffic flows; mobility characteristics of the individual mobility, obtained from networks that represent the places and movement of single users; last, characteristics of road networks and how traffic is distributed in them. The city indicators allow to compare the different areas, using this similarity measure as a way to properly weight the contribution that each source area (i.e. areas where data are available and local models could be built) should give to the construction of a predictive model for the target area (i.e. the one where no data for training a model is available). We tried several different strategies that exploit such weights in different ways, and provides an empirical comparison to find out the best one in terms of prediction performances. When comparing models, performances are an important aspect to consider, but not the only one. Indeed, two models might have a similar accuracy, and yet implement completely different logics, for instance considering completely disjoint subsets of features. In the experimental section of this work we aim to understand in depth in what aspects the different models actually differ, and we realize that through the adoption of explainable AI approaches. That allows us to provide some hints about the reasons why the transfer of the models trained on certain areas and applied to a certain target area works better than in other cases.

We evaluate the proposed methodology on three datasets of real cars moving in three different areas, namely two cities (Rome and London), and one region (Tuscany, Italy). In particular, a deep study on the models’ transferability is performed on the Tuscany dataset working at the province level, which provided a good variability of city contexts yet involving areas of comparable complexity. The results show that the individual mobility-based and context-aware modelling of the users that we propose improves the performance over the baselines that adopt state-of-art features. Also, the analysis of predictions with the SHAP explanation methods [194] reveals that, indeed, most of the main factors that lead the models to decide for the riskiness of users belong to the newly introduced features. Finally, we observe that the best results in geographical transfer learning are obtained by the solutions based on the city indicators for training the most adequate classifier in a certain area. The explanation of these transferred models with SHAP reveals that the most important aspects for the crash prediction on the transfers are related to events that happens

while driving towards regularly visited locations such as harsh accelerations or harsh cornerings.

### Related Work

In this section we report an overview of the most relevant works related to the two research areas involved in this paper: crash prediction and transfer learning.

**Crash Prediction.** The literature on crash prediction is relatively large, studying car accidents from various perspectives, such as the risk of roads, the failure of safety devices or drivers' lack of attention. Yet, at the time of writing there are no works trying to exploit mobility data analysis and user modeling for crash prediction and risk assessment, with the only exception of [126]. A large part of the works focuses on real-time prediction of individual crashes, i.e., try to identify the events that lead to a crash in the next few seconds, thus providing feedbacks to the user as she drives [317]. Similarly, [269] developed a model for real-time collision detection at road intersections by mining collision patterns, while [20], using different data, tries to relate crashes to both behavioral characteristics and physiological parameters. Other approaches (e.g., [172, 3, 201]) work on identifying areas that show characteristics usually associated with accidents, such as increased traffic density, adverse weather conditions, etc. Besides features describing areas, the work in [167] also used individual vehicular data of cars (speed and time headway) passing through predefined detector stations for improving the performance of a probabilistic model. In [192] it is presented a review of the key issues associated with crash-frequency data as well as strengths and weaknesses of similar methodological approaches. While extremely useful, such approaches result in being not applicable to fields like car insurance, where the focus is in creating a general risk profile of the user, thus implicitly involving the prediction of her crash risk in the long run, such as few months in the future. Only a few, preliminary works are available in this direction. The most significant one is [322], which applies machine learning methods to predict the users' driving behaviors, based on movement statistics. In particular, the authors extend the standard approaches, which consisted in global aggregates of speed and mileage information, by separating daytime and nighttime driving statistics, and computing minimum, maximum and average aggregates. This increased detail of aggregation was shown to improve performances over simpler statistics. The work in [126], which provides the starting point of our work, further develops the general idea, and designs a data-driven model for predicting car drivers' risk of experiencing a crash based on the Individual Mobility Network model of the user and on statistical features which describe her driving characteristics. Here we extend the work and results of [126] with additional experimental studies and by boosting the crash prediction model with geographical transfer learning.

**Geographical Transfer Learning.** Individual mobility models and crash predictors, which are the basis of our proposed approach, are expected to strongly depend on the specific geographical area under study. For instance, it has been empirically verified that the trip purpose classifiers in [257] work very well in the geographical area where they were extracted, but their performances dramatically

degrade if applied to areas with different characteristics. Since some geographical areas could be insufficiently covered by data, due to the non-homogeneous penetration of tracking devices, it would be very difficult to build different models for different areas from scratch. A possible approach to the problem, then, is given by methodologies that make it possible to adapt models built in data-rich areas to less rich ones, which is basically a geographical instance of the general transfer learning problem [225, 345]. The transfer learning research area aims to transfer the knowledge available in one domain, called the *source domain*, to another one, called the *target domain* [227]. We refer to the particular case where the different domains are actually different geographical areas as *geographical transfer learning*. This specific topic is studied only sparsely in the literature, usually with objectives rather different from ours. The most common problem considered is image recognition, typically satellite image labeling, as in [32] and [290]. Both papers deal with deep learning classifiers that are requested to work on data-poor areas, and therefore the models learned in data-rich areas (usually CNN-based models) are adapted to the new domain. The authors of [24] focus on crime prediction and, again, try to exploit the knowledge available in some areas to make reliable predictions on a different one having too little data to build a model. Finally, [188] builds shared bike demand prediction models over some cities (especially large ones, where more data is generally available) and then adapt them to other (usually smaller) ones. The work in [147] shares some ideas with ours since it tackles the problem of labeling road networks and shows how assessing the similarity of street networks improves the transfer of a model from one city to another one. Our work tackles a more complex prediction problem, and compares areas through a multi-dimensional view, yet our results confirm the general message of the cited paper. The methods we propose start from the city indicators work (see Sec 4.2 and [211], which exploited a set of descriptive features of cities to assess their similarities, studying whether the transfer of models across cities works better among similar ones. Both the prediction problems tackled and the model transfer method adopted were very simple. In this work, we expand those results considerably, considering a complex crash prediction problem and developing several more sophisticated model transfer strategies, yet still, exploit city similarities.

### Problem Formulation

We define the *crash prediction problem* as the association of a user’s probability of having an accident in the next time period with their recent historical mobility. The duration of the user’s history to consider and of the next time period for which we make predictions are two fixed parameters. Reasonable durations for the context at hand will have the scale of one or more months.

**Definition 4.3.1** (Crash Prediction and Risk Assessment). *Given the prediction time  $\tau_p$ , history depth  $\tau_h$  and prediction span  $\tau_s$ , we define the two time intervals  $\bar{z}_p = [\tau_p - \tau_h, \tau_p]$ , named predictors interval, and  $\bar{z}_t = (\tau_p, \tau_p + \tau_s]$ , named target interval. Then, the crash prediction problem consists in evaluating if user  $u$  will have*

a car crash during period  $\bar{z}_t$  and what is the crash probability, based on the analysis of the user's mobility during period  $\bar{z}_p$ . More formally, we want to estimate:

$$p_{crash}(u) = P(u \text{ has crash in } \bar{z}_t \mid H_u^{\bar{z}_p})$$

The period  $\bar{z}_p$  is the knowledge we have about the user at the moment of assessing her risk, while  $\bar{z}_t$  is redwhere/the period when the crash to predict will or will not happen.

In a geographical transfer learning context, crash prediction has the same overall objective, yet the available information for estimating  $p_{crash}$  mainly comes from areas that are different from that of the user.

**Definition 4.3.2** (Geographically Transferred Crash Prediction). *Given a set  $A = \{A_1, \dots, A_n\}$  of  $n$  geographical areas, each associated to a set  $U_i$  of users, to a function  $\pi^{(i)}$  that estimates  $p_{crash}$  within  $A_i$  ( $1 \leq i \leq n$ ), and to the training set  $H_u^{train}$  of each user used to infer  $\pi^{(i)}$  ( $u \in U_i, 1 \leq i \leq n$ ); the predictors and target intervals  $\bar{z}_p$  and  $\bar{z}_t$ ; and an area  $A^* \notin A$ , associated to a set  $U^*$  of users; the geographically transferred crash prediction problem consists in computing the function  $\pi^*$  estimating the crash risk probability for each user  $u \in U^*$ :*

$$\pi^*(u) = P(u \text{ has crash in } \bar{z}_t \mid H_u^{\bar{z}_p}, \{\pi^{(i)}\}_{1 \leq i \leq n}, \{H_v^{train}\}_{v \in U_i, 1 \leq i \leq n})$$

The definition emphasizes the fact that the crash prediction function can use both the training data and the locally inferred models of the geographical areas in  $A$ , while for the area  $A^*$  we do not have access to a training dataset, the only information available being the data of the user in the predictors interval  $H_u^{\bar{z}_p}$  ( $u \in U^*$ ). Also, while it is in general possible that a user  $u$  belongs to two or more different areas, in the rest of the paper we will assume for simplicity that  $\forall i. U^* \cap U_i = \emptyset$ , i.e. the users in the target area are completely disjoint from those in the source ones.

**Features Importance-based Explanations** Given a machine learning classifier  $b$  trained on a dataset  $X$ , a feature importance-based explanation method takes as input  $b$ ,  $X$ , an instance  $x$  for which we want to explain the decision  $b(x)$  taken by  $b$  on  $x$ , and returns for each feature an importance value which represents how much that particular feature was important for the prediction of that instance. For understanding the contribution of each feature, the sign and the magnitude of each value are considered. A positive value means that a feature contributes negatively for the outcome; otherwise, the feature contributes positively. The magnitude, instead, represents how great the contribution of the feature is to the final prediction. *SHAP*, SHapley Additive exPlanations [194], is a local-agnostic explanation method that calculates feature importance based on the Shapley values, a concept from cooperative game theory. In particular, the explanation returned SHAP are *additive feature attributions* and guarantee the fact that the sum of all the contributions corresponds with the deviation of the prediction of a certain outcome with the baseline prediction, i.e., the average prediction among the instances in the training set.

## Methodology

In this section we present the methodology proposed in [126] for long-term crash risk prediction based on IMNs. Finally, we design a set of novel strategies for the geographical transfer of crash prediction models across different areas.

### 4.3.1 IMN-based Crash Risk Prediction

Our objective is to estimate the probability  $p_{crash}(u)$  in the crash prediction problem definition. In this section we do that through approximation, along two steps: (i) first, the knowledge contained in  $H_u^{\bar{z}_p}$  is represented through a set of meaningful yet (necessarily) lossy features, that will be discussed in details in the next sections; then, (ii) the probability function is learned through machine learning predictors.

**Predictive Features** Each user  $u$  is represented by a vector of  $m$  features computed over her predictors interval, namely:  $x_u^{\bar{z}_p} = \langle f_1, f_2, \dots, f_m \rangle$ . We denote with  $X^{\bar{z}_p} = \langle x_1^{\bar{z}_p}, x_2^{\bar{z}_p}, \dots, x_n^{\bar{z}_p} \rangle$  the matrix of  $n$  vectors describing the behavior of  $n$  users. We indicate with  $y^{\bar{z}_t}$  the vector saying if a user has experienced a crash in the target interval  $\bar{z}_t$ , i.e.,  $y_u^{\bar{z}_t} = 1$  if user  $u$  had a car crash in period  $\bar{z}_t$ ,  $y_u^{\bar{z}_t} = 0$  otherwise.

**Machine Learning Models** The matrix of features  $X^{\bar{z}_p}$  and the vector of target values  $y^{\bar{z}_t}$  are used to train a machine learning classifier, which yields as output a car crash predictor function  $p_{crash}(\cdot)$ . The crash predictor takes as input a vector  $x_u^{\bar{z}_p}$ , describing user  $u$ 's mobility in a given predictors interval  $\bar{z}_p$ , and returns the probability she will have a crash in the corresponding target period  $\bar{z}_t$ , based on the training performed on  $X^{\bar{z}_p}$  and  $y^{\bar{z}_t}$ . As machine learning classifiers [293] we considered several possible options, including K-Nearest-Neighbors, Decision Trees, Support Vector Machines, Deep Neural Networks, Random Forests, LightGBM, etc. Indeed, any prediction model working on standard tabular data could be in principle applied, since the specificities of the data domain are already captured by the user's features  $x_u^{\bar{z}_p}$ . Through preliminary experiments, we decided to mainly focus on Random Forest (RF), Deep Neural Network (DNN), and LightGBM (LGBM), since they yielded the best and most stable results. The case studies in Section 4.3.2 are based on these models.

A secondary (yet very relevant) objective of our work is to find the possible factors that lead to a crash, whatever the nature of each factor, either causal or simply correlated. In order to achieve that, we adopt three ways to infer the role played by each feature in the classification. The first one comes as a built-in feature of RFs, namely the *feature importance* score, which says how much important is overall a feature, though not describing if that is a positive or negative factor. The second way exploits recent results in the explainable AI domain, in particular, the SHAP method [194], which assigns the positive/negative impact of each feature on every single prediction allowing to make both single-user and collective considerations. The third approach consists in aggregating the absolute SHAP values of different



predictive models, in order to compare them and get a glimpse of their differences in terms of *logics* followed, in addition to performances.

**Predictive Features** A key component of the proposed approach consists in translating the raw mobility information contained in  $H_u^{\bar{z}^p}$  into a set of features  $\langle f_1, \dots, f_m \rangle$  able to capture its significant elements, and in particular, those useful for crash prediction. The following were computed:

- *Trajectory-based features.* These features include *position-based features*, containing classic indicators of trajectories, i.e., number of trajectories, length, duration, speed. Each indicator is aggregated through four operators: counts, sums, means, and standard deviations. Moreover, aggregates are computed over several time periods: morning (6am - 12am of all days), afternoon (12am - 6pm), evening (6pm - 10pm), night (10pm - 6am). The same applies for *event-based features*, measuring characteristics of the acceleration- and direction-related events contained in the data.
- *IMN-based Mobility features.* These features adopt IMNs as higher level of aggregation of the user's mobility, to extract three different types of information: *(i)* the network properties of the IMN, *(ii)* mobility aggregates focused on high-frequency locations and movements, and *(iii)* temporal stability measures of the IMN. A not exhaustive list includes the number of locations, i.e., nodes in the IMN, the number of movements, i.e., edges, the average degree, the IMN density, etc. In addition, for every feature is reported the variation between consecutive time periods in which the IMN is calculated.
- *Mobility Context features.* These features estimate contextual indicators by extracting collective aggregates from the history of all users in the dataset. Information like the number of events, average speed, and acceleration statistics are computed on *geographical sections* (a partitioning of space obtained through a *quadtrees* structure derived from the distribution of Points-of-Interest on the territory, ref. [126], Section IV-E), and they are associated to the single user based on which sections they stopped in at least once, compute an average of each characteristic of the sections. A not exhaustive list includes indicators of other users with respect to the areas visited by the user described in terms of number of starting and stopping trajectories, average speed, average accelerations, number of different events, etc.

Details for each family of features are available in [126]. The features considered can be inferred from the basic information that any car telematics service is expected to provide, and in that sense provides a minimal solution that can be very easily adapted to work in different geographical areas. Where available, this set can be extended with other useful measures about details of accidents, physical features of roads (pavement quality, size, visibility, etc.), weather, and so on. Real applications that need to be fine-tuned over a specific geographical area could indeed benefit from other information layers that can be easily integrated into our solution as additional

features. Considering such extra layers and studying their impact, however, goes beyond the scope of this paper, and is left as interesting future work. Finally, we highlight that typically, state-of-art car crash approaches used in the insurance practice, are only based on trajectories and do not account for all the mobility aspects considered by our proposal.

### **Geographical Transfer of Crash Prediction Models**

The basic idea of transfer learning is that the phenomena we want to capture (and that determine the value of the target variable to predict) are inherently present in other datasets, although in different proportions and maybe in different shapes. Therefore, the problem is to understand which parts of the data (in our case, which geographical areas) are more likely to contain cues and information useful to capture relevant phenomena, and thus exploit them for predictions. Hence, our objective for geographical transfer learning in crash prediction, is to explore ways for exploiting all the knowledge available on areas different from the target one, i.e., the one where we need a predictive model. With respect to the categorization presented in Section 2.3, we design a geographical transfer learning which is homogeneous (the data and the prediction tasks in the source and target domains are of the same type), multi-source (in general, we have several geographical areas with data we can exploit in the transfer) and transductive (we assume that labeled data is available in meaningful quantities only in the source domains).

The solutions proposed in this work try to overcome some of the main issues highlighted in [126] (and further confirmed in our experiments in Section 4.3.2). First, blindly applying a model from one region to another does not consider at any level the differences that the two areas might have. In our context, for instance, the road conditions in one area might require a different driving style than another one (reflected in the accelerations and contextual features), or the city size and traffic might impact the routine behaviors of users. Second, adopting standard weighting schemata based on feature distribution is possible only if rather significant data is available for the target domain, although unlabeled, which can be difficult in practical applications. In particular, in our reference insurance case study, the data is always associated with labels (crash or no-crash), the problem being instead to reach in a geographical region a sufficient mass of historical data. Also, since in our experiments we study the transfer between areas in the same region (Tuscany), it resulted that the differences between the features distributions are in most cases not significant. Third, the empirical studies in [126] focused on rather large areas. This leads to building models that are more generic, and therefore might not be able to capture local behaviors of smaller locations.

In the following, we introduce a few solutions based on the following principles:

- a good prediction model for an area can profit from the information (data or models) coming from other areas, the main open question being how to account for the differences;

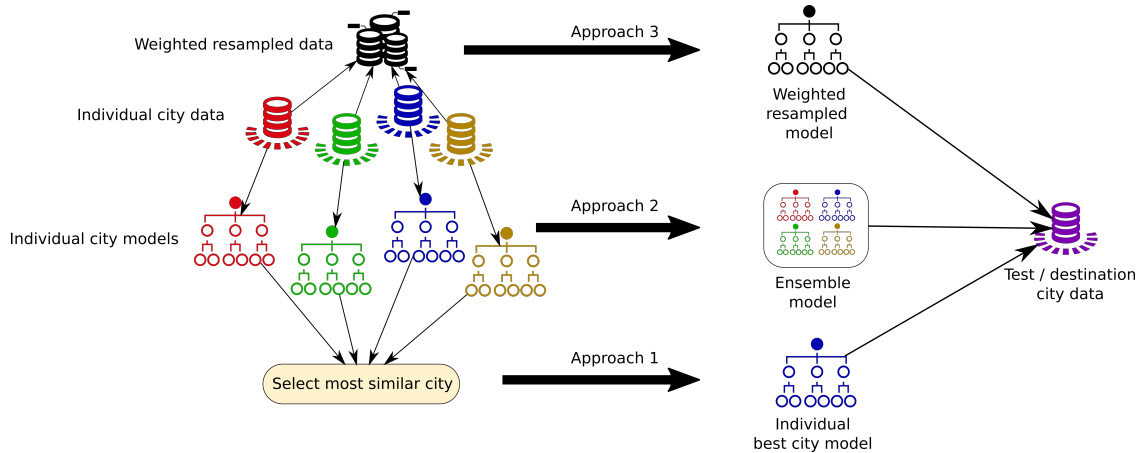


Figure 4.3.1: *Schema of the three geographical transfer learning approaches explored. The input city data is used either to extract individual city models (downward) or create a resampled dataset (upward). In the first case, Approach 1 selects the best model, while Approach 2 creates an ensemble. In the second case, a new model is built on the resampled data.*

- while each area might have its own local factors and patterns, driving and crash risk are expected to follow a common (potentially large and diversified) set of rules, although each area might adopt them in different proportions – total absence being mainly an exception;
- the factors behind the events to predict, i.e., crashes, are strongly linked with the mobility context where the users move, therefore the city indicators described in Section 4.2 should provide a good basis for understanding how much two areas share the same type of context.

Based on these principles, we propose three approaches of varying complexity that follow them at different extents. Each solution is described in detail below, while a schematic summary is provided in Figure 4.3.1.

**Approach 1: Best City Transfer** This is a direct application of the lessons learned in [126], namely that the model built on a city (or geographical unit) can be sometimes usable *as is* in another one, and that compliance is generally more likely to happen between cities that have similar spatial and mobility characteristics. Following this idea, Approach 1 selects among the source domains, i.e., the source cities where a model can be trained, the one that best matches the destination city in terms of city indicators, and applies its corresponding predictive model to the destination. With reference to Figure 4.3.1, the process starts from the individual city data, representing all possible source domains, over which we build individual city models. Finally, based on city indicators, we identify the source city that is most similar to the destination, and select its model. More formally:

$$p_{crash}^{best}(u) = p_k(u) \quad \text{with} \quad k = \arg \max_i sim(d, i) \quad (4.3.1)$$

where  $p_i(u)$  is the crash probability of user  $u$  estimated by the individual model of source city  $i$ , and  $sim(d, i)$  is the similarity between cities  $d$  (the destination) and  $i$  (the sources). More precisely,  $sim(d, i)$  is computed from the Euclidean distance between the corresponding (normalized) city indicators of  $d$  and  $i$ , i.e.:

$$sim(d, i) = EuclidDist(z\text{-score}(CI(A_d)), (z\text{-score}(CI(A_i))))^{-1} \quad (4.3.2)$$

where  $z\text{-score}$  computes the attribute-by-attribute normalization of the city indicators.

We name the model *individual best city model* (bottom line of Figure 4.3.1).

**Approach 2: Weighted Ensemble Model** It extends the ideas used in Approach 1, considering that each individual city dataset brings not only information that is specific for that location, but also information of more general validity, that might apply to all cities or at least to a subset. That means that each individual city model might, in principle, highlight a pattern or rule of general validity that, for statistical reasons or noise in data, could not be spotted in other cities. The idea is, therefore, to combine together the knowledge brought by all the individual models in an ensemble fashion, i.e., a meta-model is built by combination of the single ones, and predictions are performed by a voting schema where every single model provides a prediction, and the collection of results are combined. Since more similar cities are more likely to share common rules, the models in the ensemble can be associated with a weight corresponding to the city indicators-based similarity. Also, since our models provide a crash probability, the single predictions are combined through a weighted average. Formally:

$$p_{crash}^{ensemble}(u) = \sum_{i=1}^N w_i \cdot p_i(u) \quad \text{with} \quad w_i = \frac{sim(d, i)}{\sum_k sim(d, k)} \quad (4.3.3)$$

As before,  $sim(d, i)$  is the similarity between the destination city  $d$  and sources  $i$ , and  $p_i(u)$  is the crash probability of  $u$  estimated by the local model of source city  $i$ . In Figure 4.3.1 this corresponds to the central arrow, which yields the *weighted ensemble model* (or simply *ensemble model*, if clear from the context) that is then applied to the destination city data.

**Approach 3: Weighted Sampling** The ideas of the ensemble approach are applied here from a slightly different perspective. The ensemble model assumes that if the overall dataset contains a pattern or rule that is relevant for the destination city, then at least a subset of the individual models should be able to identify it, allowing the voting schema to bring it to the destination. However, this is expected to hold only for relatively strong rules, which can emerge from individual datasets, while that might not work for smaller patterns that leave many weak traces in the various datasets. Basically, the ensemble approach filters at the source weaker patterns, some of which might actually result to be significant overall. As possible counter-measure for this effect, Approach 3 creates an ensemble of datasets rather

than models, i.e., it builds a representative dataset by a weighted sampling of all individual datasets. This combined dataset, then, is used to build a predictive model. Since, again, we expect to find more useful information in source cities that are similar to the destination, the sampling weights are proportional to the city similarities. More formally:

$$p_{crash}^{resample}(u) = P\left(u \text{ has crash in } \bar{z}_t \mid H_u^{\bar{z}_p}, \{H_v^{train}\}_{v \in D}\right) \quad (4.3.4)$$

where  $D$  is the data sample built for destination  $d$  from sources  $A$ , and is defined as:

$$D = \bigcup_{A_i \in A} D_i \quad \text{with} \quad D_i \subseteq U_i \quad \text{s.t.} \quad |D_i| = N \cdot w_i \quad (4.3.5)$$

where  $U_i$  represents the set of users described in source city  $i$ , and  $N$  is the requested size of the sampled dataset, i.e.,  $N = |D|$ . Weights  $w_i$  are computed as for Approach 2. The more complex form of Equation 4.3.4 highlights the fact that this approach requires learning a model from scratch rather than simply combining or selecting existing local ones.

In relation to existing generic transfer learning solutions, the first two approaches presented above provide a form of *relational-based* transfer learning, since the models built in one domain are used (possibly adapted) in the other; the last approach, instead, works through an instance weighting strategy, which belongs to the category of *instance-based* transfer learning [227]. In particular, the latter is close in principle to Domain Weighting [35], yet it relies on a higher-level notion of city similarity, rather than a comparison of features distribution – which might be difficult to implement if only little (unlabeled) data is available in the target domain, as it is expected to happen in our application scenario. Also, as already mentioned, depending on the spatial granularity, in some cases the attribute distributions might not vary significantly across geographical units, thus making it a weak criterion. Indeed, preliminary tests on the datasets adopted in our experiments (see Section 4.3.2) showed that the feature distributions over the provinces were rather similar, being statistically not clearly distinguishable at the level of single features (around 58% of province-vs-province comparisons over all features did not pass the Kolmogorov-Smirnov rejection test [59] with threshold 0.05), and obtaining PCA projections over the two largest principal components having visually almost identical distributions.

### 4.3.2 Experiments

In this section, we present a case study on two datasets of private cars in which we employ the proposed methodology. We first introduce the datasets, and then summarize the results obtained on the crash prediction problem with and without geographical transfer learning, with a comparison between our solution and some baselines. We also extract explanations of the predictions returned by the various models, and try to infer useful general hints for improving personal driving behaviors.

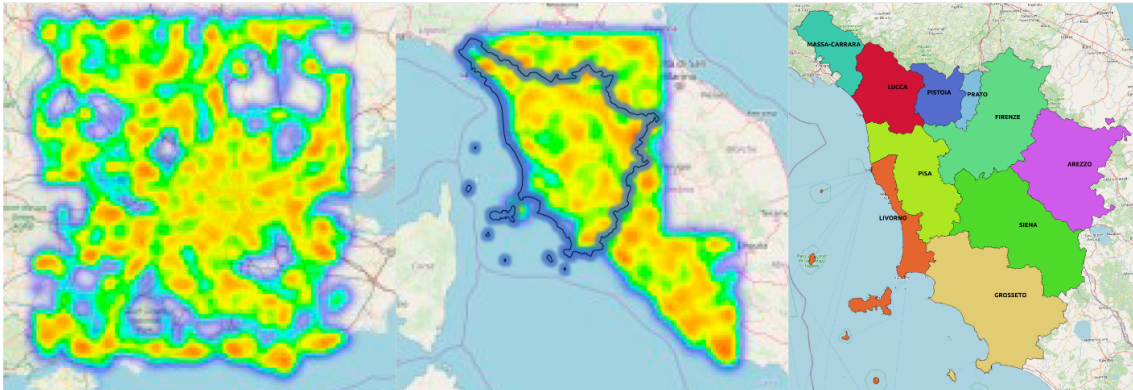


Figure 4.3.2: *Geographical areas of experiments. Dataset 1 includes London in UK (left), Tuscany and Rome in Italy (center). Dataset 2 is a zoom on the Tuscany area (highlighted in the center) by also considering its 10 provinces, shown on the right.*

### Dataset Description

The two datasets considered in our experiments consist of GPS traces of private vehicles tracked by an international car telematics company and made accessible to us within the Track & Know project[142]. The first dataset, named Dataset 1, includes London in UK (Figure 4.3.2 left), Tuscany and Rome in Italy (Figure 4.3.2 center), each area having about 5,000 drivers. The drivers were sampled among those that had consistent data throughout the 12 months, and also ensuring to keep all those that had at least one crash in the year. This latter step was not possible on Dataset 2, a side effect being that Dataset 1 has a higher percentage of crash events. The second dataset, named Dataset 2, includes about 26,000 drivers and it is a zoom on the Tuscany area (highlighted in Figure 4.3.2 in the center) by also considering its administrative division into 10 provinces (Figure 4.3.2 right). We consider the partitioning of the Tuscany region in subareas in order use them as source and destination domains for transfer learning experiments. Each subarea is populated with the data of users whose most frequent location is contained in that subarea. We decided to report results with respect to provinces because they provide a good trade-off between granularity and data availability on each partition. While testing model transfer across very different areas as Rome and London would be interesting, the different scale and complexity of these cities would require a more extensive dataset covering many other international cities, which was not possible in the scope of this work. In the rest of this section we will use the terms *city*, *geographical unit*, and *province* interchangeably, when there is no risk of confusion.

For both datasets, the raw mobility data consists of anonymized traces of vehicles of car insurance customers, containing the following information: (i) a list of GPS timestamped *positions* (latitude and longitude); (ii) a list of *events* in the form of timestamped position data enriched with labels describing events such as harsh acceleration, harsh braking and (possibly multiple) harsh cornering, with addi-

tional accelerometer metrics related to each event position. These data are collected whenever the accelerometer detects an acceleration exceeding predefined parameters; (iii) a list of *crashes* in form of timestamped position data related to crash events. Such events were originally detected through onboard accelerometers and filtering algorithms, and later checked by human operators with customers to remove false positives. The dataset is collected at an average rate of one position every 1.5 minutes, though there are some exceptions.

### Experimental Settings

We organize the experimentation as follows. We use Dataset 1 to analyze the performance of the models for the basic car crash prediction problem, focusing the attention on the effect of the various features described in Section 4.3.1 and on the temporal dimension. On the other hand, we rely on the greater data availability of Dataset 2 to address the geographically transferred crash prediction problem with the city indicators described in Section 4.2 through the transfer learning methodologies illustrated in Section 4.3.1.

**Local Crash Prediction.** In the experimental setting for Dataset 1 (*D1*), we consider different time periods, corresponding to prediction times  $\tau_p^1 = \text{end of March}, \dots, \tau_p^9 = \text{end of November}$ . The corresponding experiment periods  $\bar{z}_i$  are obtained by fixing the history depth  $\tau_h$  to 3 months (used to compute features) and prediction span to 1 month (the period where crashes are checked). We run the experiments in three different experimental settings, depending on how we consider the temporal and geographical components. In the first setting (*D1.1*) we keep separated each experiment period  $\bar{z}_i$  and each spatial region  $r$  ( $r \in \{\text{London, Rome, Tuscany}\}$ ) from all the others. In particular, for each given pair  $(\bar{z}_i, r)$  we train a classifier over the corresponding data of all the users in  $r$ , namely  $X^{\bar{z}_i, r}$  and  $y^{\bar{z}_i, r}$ , and then use the model to make predictions one month later, i.e., it is applied over  $X^{\bar{z}_{i+1}, r}$  and the results are compared against the ground truth in  $y^{\bar{z}_{i+1}, r}$ . Notice that we must have  $i + 1 \leq 9$ , therefore we obtain a total of  $|\{\tau_p^i\}| \times |\{r\}| = 24$  sets of experimental results. In the second setting (*D1.2*), we still keep regions separated, while all experiment periods are considered together. Users are split into a training set and a test set, following a hold-out division, all the 9 experiment periods of a user in the training set are used (as 9 separate records) in the model training and, similarly, all the 9 experiment periods of a user in the test set are used for the model testing. The main difference between the two settings is that in (*D1.1*) we check if we can predict the crash of observed users in the future using a limited amount of data, while in (*D1.2*) we try to predict the crash of unobserved users using a consistent amount of data but without a temporal reference. Finally, the third setting (*D1.3*) amplifies the effects obtained by (*D1.2*) by putting the users of different areas in a unique training dataset.

**Geographical Transfer Learning.** The experimental setting for Dataset 2 (*D2*) is organized similarly to (*D1.2*), i.e., geographical areas are kept separated, yet putting together all time periods. The main distinction is that now we have 10 areas corresponding to the provinces of Tuscany. In turn, each province will be

Table 4.3.1: *Datasets summary as average values of some features.*

	#users	% crash	#traj	#traj/day	#evnt	#evnt/day	#mov	#loc	degree
London	5k	1.08	280.54	3.39	2967	34.81	66.84	31.23	4.31
<i>D1</i> Rome	5k	2.82	307.48	3.13	2655	25.74	82.80	41.10	4.02
Tuscany	5k	3.12	327.11	3.28	3041	29.13	81.48	41.19	4.07
<i>D2</i> Tuscany	26.7k	0.84	375.41	3.92	1088	11.59	77.64	34.81	4.53

selected as *target domain*, while all the others are used as *source domains*, the task being to make predictions on the former using the models or data from the latter. The data related to each province is partitioned into a training and a test set, which are used to extract a local predictive model for each province, and then to test it on the other ones. The transfer learning approaches proposed will either select or combine such local models or build a training set by resampling the local training data, and then test the resulting model over the test partition of the province under analysis.

## Datasets Preparation

In both experimental settings, before training the classifiers, we face two problems with the datasets analyzed. The first one is a class imbalance issue. Indeed there is a very low number of crashes compared to the number of no crashes (see Table 4.3.1). We tackle this problem by adopting the SMOTE oversampling approach [62]. The minority class is over-sampled by taking minority class samples and introducing synthetic examples along the line joining the  $k_{SMOTE}$  minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the  $k_{SMOTE}$  nearest neighbors are randomly chosen. We adopt  $k_{SMOTE} = 5$  by default as suggested in [62]. The effect of adopting SMOTE is to improve class balance and to reinforce the presence of the minority class in the decision regions where it appears. We highlight that we re-balance only the training datasets and not the test ones making the evaluation *harder* but more realistic. The second problem is the high dimensionality of the datasets analyzed. Indeed, the rich data engineering described in the previous sections leads to the construction of more than 400 features, some of them being highly correlated and redundant. This high dimensionality can cause difficulties in the learning of classification models. Thus, we adopt a dimensionality reduction technique based on correlation analysis. We calculated the Pearson correlation coefficient [294] between every pair of features for the various settings. Then, we removed one attribute for each couple having a correlation higher than 0.85. This operation reduced the dimensionality of the datasets to 162 features, with a balanced presence of trajectory-based, event-based, IMN-based, and contextual features. Table 4.3.1 reports the per-user average values of a small sample of features.



## Machine Learning Models

Our crash prediction approach and our geographical transfer learning strategies can be in principle applied using any existing machine learning algorithm as an underlying predictive model. In this work, we consider three modern and powerful types of classifiers: Random Forests (RF, basically an ensemble of several small decision trees), LightGBM (LGBM, a decision tree algorithm based on gradient boosting, with an emphasis on scalability) and Neural Networks (NN, here used in the simple form of a multi-layer perceptron).

**Configuration details.** For LGBM we used the `lightgbm` library [1], while for NN we experimented with both the Keras and Scikit-Learn libraries [170]. Since the latter two libraries are applied to the same algorithm type (NN), and the models obtained with Keras yielded worse performances than Scikit-Learn, in the next sections we show only results for the latter. For all models we used the Randomized Search Cross Validation to select the best combination of parameters. The parameters of the estimator used to apply these methods are optimized by cross-validated search over parameter settings. For RF, we use the `RandomForestClassifier` that is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the "max samples" parameter if "bootstrap=True" (default), otherwise the whole dataset is used to build each tree. We try different settings to decide the number of trees in the forest (`'n_estimators'`: [8, 16, 32, 64, 128, 256, 512, 1024]), the minimum number of samples required to split an internal node and the minimum number of samples required to be at a leaf node (`'min_samples_split'`: [2, 0.002, 0.01, 0.05, 0.1, 0.2], `'min_samples_leaf'`: [1, 0.001, 0.01, 0.05, 0.1, 0.2]). Final setting we adopted is the following:

- `'number of estimators'`: 128,
- `'min samples split'`: 0.05,
- `'min samples leaf'`: 0.05,

For NN we use the `MLPClassifier`, a Multi-layer Perceptron classifier that optimizes the log-loss function using stochastic gradient descent. Also in this case we tried different settings in order to find the optimal hidden layer size and the learning rate. We tried the `'relu'`, `'tanh'` and `'logistic'` functions as activation ones and we made experiments to try all configurations: `'hidden layer sizes'`: [(64, 128), (128, 256), (512, 1024), (512, 1024, 256), (1025, 512, 256)]. After testing, the final setting we adopted is the following:

- `'hidden layer sizes'`: (128, 256),
- `'activation function'`: `'relu'`,
- `'learning rate'`: `'constant'`,

LightGBM is a gradient boosting framework that uses tree based learning algorithms. It has a high training speed and low memory usage. LightGBM uses the leaf-wise tree growth algorithm to get good results, and requires to select a few important parameters. The number of leaves (*num leaves*) is the main parameter to control the complexity of the tree model. Theoretically, we can set  $num\ leaves = 2^{max\_depth}$  to obtain the same number of leaves as a depth-wise tree. However, this simple conversion is not good in practice. We tried to use  $num\ leaves = (10, 31, 50)$  with a  $max\_depth = (-1, 2, 5, 10)$ . The best parameters setting found is the following:

- ‘number of leaves’: 31,
- ‘max depth’: 5,
- ‘boosting type’: ‘gbdt’,

About the Keras experiments, we use the same configurations of MLPC Classifier with the only addition of the dropout parameter that is used to regularize the neurons activation and selection during the training phase. For our experiments we set ‘dropout rate’=0.1.

## Evaluation Measures

Given the application context around this work, our objective is to highlight future risky and potentially harmful events, also with the aim of raising an alarm that might help to prevent them. From this perspective, false positives are less critical than false negatives. To this aim we use as main evaluation guidelines [294] the *recall* of the positive class ( $rec_1$ ), i.e., aiming to find as many risky drivers as possible, the *f1-measure*, i.e., the harmonic mean of precision and recall of the positive class weighted with respect to the number of crashes ( $f1_1$ ), and the *area under the roc curve* ( $auc$ ) of the positive class that is the area under the curve comparing the false positive rate ( $FPR$ ) and true positive rate ( $TPR$ ). All measures range from 0 to 1, the optimum being 1.

## Crash Prediction Evaluation

In this section, we evaluate the results for the experimental settings in *D1*. Among the various classifiers, we found out that *Random forests* (*RF*) overcome those of the other algorithms. Thus, in the following, we report the results obtained using RF classifiers. In particular, we used RF with 100 estimators, allowing leaves with at least 1% of the training data, and with a cost matrix weighting a crash 100 times more than a no crash. We show the effectiveness of RF using the sophisticated IMN-based and contextual features described in Section ?? by comparing against three alternatives. The first two are baselines: a *constant* classifier (*CST*) always returning the positive class (crash); a *random* classifier (*RND*), predicting uniformly randomly crash or no-crash.

Table 4.3.2: Performance for the experimental setting *D1*. For *D1.1* the metrics are aggregated in terms of means and standard deviation over different periods. The best performance results are those underlined.

<i>Model</i>	Rome			Tuscany			London			
	<i>rec</i> <sub>1</sub>	<i>f1</i> <sub>1</sub>	<i>auc</i>	<i>rec</i> <sub>1</sub>	<i>f1</i> <sub>1</sub>	<i>auc</i>	<i>rec</i> <sub>1</sub>	<i>f1</i> <sub>1</sub>	<i>auc</i>	
<i>D1.1</i>	CST	<u>1.00±.00</u>	.024±.01	.500±.00	<u>1.00±.00</u>	.025±.01	.500±.00	<u>1.00±.00</u>	.009±.00	.500±.00
	RFI	.877±.10	.149±.08	<u>.588±.05</u>	.992±.01	<u>.056±.04</u>	.719±.05	.994±.01	<u>.574±.02</u>	<u>.962±.01</u>
	RFP	.891±.08	.140±.07	.574±.03	.992±.01	.042±.03	.577±.04	<u>.719±.10</u>	.308±.05	.612±.04
	RND	.486±.05	<u>.352±.02</u>	.500±.00	.488±.03	.355±.01	.500±.00	.499±.08	.341±.00	.500±.00
<i>D1.2</i>	CST	<u>1.00</u>	.028	.500	<u>1.00</u>	.029	.500	<u>1.00</u>	.010	.500
	RFI	.882	.216	<u>.619</u>	.944	.243	<u>.775</u>	<u>1.00</u>	.580	<u>.955</u>
	RFP	.866	.180	.586	.970	.061	.584	.624	.329	.574
	RND	.500	<u>.361</u>	.500	.480	<u>.355</u>	.500	.489	<u>.344</u>	.500
<i>D1.3</i>	<i>Model</i>	All								
		<i>rec</i> <sub>1</sub>	<i>f1</i> <sub>1</sub>	<i>auc</i>						
	CST	<u>1.00</u>	.022	.500						
	RFI	.991	.206	<u>.776</u>						
	RFP	.996	.025	.641						
RND	.485	<u>.352</u>	.500							

Their purpose is to provide reference performance values that can help interpreting the results of the other methods. The third one (RFP), instead, implements the approach in [322] by adopting an RF based on the features suggested in the state-of-the-art of crash prediction, including both those used in [322] (aggregates of speed and mileage, divided by night and day) and those suggested in previous works (e.g. statistics about accelerations [320], and harsh turns [159]). We name RFI the RF classifier that improves over RFP by extending the classical features used in literature with those we designed.

Table 4.3.2 reports the result for the experimental settings in *D1*, showing the evaluation measures returned by the classifiers for Rome, Tuscany, and London. Note that for the *D1.1* case the values are averaged among the various periods. The overall results we observe in the various experimental settings of *D1* are the following. The simultaneous analysis of the reported indicators shows that RFI provides the best and most reliable performances. Indeed, the CST baseline obviously has the highest recall but a zero precision on no crashes, making it useless for practical usage. On the other hand, RND easily gets a high *f1*<sub>1</sub>, thanks to the high imbalance of data, but it loses half of the real crashes, with a recall below 0.5. RFP gives a better trade-off than CST and RND for the *f1*<sub>1</sub>, but it shows an *auc* just slightly better than CST and RND, with a value around 0.6. On the other hand, RFI has always similar or larger *f1*<sub>1</sub> and recall compared to RFP, and it has systematically a higher *auc*<sup>3</sup>insert ý <sup>3</sup>An ablation study (omitted due to space limits) showed that both IMN- and context-based features significantly contributed to such performances..

In *D1.1* we observe different behaviors of RFI in the three areas considered. In

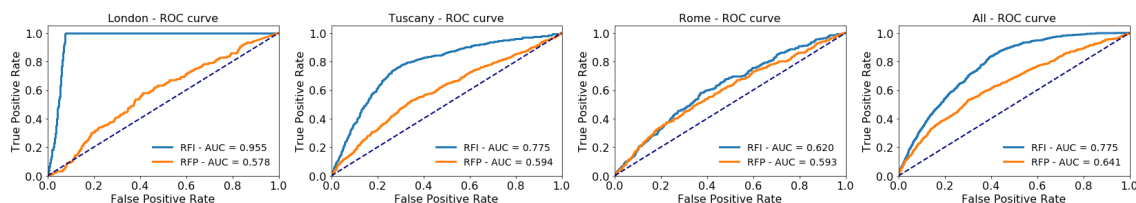


Figure 4.3.3: ROC curve for different areas for  $D1.2$  and  $D1.3$ .

London, RFI has the highest  $rec_1$ ,  $f1_1$ , and  $auc$ . Notice that the other methods considered show much worse results. In other words, the new features introduced in this paper appear to make crashes easy to predict in London. Understanding the reasons for this effect is part of our future works. For  $D1.2$  we observe how the increased number of available records for the training leads to a not negligible improvement in the performance of the classifiers in the Rome, Tuscany, and London areas when compared to those of  $D1.1$ . In addition, the setting  $D1.3$  that puts together records from all the different areas (“All” section in Table 4.3.2) leads to a classifier even better than those resulting from  $D1.2$ . We highlight in Figure 4.3.3 the Receiver Operating Characteristic (ROC) curve of the classifiers for the experimental settings  $D1.2$  and  $D1.3$ . These plots show that London classifiers are much more accurate than the others and that RFI classifiers markedly benefit from the usage of IMN-based and contextual features with respect to RFP, whose ROC curve is always below.

**Role of the features.** By exploiting the *feature importance* indexes of the models extracted it was possible to evaluate which features are more heavily used in making predictions. In general, the top ones involve driving events data jointly with the annotations inferred from IMNs: the number of starts in IMN locations labeled as occasional, the angle of accelerations around the most frequent locations, the radius of gyration of regular trips, etc. Then, various aggregations of simple driving features (duration of cornering events, standard deviation of speed, average speed during nighttime, etc.) as well as purely structural features of IMNs (betweenness coefficient of regular trips, centrality of second most frequent locations, etc). A more detailed evaluation can be found in [126].

**LSTM-based approaches** The key component of our approach that makes it superior to its closest competitor (RFP) is its extensive set of carefully engineered features, which are the result of a long experience in mobility analytics and driving behavior modeling. However, recent works in machine learning show that deep learning solutions are able to skip the human-made features construction phase in many tasks, and autonomously learn effective data representations directly from raw data, achieving exceptionally good performances. It is, therefore, natural to wonder if that can be the case also in the complex scenario we are considering. Along this line, we tested an alternative approach to our problem-based deep learning. In particular, we model the user’s mobility as time series of basic mobility indicators,

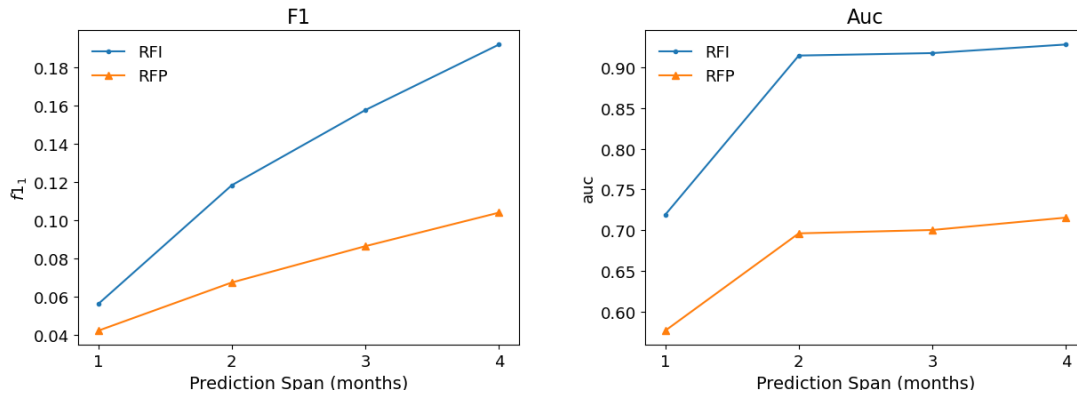


Figure 4.3.4: *F1 score and auc for the RFI and RFP approaches on the Tuscany dataset by varying the prediction span from 1 month to 4 months.*

namely: maximum speed, distance covered, driving time and average trip duration.

Then, we apply an LSTM network to learn the association between such time series (containing the values in the 3-month predictors intervals) and the target variable (crash / no-crash, observed in the 1-month target intervals, as for the previous experiments). The training and test data are partitioned exactly as in the experiments described above, and the time series has a 1-hour sampling rate. Experiments have been performed on Tuscany only since it is the richest dataset.

The network adopted follows the most commonly used structure for LSTM and time series classification: one LSTM level with 1024 units, followed by a drop-out of 0.5; then a dense layer with 256 nodes, followed by a drop-out of 0.2; finally, another dense layer with 64 nodes, and a drop-out of 0.01. In particular, the drop-out was necessary for the unbalance of the classes. A ReLu activation function was used in the internal layers, and a sigmoid function for the output. The training adopted an Adam optimizer with a binary cross-entropy loss function, using the area under the ROC curve (*auc*) as evaluation metrics. The misclassification weights were set to 0.5 for no-crashes and 95 for crashes, again due to the class unbalance. The preliminary results obtained, however, show rather poor performances. The *auc* has values close to random classification ( $0.5 \pm .008$ ), and the *f1* measure is significantly lower than those obtained with the other methods ( $0.01 \pm .005$ ). That is mainly caused by a low precision ( $0.005 \pm .003$ ), whereas the recall is relatively good ( $0.66 \pm .491$ ) yet rather unstable and lower than the other methods.

Our conclusions are, therefore, that the approach, although interesting and worth exploring, does not work well with the basic features and the standard setting adopted, and further investigations are needed. We point them out as possible future works..

### Testing longer prediction spans

An interesting aspect to study is whether predicting crashes over a longer time horizon is harder or actually simpler. Indeed, on the one hand we are trying to

Table 4.3.3: *Crash prediction performance for the various geographical units inside Tuscany in D2. Each model is trained and tested in the same area similarly to D1.2. The last line report the performance of a model trained and tested on the whole dataset similarly to D1.3.*

City	RF			NN			LGBM		
	$rec_1$	$f1_1$	$auc$	$rec_1$	$f1_1$	$auc$	$rec_1$	$f1_1$	$auc$
Arezzo	0.15	0.08	0.84	0.27	0.14	0.81	0.00	0.00	0.50
Florence	0.91	0.09	0.92	0.31	0.11	0.84	0.00	0.00	0.90
Grosseto	0.04	0.07	<u>0.94</u>	0.12	0.15	0.93	0.00	0.00	0.50
Livorno	0.83	0.10	0.90	0.00	0.00	<u>0.97</u>	0.00	0.00	0.50
Lucca	<u>0.98</u>	0.07	0.89	0.32	0.16	0.85	0.04	0.00	0.43
Massa	0.89	0.11	0.88	0.32	0.15	0.89	<u>0.95</u>	<u>0.09</u>	0.80
Pisa	0.53	0.12	0.92	0.31	0.26	0.85	0.00	0.00	0.10
Pistoia	0.31	0.06	0.83	0.40	0.07	0.85	0.00	0.00	0.50
Prato	0.35	<u>0.25</u>	0.91	<u>0.45</u>	0.21	0.94	0.00	0.00	<u>0.91</u>
Siena	0.36	0.20	0.86	0.36	<u>0.34</u>	0.93	0.00	0.00	0.50
All	0.44	0.12	0.91	0.34	0.11	0.96	0.46	0.07	0.83

infer events that are further in the future, and therefore harder to capture; on the other hand, since we are enlarging the prediction window, and not just moving the same window further, the number of positive cases we are considering in the training phase is bound to increase, making the problem less unbalanced. In order to understand what is the resulting trade-off, we repeated the experiments made on the Tuscany area by changing the prediction span, now ranging from 1 month (the value used in the previous experiments) to 4, and measuring the  $f1$  and  $auc$  scores. The results are plotted in Figure 4.3.4, where also the values obtained by our main competitor RFP are given. In both cases, we can observe that longer spans are overall better captured by our models, meaning that the class unbalance is a stronger factor of the problem. We see, in particular, that while the  $f1$  score grows at an almost constant rate, the  $auc$  quickly reaches a sort of plateau, meaning that the associated risk probabilities produced by the model form a significantly better sorting when passing from 1 month to 2, yet no large improvement is given by further extending the window to 3 and 4 months. Interestingly, RFP follows exactly the same behavior, yet with much worse performances.

### 4.3.3 Geographically Transferred Crash Prediction Evaluation

In this section we evaluate the three geographical transfer learning strategies proposed in Section 4.3.1 in the experimental setting (*D2*).

**Testing local models** First, we analyze the performances of local models built separately on each province, applying them to the test set of the same area, similarly

Table 4.3.4: Geographically transferred crash prediction *auc* for NN and RF. The best transfer are underlined, the transfer suggested by Approach 1 – Best City Transfer w.r.t the similarity of city indicators are in bold.

target → source ↓	NN <i>auc</i>										RF <i>auc</i>									
	Arezzo	Florence	Grosseto	Livorno	Lucca	Massa	Pisa	Pistoia	Prato	Siena	Arezzo	Florence	Grosseto	Livorno	Lucca	Massa	Pisa	Pistoia	Prato	Siena
Arezzo		.73	.73	<b>.80</b>	.81	<b>.81</b>	.73	.84	.80	<b>.79</b>		.82	.83	<b>.83</b>	.82	<b>.83</b>	.83	.82	.63	<b>.82</b>
Florence	.69		<u>.80</u>	.86	.89	.90	.85	.93	.90	.86	.93		.92	<u>.93</u>	.92	.92	.93	.92	.57	.83
Grosseto	.65	.87		.92	.82	.88	<b>.81</b>	.91	.90	<u>.90</u>	.92	.93		.92	.92	.93	<b>.91</b>	.90	.55	.84
Livorno	<b>.57</b>	<u>.93</u>	.78		<u>.92</u>	.93	.90	<u>.97</u>	<u>.97</u>	.89	<b>.96</b>	<u>.97</u>	<u>.96</u>		<u>.97</u>	<u>.97</u>	<u>.97</u>	<u>.98</u>	.49	<u>.98</u>
Lucca	.70	.72	.72	.87		.86	.74	<b>.89</b>	.85	.83	.88	.87	.89	.88		.87	.86	<b>.87</b>	.66	.86
Massa	.58	.78	.64	.80	.86		.81	.87	.83	.77	.88	.87	.87	.89	.89		.85	.86	.68	.74
Pisa	.46	.81	.71	.86	.89	.87		.88	.88	.83	.91	.90	.91	.91	.89	.89		.89	<u>.68</u>	.79
Pistoia	.62	<b>.63</b>	.60	.79	<b>.82</b>	.83	.79		<b>.82</b>	.80	.84	<b>.86</b>	.84	.86	<b>.86</b>	.86	.83		<b>.66</b>	.80
Prato	.69	.86	.69	<u>.93</u>	.85	<u>.91</u>	.73	.89		.85	.91	.90	.91	.92	.93	.89	.91	.91		.84
Siena	<u>.74</u>	.73	<b>.59</b>	.86	.90	.89	<u>.86</u>	.90	.87		.91	.88	<b>.90</b>	.92	.93	.92	.89	.91	.64	

Table 4.3.5: Geographically transferred crash prediction *auc* for NN and RF for the various approaches. Best results for each target area are highlighted in bold.

City	NN <i>auc</i>					RF <i>auc</i>				
	A0	A1	A2.1	A2.2	A3	A0	A1	A2.1	A2.2	A3
Arezzo	.546	.575	<b>.828</b>	.813	.813	.822	<b>.969</b>	.841	.841	.892
Florence	.501	.636	<b>.882</b>	.845	.849	.921	.864	<b>.928</b>	.915	.848
Grosseto	.645	.590	.849	<b>.931</b>	<b>.931</b>	.686	.908	.918	<b>.938</b>	.888
Livorno	.493	.803	.775	<b>.966</b>	.961	.885	.834	.885	<b>.896</b>	.863
Lucca	.451	.824	.842	<b>.847</b>	.808	.781	.861	.885	<b>.890</b>	.888
Massa	.602	.811	.852	<b>.887</b>	.886	.678	.836	<b>.890</b>	.885	.865
Pisa	.548	.818	.844	<b>.854</b>	<b>.854</b>	.877	.918	.898	<b>.920</b>	.868
Pistoia	.561	<b>.892</b>	.763	.847	.850	.728	<b>.872</b>	.864	.833	.811
Prato	.735	.823	.863	<b>.937</b>	<b>.937</b>	.843	.661	.905	<b>.906</b>	.860
Siena	.522	.799	.869	<b>.925</b>	.920	.783	.826	<b>.916</b>	.856	.686
Avg	.561	.756	.836	<b>.885</b>	.880	.800	.854	<b>.893</b>	.887	.847
Std	.08	.11	.03	<b>.05</b>	.05	.08	.08	<b>.02</b>	.03	.06

to what was done for setting  $(D1.2)$ . We adopt and compare the three predictive models described in Section 4.3.2: Random Forests (RF, the same used in  $(D1)$ ), Deep Neural Networks (NN) and LightGBM. The results are summarized in Table 4.3.3, reporting recall,  $f1_1$  and *auc* for each province and each algorithm. We can easily see that both RF and NN have high and stable performances, especially in terms of *auc*, which is the most informative measure. On the contrary, LGBM performs poorly in most provinces (7 out of 10), and is always worse than the other methods. This led us to focus the rest of the experiments only on RF and NN. The last line of Table 4.3.3 reports the performances obtained merging the data of all the provinces, thus building a unique global model and testing it on all provinces. This is equivalent to setting  $(D1.2)$  on a different data sample or, from a different perspective, to setting  $(D1.3)$  at a smaller, regional scale. The results show performances that are perfectly aligned with the single provinces, suggesting that the

larger training set of the global dataset is well balanced by the specificities of the local models of the provinces. In particular, this means that the local training data of provinces is sufficient to infer reasonable models.

**A0: Baseline approach** The straightforward approach to exploit the data available in the source domains is to directly build a model using all the data, and try to apply it as is to the target domain. We experimented this approach as a *solution zero*, and its results are shown in Table 4.3.5, which will be used in the rest of this section as a reference for evaluating our proposed approaches A1-A3. As expected, this baseline results to be competitive with (though generally worse than) the simpler approaches (A1), and in most cases, significantly worse than the more sophisticated ones (A2-A3).

**A1: Best City Transfer** Here we consider the first geographical transfer learning strategy we proposed, namely to make predictions on a target domain (i.e., the province under analysis) using a local model selected among the source domains (in our case, the 9 provinces left) by taking the province which is most similar to the target one. The results are summarized in Table 4.3.4, which reports the performances for all the pairs “source province vs. target province“, marking in bold the values suggested by our first strategy. The performances are reported in terms of *auc*, and are shown for both the NN and RF algorithms. The values obtained suggest that the strategy works slightly better with RF, yet in general, it does not achieve satisfactory results, in most cases performing worse than the average. Apparently, single models do not provide knowledge which is directly usable, *as is*, in other areas, and then something more refined is needed.

**A2: Weighted Ensemble Model** We test the second proposed approach, which consists of combining all the local (source) models into an ensemble, where their predictions over the target domain are aggregated. We compare our weighted combination, where each province votes with a weight proportional to its similarity w.r.t. the target, against a baseline where the weights are perfectly homogeneous. The baseline is named *A2.1*, while the weighted solution is named *A2.2*. Table 4.3.5 reports the results obtained for the two methods over each province, taken in turn as target domain, compared against the corresponding results of the best city transfer approach, named *A1*. Again, the results are shown bot for NN and RF, using *auc* as reference metrics, and highlighting in bold the best results. We can see that both *A2.1* and *A2.2* consistently improve over *A1*, thus confirming that combining the information of multiple sources is better than focusing only on one. At the same time, we can observe that *A2.2* performs overall much better than *A2.1*, especially with NN, proving that in this strategy, the similarity information becomes much more useful than what happened with the single-domain approach. Besides that, we can also notice that between NN and RF there is not a clear winner.



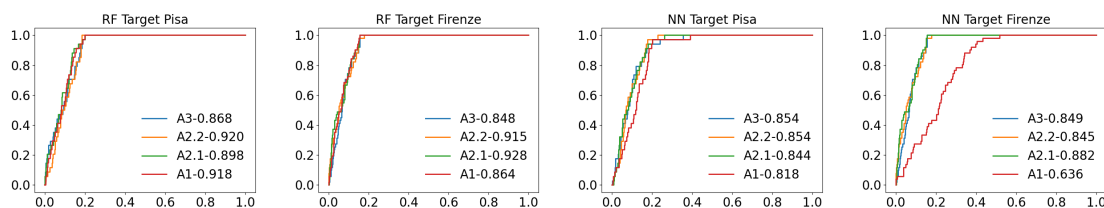


Figure 4.3.5: Receiver Operating Characteristic (ROC) curve for geographically transferred crash prediction with target areas Pisa and Florence for  $D2$ .

**A3: Weighted Sampling** With the third strategy, we combine the local information of all (source) provinces at a lower level, combining data rather than models. As before, each province is considered in turn as target domain, yet this time we build a predictive model from scratch, obtaining the training data by sampling the training set of each source domain, taking larger samples from more similar provinces. The results are shown again in Table 4.3.5, under the column  $A3$ . Since the method involves a random sampling, the values shown are obtained as average over 10 distinct runs. The values point out that the strategy works relatively well in combination with NN, reaching very often performances equal or close to the best ones, yet providing overall slightly less convincing results (on average, there is a drop of 0.5% of  $auc$  w.r.t.  $A2.2$ ). Also, the performances with RF are much worse since the average drop is 4%, and it never gets close to the best solutions.

An additional overall comparison of the results is provided by Figure 4.3.5, which shows the ROC curves of the models obtained with all four strategies discussed above, over two sample provinces: Pisa and Florence. In the case of  $A3$ , one of the 10 models generated was (randomly) selected. The plots show that in both cities, despite the differences in total  $auc$ , all strategies provide rather steep curves, and thus reasonable results, except for  $A1$ , which is less stable and, indeed, in the case of the NN predictor has significantly worse performances w.r.t. the others.

**Conclusions on selecting the best transfer learning method** Summarizing the results seen above, we can conclude that combining the local knowledge of multiple sources is the key to improve performances in this transfer learning setting. This means, in particular, that using the baseline method  $A0$  and the single-source method  $A1$  is not recommended. In addition, the best level to perform such combination appears to be the weighted ensembling of local models ( $A2.2$ ), rather than directly combining local datasets ( $A3$ ), suggesting that in our data, the more detailed information that resampling strategies could in principle provide is outweighed by the noise that they introduce – noise that the local models have lost, together with other bits of (potentially useful) information. However, the data size and variability in different applications might change this equilibrium. Thus we suggest considering both approaches as reasonable candidates to test.

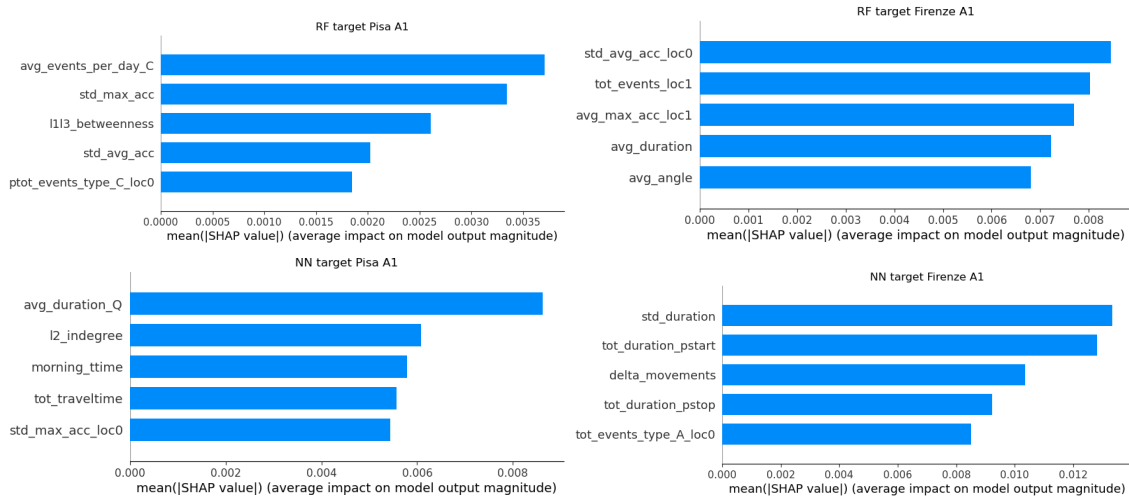


Figure 4.3.6: *Aggregated SHAP explanation of the five most important features for geographically transferred crash prediction with target areas Pisa and Florence for D2 using A1.*

**Geographically Transferred Crash Prediction Explanation** Like in [126], a parallel objective of this work is to understand which behaviors in a driver more likely could lead to future crashes. We realize it by adopting the SHapley Additive exPlanations (SHAP) method [194] to locally estimate for each prediction the expected contribution of each feature. SHAP returns the shapely values: the higher is a shapely value, the higher is the contribution of the feature; if the shapely value is positive, it contributes towards the positive class (crash); otherwise it contributes towards the negative class (no crash). From [126] emerges that IMN-based features and collective features are fundamental for detecting crashes: the average maximum acceleration of break events in areas visited occasionally performed by other users is crucial in pushing towards the crash. Another feature having this effect is the number of acceleration and break events between the second and third most visited locations.

In the following, we summarize SHAP explanations by reporting the mean values of the absolute SHAP values for the drivers having a car crash. We focus our study on A1 and A3 to observe the differences between an approach trained on a single geographical unit (A1), and an approach trained on multiple weighted areas (A3). The idea is to understand which features are the most important for recognizing crashes in geographical transfer learning. The results are reported in Figure 4.3.6 for A1 and in Figure 4.3.7 for A3. We report the explanations for the records for both NN and RF, using Pisa and Florence as target domains. The longer is the value bar, the higher is the contribution of the corresponding feature. We focus on the top five values.

In general, we observe that there is not a clear pattern among the different classifiers and geographical units. Similarly to the observation reported in [126], for

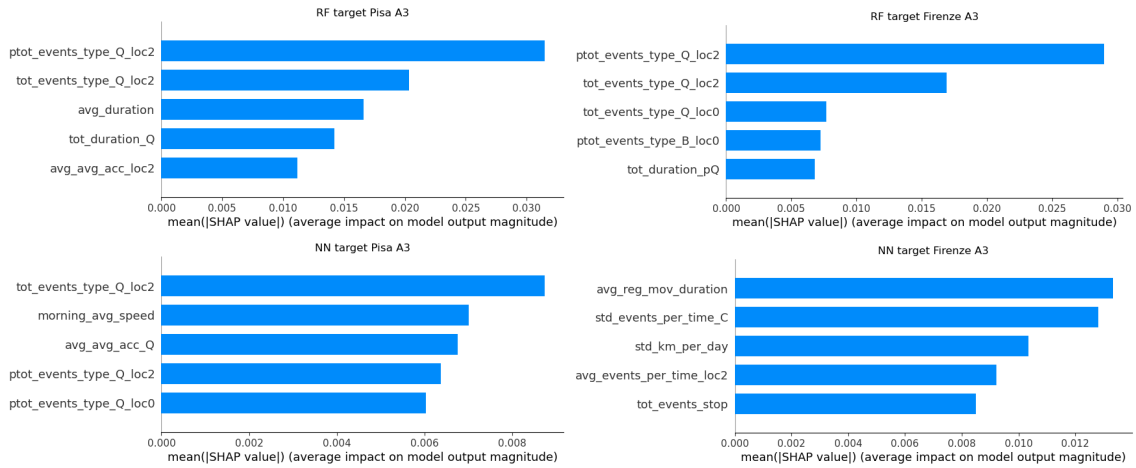


Figure 4.3.7: *Aggregated SHAP explanation of the five most important features for geographically transferred crash prediction with target areas Pisa and Florence for D2 using A3.*

A1 in Figure 4.3.6, we have the presence of several IMN-based features like the betweenness of the movement from the first and third most important locations (`l1l1_betweenness`), the number of incoming edges in the second most visited location (`l2_indegree`), the events at the most important locations (`tot_events_loc1`), and the acceleration for reaching them (`avg_max_acc_loc1`). Moving the observations to Figure 4.3.7, we notice how all the classifiers highly rely on features related to events. This means that, when aggregating data from different sources, it becomes fundamental to predict a crash to discriminate along dimensions involving harsh accelerations, harsh braking, and harsh cornering. In particular, besides the events happening in general (like `tot_duration_Q` that means the total duration of harsh cornering), we notice how the focus is on events happening when driving towards the second most visited location (like `tot_events_type_Q_loc2` that counts the number of harsh cornerings for going to `loc2`). Finally, we underline again how IMN-based features are important. For instance, with NN over Florence using A3 (bottom right of Figure 4.3.7) we have that the most important feature for deriving a car crash is `avg_reg_mov_duration`, i.e., the average duration of the movements performed regularly. This suggests that performing general actions to reduce the travel time for such a specific portion of the mobility can have a significant impact on the probability of a crash in the area, improving safety overall.

#### 4.3.4 Notes on privacy and ethical issues

Many large-scale urban services have explicitly or implicitly collected anonymized mobility data of residents to understand personal mobility patterns, e.g., cellphone data (CDR and Connection), vehicular GPS data [34], and electronic toll collection data (ETC) [36]. Unfortunately, as it is shown in Sec. 4.4, while providing the

improvement for urban services, the collection of human mobility data is sensitive and it is possible to reidentify users from the dataset. In the context of our work some key aspects have to be considered about the IMNs adoption and the geographical transfer learning task. An IMN describes the individual mobility of a user through a graph representation of her locations and movements, grasping the relevant properties and removing unnecessary details. Its nodes correspond to locations that represent a group of stop points identified through a spatial clustering-based aggregation; and its edges correspond to movements representing groups of similar trips between two locations. That means we are able to identify the major point of interest of every driver (home place, work place, ecc) and reconstruct their daily trajectories. As mentioned in the previous chapter this could involve different levels of privacy. From the point of view of the single IMNs the risk hidden behind the use of these models are related only to the location of the most frequented places. However, no track or GPS data used is actually linked to a user but each trajectory is associated with an ID. The GPS data of our work is made anonymous through several techniques during each preprocessing phase.

A different situation applies to the transfer learning context. As explained in details in Sec. 2.3 Transfer Learning has quickly become an established pillar of deep learning for good reason. Base models allow for a new wave of extraordinarily accessible and effective deep learning applications. Whilst being incredibly powerful, one should always remember the general modeling principle that has proven applicable in all sorts of regimes: shortcuts and advancements in performance are almost invariably associated with externalities. In the case of transfer learning, there is one externality: security. A malicious hacker, it has been demonstrated, can effectively corrupt any deep learning model that utilizes transfer learning even if they don't have any access to it by exploiting publicly available base models. As deep learning models are increasingly being deployed in important decision-making applications, they become subject, like any other global technology, to hijacking and corruption. Adversarial learning is the name given to the study of how the fabric of deep learning models' learned knowledge representations can be exploited to yield malicious outputs. On the other hand, no major risks have been proven on personal privacy with the use of transfer learning: there are quite various works in the literature on techniques for preserving privacy using transfer learning. For example, in [329] authors presented an efficient mobility privacy risk prediction model, TransRisk, to predict the privacy risk of users based on transfer learning. Compared to previous work, TransRisk unifies multiple mobility datasets and employs an additional input, spatial-temporal tensor, to represent the spatial-temporal information of users from mobility data. In [132] a novel method is proposed, which is based on HTL, to solve the knowledge sharing problem from the source to the target. This method draws a best balance between privacy-preserving concern, transfer learning performance and target-domain data sizes proving that the algorithm has an  $\epsilon$ -differential privacy guarantee for both the source and target. Therefore this work does not present any further risks than those detailed in the previous section.

### 4.3.5 Conclusion and Future Work

In this section, we have introduced the long-term car crash prediction problem, its associated task of risk assessment and the geographically transferred car crash prediction problem. For the first problem, we proposed a solution consisting in extracting sophisticated features of the user’s mobility, able to capture not only basic characteristics of her mobility, but also higher-level information derived from a network view of her mobility history as well as contextual knowledge directly inferred through analysis of the collective data of all users. On top of such features, machine learning models can be trained and successfully employed. Experiments on real data showed that our solution outperforms basic solutions based on state-of-art features, and a preliminary inspection of the prediction models through explainable AI methods allowed us to identify a few representative features associated with crash risk. For the second problem, the solution proposed consists in exploiting city indicators that can be derived from mobility data to design geographical transfer learning solutions based on the ensemble principle and weighted through city similarities. The experimentation on real data demonstrated that solutions employing city indicators for driving the transfer overcome standard baselines that do not use them. Explanation techniques also revealed some of the features that are most important for the success of the transfer learning methodology.

The results and insights obtained with this work opened several research and practical questions that we would like to address in the future, among which we mention the following. First, the IMN representation adopted in the driving modeling phase appears to be the right tool for enriching the data with higher-level semantics, such as the purpose of trips and stops, as done in [257], the driving *moods* (e.g., through unsupervised analysis of speeds and accelerations, or driving through dangerous intersections [269]), or by better describing the evolution of driving habits. Also, contextual data might be expanded, integrating several external, public data sources, such as the presence of Points of Interest, the road network structure, weather conditions, etc. While the model explanation tools were used in this work as a means for understanding the causes of crashes, their application can be further extended to improve the performance of the models by integrating feedback from domain experts – a *human in the loop* approach that can be made possible by model explanation itself. The city indicators we adopted, which are at the basis of our transfer learning proposals, are just a subset of a large spectrum of possible choices, our current purpose being to yield a general characterization of the urban areas involved. However, searching the optimal set of city indicators to reach the best model transferrability on the specific prediction problem would be indeed an interesting extension of the current work. Also, while the paper was focused on crash prediction, the transfer learning methods proposed are based on rather general principles (see Section 4.3.1) that can apply to a much broader set of problems. In particular, any learning problem related to mobility in the urban context might fit the framework, from the classification of points-of-interests to the estimate of a driver’s fuel consumption. We consider exploring some of these alternative application settings as an interesting line of future work.

## 4.4 Assessing privacy risks in human mobility

Nowadays, mobility data include a set of data types with different origins and sources but that alone, or combined, give information on how an individual moves, where she usually goes and what activities she carries out. From a legal point of view, mobility data are not considered as such per se sensitive data (as health or political opinions data are) because they do not reveal sensitive personal information of the individual on their own as described in article 9 of the GDPR (like ethnic origin, sexual or religious preferences, political opinions, etc.). However, what we highlight is how apparently unproblematic mobility data can become risky for privacy when they are combined or thoroughly analyzed with the relevant methods and/or external data. Indeed, even if they are not per se sensitive personal data they may easily reveal sensitive and confidential information which need to be shielded. If data are mined appropriately, from mobility data it is possible to find out or infer not only the user behaviours and the places she visits, but also who the user is (initially anonymized), where she lives and what her health status might be. From the places regularly visited, often sensitive data can be inferred with a high degree of reliability. For instance, Sunday visits in a church or Friday ones to a mosque easily reveal religious beliefs as it does the presence at a political event for political opinions. From harmless data it is possible to build an identikit of anyone, and a deeply disturbing one both for its content and for its possible use. Note in fact that, although the GDPR does not apply to anonymous data (art. 2, 4 (1)) it is also true that the borderlines between anonymity and re-identification are progressively thinner. Indeed recital 26 clarifies that to determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly'. This clarification is a key element of our journey because the principles of data protection should therefore not apply to anonymous information, namely information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable'. Such reasonableness needs to be ascertained considering all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments'.

### 4.4.1 Personal Data vs Sensitive Data

Personal data [313] protection legal rules represents the technical-juridical tool through which national and EU legislators protect all the rights connected to personal identity.

A data is considered personal if it allows the identification of the individual (natural person) or if it describes the individual in such a way as to allow identification by acquiring other data. Both types of data are protected in the same way.

With the term identification, therefore, we mean the possibility of distinguishing the person from any other subject (e.g. qualification as secretary of State) or within

a category. If identification requires the acquisition of additional data for which unreasonable time and costs are required, then the person cannot be considered identifiable. Thus data are not personal and the legal rules on personal data protection do not apply at all.

However, it is not necessary to reach a high level of identification (let us think of the names that correspond to more than one person) for the data to be subject to protection.

The European Union Court of Justice has developed a test for identifiability already under the EU Directive 95/46/EC, the so called Breyer test (Case C-582/14 ECLI:EU:C:2016:779) clarifying that (at 43) “*it is not required that all the information enabling the identification of the data subject must be in the hands of one person*”). Thus, personal data is a dynamic concept, which must always be referred to the context, in the sense that even if an isolated information is not able to lead to the identification of an individual, such an information could be used for identification through crossing with other data. This determines the nature of personal data.

Hence the nature of personal data is not an absolute one but it depends on “*all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly*” (Recital 26 [313]).

What makes a “means” reasonably expected to be used depends on many factors. As anticipated, recital 26 suggests that “*account should be taken of all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments*”.

Thus the notion of “personal data” depends on many variables. Here we argue a similar path for sensitive personal data (called “special category” by the GDPR at article 9) whose borders with non-sensitive data are fading away. Whether or not a personal data turns into a sensitive one (belonging to the special category listed by art 9) depends on several factors. We move in this analysis along the lines theoretically set already by [198] adopting mobility data as a use case. In [198] authors defined as “quasi health data” those data useful to predict or determine the health status but that are not directly related to it [71].

We are going to see if the case of mobility data falls in the category. Thus the notions we are elaborating upon are anonymous data, personal data, sensitive personal data and inferred data. Personal data that has been rendered anonymous in such a way that the individual is not or no longer identifiable is no longer considered personal data. For data to be truly anonymised, the anonymisation must be irreversible with the caveats illustrated by recital 26. The notion it results in is dependent on, among else, security measures, the chosen architecture for ingesting and processing data, accessibility of data (connected or not to internet). The GDPR protects personal data regardless of the technology used for processing

them.

Examples of personal data are:

- a name and surname;
- a home address;
- an email address;
- an identification card number;
- location data

A special attention is given to the so-called *sensitive data* (special categories of data subjected to more stringent rules). These are categories of data that historically lent themselves to larger abuses against fundamental rights and freedoms (e.g. via discrimination). Their heightened protection aims at protecting the core values of our societies, human dignity and prevent possible discrimination. That is why society perceives them as more delicate than simple personal data and their processing is as a default rule a prohibition (art. 9 [313]). The general principle is that their processing is prohibited unless one of the specific exceptional grounds apply. Moreover, access to sensitive data should be limited through sufficient data security and information security practices designed to prevent unauthorized disclosure and data breaches. Article 9 GDPR lists the special categories of data considered “sensitive”. They are “personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership, and the processing of genetic data, biometric data for the purpose of uniquely identifying a natural person, data concerning health or data concerning a natural person’s sex life or sexual orientation shall be prohibited.”

Therefore, sensitive personal data is a specific set of “special categories” inside the personal data context that must be processed with extra caution.

Our departing analytical point is that the borderlines between the two categories of personal data is more fluid than appears from the formal statutory definitions. There are data that might not be considered *prima facie* sensitive as such or even personal data, but that could produce sensitive information enabling either discovery or inference of personal and even sensitive data.

These data apparently seem harmless but potentially conceal confidential information. The protection standard for this type of data is much less clear-cut. Note also that applicable legal regimes range from non-application of the GDPR to the application of its most stringent rules. Note also that in redefining the boundaries between personal and non-personal data and between personal data and special categories of personal data an important role is played by both the notion of inferred data and the kind of attacks data can endure [206].

In this work we concentrate in understanding how mobility data are classified



from a legal point of view and how they should be classified according to how dangerous they can be for privacy attacks. More in details we will illustrate that, although at first glance not considered personal (or at least sensitive) data, some mobility data can generate sensitive data or lead to infer sensitive data (without certainty on their accuracy and correspondence to reality) that are used to make decisions upon individuals impacting their rights. With reference to this last element a key point is the purpose of inferred personal data processing and the relevance of their (un)accuracy that can lead to serious violations of fundamental rights and plain violations of the core principles set in the GDPR.

#### 4.4.2 What can I infer with mobility data?

The quick evolution and wide diffusion of technologies for the localization of devices (especially smartphones and vehicles' GPS) as well as location-based services, is leading to the production and collection of large and diversified traces of human mobility, every day more detailed and pervasive. These traces potentially contain a huge amount of information that might allow inferring models of human mobility spaces at unprecedented levels of precision and depth. They would be key enablers of many applications, ranging from monitoring urban traffic features to reconstruct inter-city mobility demands and region scale structures, which could help in making modern urban spaces more sustainable, efficient and comfortable for citizens. They can also enable the monitoring of epidemic like the COVID-19.

Starting from trajectory reconstruction (translating sequences of single location fixes into a complete movement trajectory, possibly including map-matching) is possible to develop several methods for processing and analyzing mobility data.

In this section we would like to give an overview about mobility data and their potential to "generate" sensitive personal data. Based on the level of data enrichment, it is possible to infer more and more information about individual users. Furthermore, the addition of semantics or external information (road conditions, weather conditions, etc.) makes it even easier to make predictions.

We will start considering the so called "Observed data" to see what happens enriching them with more information step by step, until arriving to have "Inferred" or "Predicted" data.

There is a long way to go from raw data to useful representations of mobility behaviors: we can call it a mobility knowledge discovery process [115].

##### Raw Data

Once you have an available mobility dataset what you really have is a sequence of points with a different sensitivity depending on the data type (if we have CDRs the spatial sensitivity is lower than the GPS one).

Let us consider a dataset compound of GPS points which are spatio temporal points (longitude, latitude and timestamp).

The first step is to analyze the data in order to recognize and build the trajectories

and paths taken by users. A strict definition of movements relates this notion to change in the physical position of an entity with respect to some reference system within which one can assess positions.

A trajectory is a path made by the moving entity through the space where it moves. In studying movements, an analyst attends to a number of characteristics, which can be grouped depending on whether they refer to states at individual moments or to movements over time intervals. Moment related features include position in a particular moment, position of the entity in space, direction of the entity's movement, change of direction, speed of the movement [112].

Several mobility data sources also provide information about events of various types, detected by the device. They are usually related to acceleration and direction, or to events happened within the device: harsh acceleration, harsh braking, harsh cornering, multiple cornering, vehicle switch-on(start) and switch-off(stop). In some cases the acceleration magnitude, the maximum acceleration, angle and duration are available too.

Now suppose we can add semantic data to our dataset. For instance if we consult a road map we could overlap the GPS path with the real streets in order to discover the geographic movements of the users. A road map is enough to start inferring knowledge: which are the most frequent routes, which are routine paths and which just occasional ones. Moreover, if we suppose to have also information about road conditions, the speed limits and the synchronization of the traffic lights we can define how a user drives or how a pedestrian moves in the city.

An analogous reasoning could apply to wearable devices. The terms 'wearable devices' and 'wearables' all refer to electronic technologies or computers that are incorporated into items of clothing and accessories which can comfortably be worn on the body [296]. A wearable should have sensors for the physical environment such as location (for example GPS), cameras, microphones, temperature, humidity, movement, etc. This list of sensors can be extended with biometric signal sensors, such as heart rate or Galvanic Skin Response sensor (GSR), etc. In any case, these definitions could be applied to many different kinds of devices.

Indeed, a plethora of devices can be found in the market fitting more or less in the previous definitions, from small appliances that are built into the shoes sole or the insole to small devices that can be incorporated into users' glasses. Despite all these options, the more adopted wearables today are wrist wearables namely smartbands and smartwatches [81]. During the past decade, rapid progress in wearable sensor technologies eased long-term physical activity behavior monitoring in real-life conditions. Among the existing sensors included in the wearable devices, three-dimensional (3D) accelerometers have gained the most attention. A 3D accelerometer measures acceleration forces in  $x$ ,  $y$  and  $z$  dimensions, and therefore can sense the status of a body's motion or postures [7].

Combining GPS and accelerometer sensors has been useful in improving movement monitoring of humans, particularly in daily life. In the transport mode detection domain, the combination of GPS and accelerometer sensors is more useful than

using each sensor individually, specifically in differentiating transport related activities such as walking, cycling and running.

We can categorize the use of GPS sensors into two broad applications. The first application mainly focuses on utilizing GPS spatial coordinates to link mobility behavior derived from accelerometer data to the location and relevant spatial data such as land use, walkability, green spaces, neighborhood and exposure in a geographic information systems environment [8]. These links enhance our contextual knowledge of the relationship between objectively measured physical activities and social environments [58]. The second application uses features such as time, distance, altitude and speed derived from GPS data to inform classifiers in mobility detection[9].

So, following the examples mentioned above, it is easy to recognize how with few accessible information it is possible to find out a lot about an individual, entering his/her privacy area.

### Trajectories and POIs

As mentioned in the previous chapters, people perform movements in specific areas and time instants. These people are called users and each movement is composed by a sequence of spatio-temporal points  $(x, y, t)$  where  $x$  and  $y$  are the coordinates, while  $t$  is the timestamp. We call trajectory the sequence of spatio-temporal points which describe a movement. The set of trajectories travelled by a user makes her individual history.

Thanks to these two elements it is possible to enrich mobility data with annotations about human activities. These approaches are focused either on places of general interest (like restaurants, shopping center) or on individual based destinations (like home or work) and yet they might lead to discover other individual destinations (e.g. clandestine meeting points for mistresses, political activities,...).

The mobility history of a driver may enable many services such as location recommendation or sales promotion.

Hence, analyzing the trajectories of individuals, it is possible to obtain a great deal of information. For every user a data scientist can create a mobility profile that describes an abstraction in space and time of her systematic movements, ignoring exceptional paths. Thus, the systematic behavior of every driver can be modeled with her mobility profile and the daily mobility of each user is characterized by her routines [123].

Instead, if we focus the attention not on the single user but we consider the collective aspect, having the trajectories it is possible to trace even the relationships between users. Indeed using raw trajectories, we can first compute flocks and encounters, then from these encounters find a method to infer relationships.

The basic idea is the following: if two users travel the same roads at the same time or attending the same places then they are likely to know each other.

Moreover, counting how many times two people stay together (according to how many time their trajectories coincide) it is possible to build a hierarchy of relationship in order to understand the degree of relationship between two users.

A similar work is found in [105]. Therefore we can see that already by considering the trajectories instead of the raw data allows us to reach a much higher level of inference and deduction turning mobility data into sensitive data. A conclusion already reached by the WP29 [88]. The same WP29 in an earlier opinion concluded that special categories cover “*not only data which by its nature contains sensitive information but also data from which sensitive information with regard to an individual can be concluded*” [73].

### Social Media Data

The introduction of location-based services in social media applications of smartphones has enabled people to share their activity related choices (check-in) in their virtual social networks (e.g. Facebook, Instagram, Twitter etc.) providing unprecedented amount of user-generated data on human movement and activity participation. This data contains detailed geo-location information, which reflects extensive knowledge about human movement behavior. Moreover, the venue category information for each check-in is recorded from which user activities can be inferred.

If analyzed properly, such data can help to better understand how citizens experience the cities they live in. Note that all these data are already from the outset personal data since they are linked to specific profiles. Also they can help identify mobility data which are not related to individuals by allowing the association of devices to individuals and to run cross-device associations[38].

Compared with other data sources, social media data has its unique characteristics such as more social information, which provides a multidimensional view of studying human mobility patterns. A direction to obtain accurate estimates of people’s activities is to combine data from different sources, for example combining GPS data with geo-tagged social network data could be very useful to improve the data mining process [121]. The former provide a sample of a user’s whereabouts but are noisy and lack semantics, the latter provide visits to venue of exact locations but they are not able to give information about the paths.

There have been extensive studies in mining geo-tagged social media data. For example in [137] the authors analyzed urban human mobility and activity patterns using location-based data collected from social media applications also exploring the frequency of visiting a place with respect to the rank of the place in individual’s visitation records. Therefore, if there is the possibility to collect different kind of mobility data, from raw data to social networks ones, we not only have the ability to predict future behavior but also to reconstruct their personal information and relationships. Attackers may combine the data to identify the anonymized users invading the privacy area of everyone [71].

By following all these crumbs and connecting the dots any attacker could reconstruct the personal’s file of everyone harnessing inferential analytics[104].

With wearable sensors data it is possible to make a similar reasoning since they are another type of data that is interesting to recall. As mentioned in the previous

paragraphs wearable devices offer new opportunities to monitor human mobility activity continuously with the miniature wearable sensors embedded. However, there are few challenges faced on smartwatches about security issue which put users' safety and privacy appear at risk [66].

For instances sensors as accelerometer, which is used to measure linear acceleration and it can determine whether the device is horizontal or vertical, and whether it is moving or not counting the steps a user takes, may hide several others functionalities. GPS sensors are integrated in wearable devices too, in order to locate a person's location and create a whole picture of her own mobility history.

But these kind of sensor allows also to go beyond their primary purpose: for example using accelerometers is possible to detect a range of activities including step counts, worn/not worn state, overall physical activity levels, eating behavior, pill bottle opening movements, scratching, cardiopulmonary resuscitation (CPR) compression depth and frequency.

In [299] a study used smartwatch accelerometer as compared to ground truth video to identify eating moments in 7 participants for a single day with 66.7% precision and 1 participant for 31 days with 65.2% precision. Besides that in [190][79] authors used accelerometry to detect seizures in epilepsy patients and tremors. That is not all, since wearable devices also include sophisticated sensors specially designed to monitor health parameters which provide human activity measurement such as sleep quality, burned calories and other personal health metrics like heart-rate, body temperature, stress and hydration levels.

Some studies used gyroscope data in conjunction with accelerometer data for posture diagnosis and to detect palm- upward wrist rotation with 100% recall and 82.5% precision [164].

Wearables are collectors of a large set of confidential information in a way that allows to infer a lot about people lifestyle and their own health status. Just to give a current example: during 2021 Covid pandemic researchers and experts fielded a new app that aimed to exploit data extracted from smartwatches to prevent Covid-19 cases [133]. In short, researchers wanted to develop a new remote computer model capable of carrying out a first screening in the monitoring activity of people positive to the Covid-19 virus of a large portion of the population.

The most sophisticated smartwatches are able to measure oxygen saturation, heart rate and blood pressure, all important parameters to be included in an Artificial Intelligence engine to build the risk profiles of the individual citizens.

However, even in these moments, when the end seems to be able to justify any means, it is essential that the privacy rights of each person are preserved [87].

### 4.4.3 Hidden Risk behind Mobility Data

After considering all these data types, it is evident how the manipulation and the combination of this information can lead to obtain a whole picture of an individual's mobility. Starting from the raw data, which only supply the position

of the subject, some singularities of the individual user can already be identified. With the adequate computational capacity it is possible to analyze the data and recognize significant stops within the same trip for instance. After that, by adding semantics and recognizing the geographical areas, it is possible to understand the reason for the stops (a supermarket, take the children to school, go to the swimming pool, visiting an healthcare facility, etc.)[71].

We could offer many other examples to show the ease with which everyone could deal with this kind of data inferring sensitive personal data.

Focusing the attention on mobility paths, trajectories and semantics of the territory enable to identify daily travel routes, attackers may use trajectory data to deduce individual's mobility patterns and identify their home and workplaces or other "special" ones.

But that is not all: it is possible to reconstruct the individual behavior and understand the relationships between users who travel the same roads or frequent the same points of interest. Even if the data are originally anonymized, it is clear that if we know how a user moves, what places she attends, where she lives and works, it becomes immediate to go back to her identity.

Anonymizing user identities is not enough to protect people privacy. Then using social networks data any attacker could use the location tags (or hashtags) to verify the visit frequency of a given point of interest in order to correlate by matching people profiles and trajectory data to identify the users. They could also infer users' preferences, relationship and personal habits.

In conclusion, adding knowledge from wearable devices one can map the user, recognize the locations where she goes, the speed of her movements (how many steps, how many calories burned) and at the same time the heart rate, the percentage of oxygen usage and the hydration level. Leaving aside for a moment the specific sensors that collect health data within smartwatches it is important to underline again that is possible to infer health information, or possible risks related to that, only from tracking and mobility sensors. Only by using mobility data we get to define the health status of any person.

#### **4.4.4 Conclusions**

In summary, after the presented analysis, we could say that mobility data become quasi-health data [198] since we are able to infer users health conditions from studying their movements. Even if mobility data are not inherently medical data, without the right protection level any attackers could easily lead to conclusions about individual's health conditions. Note also that the "attacker" could be the legitimate data processor gathering the mobility data if she has a legal basis for such a further processing. We showed how the label "sensitive data" does not guarantee that there can be no privacy attacks through the use of other data not tagged as such. It is necessary to identify the non-sensitive data that provide information with a high degree of confidentiality and which are equally risky for

privacy protection.

All this, if we consider a third party attack that might lead to reidentification. The situation gets more problematic once we consider the actual/potential uses of these inferred (sensitive) data by data controllers themselves.

As we noted, location and mobility data are collected by different sources and often transferred to third parties who have other datasets of information enabling both reidentification and further data inferences by applying models and correlation patterns to the enriched mobility data. Again, regular presence on Fridays at a location corresponding to the address of a mosque easily lead to infer religious belief without the need to apply a complex model.

Mobility data described above can trigger the application of models “qualifying” the individual for specific features, health risks, for instance or sexual habits (e.g. recurrent passage and stops in an area of prostitution), religious habits. Once this qualification is obtained, what is relevant is its use, that is the actual application of the model with all the obvious implications in term of decision-making.

If a decision maker has to act upon a large number of individuals it might be satisfied with a certain degree of accuracy in the application of the model to the dataset triggering it.

Recalling the previous example of the risk prone behavior, a zip code might become the data triggering the application of the model. In other words, *“users of data mining outputs could be willing to use these results although aware that the output might not be correct”* [71]. Note that the WP29 has clearly identified as personal data those “likely to have an impact on a certain person’s rights and interests” [72].

Once the model applied to mobility data suggests a certain degree of health risk (e.g. developing diabetes, a risk prone driving attitude), what matters is not the fact that the suggestion can be considered “data concerning health” but the actual use of this inferred information as such. The emerging issues here can be characterized both in terms of ownership of the inferred information (to the data controller or the data subject) and in terms of accuracy of the information itself. On the latter, personal data need to be accurate. The accuracy principle provided for by art. 5.1.d requires that personal data are *“accurate and, where necessary, kept up to date; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay”* [244]. On the former, it has been questioned that inferred data are personal data at all [314].

However, both issues rely on the fact that such information is considered as personal data and have to cope also with the impact on groups, not only individuals. Indeed, the ability to challenge conclusions deriving from inferred data is problematic for individuals, it is even more so when the application of inferences does not reach directly the individual level [327]. Noteworthy is the fact that individuals have little or no power on data made anonymous before creating the models applied to them unless specific legislation is triggered (e.g. antidiscrimination rules). For this reason the call to establish a “right to reasonable inferences” as a normative goal

*de iure condendo*, although acceptable, lacks of bite.

Once it is accepted that mobility data, although originally anonymous, can lead to identification and to reveal special categories of personal data pursuant to art. 9 GDPR (“data concerning health” for instance) an higher level of protection can be channeled by art. 35 GDPR. It imposes a Data Protection Assessment (“DPIA”), with consequent actions, every time “*a type of processing in particular using new technologies, and taking into account the nature, scope, context and purposes of the processing, is likely to result in a high risk to the rights and freedoms of natural persons*”.

A DPIA is especially required in case of “a systematic and extensive evaluation of personal aspects relating to natural persons which is based on automated processing, including profiling, and on which decisions are based that produce legal effects concerning the natural person or similarly significantly affect the natural person”. Pursuant to the previous analysis, every time inferred data trigger the application of a model producing “*legal effects concerning the natural person or similarly significantly affect the natural person*”, especially when based on automated processing would impose a DPIA with its characteristics.

As described by art 35.7 [313] this must include “(a) *a systematic description of the envisaged processing operations and the purposes of the processing, including, where applicable, the legitimate interest pursued by the controller; (b) an assessment of the necessity and proportionality of the processing operations in relation to the purposes; (c) an assessment of the risks to the rights and freedoms of data subjects referred to in paragraph 1; and (d) the measures envisaged to address the risks, including safeguards, security measures and mechanisms to ensure the protection of personal data and to demonstrate compliance with this Regulation taking into account the rights and legitimate interests of data subjects and other persons concerned.*”

Once the implication of inferred data leads to generalize a duty to perform a DPIA, the recommended publicity of the DPIA forces upon data controllers the adoption of appropriate safeguards and information duties, expanding the protection potentials of the GDPR even before a formal recognition of a right to reasonable inferences.

In conclusion, it is clear that human mobility data since they are so an important proxy to understand human mobility dynamics and develop analytical service, need to be protected with the right level of privacy. Unfortunately this kind of data are very sensitive since they may enable the re-identification of individuals in a database in several different ways. It is essential to consider them as sensitive data and apply to them the best privacy protection framework.



# Chapter 5

## Epilogue

To keep the jargon of the play I called this final chapter "Credits". In it we summarise the results, discuss the limitations of our work, and sketch an outlook on future work.

### 5.1 Conclusions

We began this thesis by introducing in Chapters 2 and 1 the overarching theme and motivation for our research: that of finding valuable knowledge about mobility in terms of individual behaviors through a data-driven approach which combines techniques from network science and data mining.

The common thread of the work of my PhD program is the study of mobility at an individual level. All the works presented and published start from the study of the behavior of the individual user. In this thesis we investigated the potential of Mobility Data and their ability to reveal patterns useful both for single users and for mobility managers, owing to the availability of GPS data sources. We faced the challenge firstly focusing on GPS data and traffic estimation, providing efficient methods to extract traffic knowledge from low sampled GPS trajectories, discovering where drivers move.

In Chapter 3.1 we saw in details the importance of studying human mobility in terms of complex networks. The goal was to show the potential behind this kind of abstraction that allows many level of applications. We focused our studies on graph theory and on how to exploit it in the urban context by creating the Individual Mobility Networks (IMNs). Moreover, in the same chapter 3.3 we presented the work about user adaptive methods for solving the trajectory segmentation problem, a very common and useful task in mobility data mining, especially in preprocessing phases. The solutions proposed take into consideration the overall trajectory of the user, identifying an individual cut time threshold and also combining the information coming from the different users through the spatial regions they share. This process yields thresholds for trajectory segmentation which are not only user-adaptive, but also location-adaptive, thus taking into account that a stop at different places might require time intervals of different duration to be considered a

significant stay – and thus a trajectory cut point.

In the same chapter we also explored some Graph Embedding methods (Sec. 3.4) in order to understand how much our IMNs are different from typical benchmarks used in the graph embedding literature. We found that there are both semantic and statistical difference. On the one hand, the majority of benchmarks deal with human interactions data or physically connected elements of molecules, whereas IMNs are about movements, thus making recurring concepts in embedding literature like information propagation and bindings not perfectly fit; from the other, the empirical exploration of IMNs' properties resulted to be different from many of the others. The review of existing graph embedding methods highlighted the existence of a limited set of fundamental approaches, plus several variants and improvements. That makes it difficult to identify the promising methods to choose, for which reason we selected approaches from the most representative algorithms in the literature that could be applied to graph level embedding. Empirical evaluations highlighted how the (rich) node features of IMNs are fundamental to achieving acceptable results, yet also suggesting that most methods are not able to handle them properly.

In the second part of the thesis (also called *Act II*) we focused on applications enabled by IMNs. Besides that we also approached the big open challenges we mentioned in the first part of the thesis: explainability in mobility field and transfer learning. We leveraged some specific works to explore them in depth.

In particular three contributions has been presented in detailed. In Section 4.1 we proposed a methodology based on mobility data analytics, ad-hoc trip planning and simulation, that provides detailed quantitative information about what the switch from a petrol vehicle to an electric one can mean for the single users and for the collectivity. The proposed approach is efficient – thus suitable for large-scale studies – and takes into consideration the main aspects involved in EV-based mobility: limited driving range, sparse recharge infrastructures, potentially long recharge times, the possibility of recharging at home/work, and so on. The experimentation performed over an Italian region shows how the electrification process is expected to generate only minor issues at the collective level and yet individual users can expect slightly different impacts in they travel & refuel habits.

In Section 4.2 we have defined a large array of local and global city indicators providing that they can be successfully exploited in a task of mobility transfer learning. In particular, we have clustered municipalities based on the mobility behavior described by the city indicators and we have assessed the transferability of a machine learning model for traffic forecasting.

In Section 4.3 we have introduced the long-term car crash prediction problem, its associated task of risk assessment and the geographically transferred car crash prediction problem. For the first problem, we proposed a solution consisting in extracting sophisticated features of the user's mobility, able to capture not only basic characteristics of her mobility, but also higher-level information derived from a network view of her mobility history. On top of such features, machine learning models can be trained and successfully employed. Experiments on real data showed

that our solution outperforms basic solutions based on state-of-art features, and a preliminary inspection of the prediction models through explainable AI methods allowed us to identify a few representative features associated with crash risk. For the second problem, the solution proposed consists in exploiting the city indicators mentioned before to design geographical transfer learning solutions based on the ensemble principle and weighted through city similarities. Explanation techniques also revealed some of the features that are most important for the success of the transfer learning methodology.

In conclusion, it is essential to outline another significant aspect of the thesis: the analysis about the privacy aspects and the risks behind mobility data usage. As a larger part of modern life is digitized, individuals generate an increasing volume and variety of digital traces, which reveal information about their everyday activity and movements. In this context, we analyzed from a technical and legal point of view the potentials to infer personal sensitive data from mobility information and underline the importance of GDPR rules in terms of citizen safe.

## 5.2 Future Works

The natural evolution of this thesis can go in many directions. Each of the topics described involves future ways or research.

About the *Segmentation* task, the results obtained so far suggested us to explore the feasibility of some more flexible individual mobility models. In particular, the idea is to depart from the notion of single trips, and instead allow a multiresolution, hierarchical view where the same movement is interpreted both as one trip and, possibly, as a sequence of several small ones. The different levels of the hierarchy might be obtained by moving the time threshold up and down, linking the segments that originate from a split of an existing one. The resulting model would clearly be complex and its computation and management challenging. Moreover it will be interesting the integration of our methods into existing applications in the domain of crash prediction and simulations of Electric Vehicles mobility which are based on a detailed modeling of users' mobility history.

Also the *Graph Embedding* task provides the opportunity for new researches and improvements. In particular we think there are two main directions: the first is specific for IMNs and includes the development of embedding methods ad hoc for them, aimed to better exploit the abundant features at node and edge level and to cope with their particular graph structure. The second one is more theoretical and aims to classify existing methods based on what kind of information is actually built from the input graph structure and features, abstracting away from formalization and computational aspects, which often cover very similar concepts behind different covers.

In the framework of *Electric Vehicle Simulations*, the proposed approach is still amenable to improvements in several directions that we aim to explore, such

as integrating a longer look-forward planning, prioritizing recharges on trips that appear to have a more flexible timing and considering waiting times due to queue length at stations.

While, about the work on the *Car crash prediction*, the transfer learning methods show a plenty of possibilities to improve and develop new solutions. Indeed the methods are based on rather general principles (see Section 4.3.1) that can apply to a much broader set of problems. In particular, any learning problem related to mobility in the urban context might fit the framework, from the classification of points-of-interests to the estimate of a driver's fuel consumption. We consider exploring some of these alternative application settings as an interesting line of future work.

# Bibliography

- [1] *LightGBM's documentation*. <https://lightgbm.readthedocs.io/en/latest/index.html>.
- [2] *Octo telematics s.r.l.* <http://www.octotelematics.itg>.
- [3] Mohamed A Abdel-Aty and Rajashekar Pemmanaboina. Calibrating a real-time traffic crash-prediction model using archived weather and its traffic data. *IEEE Transactions on Intelligent Transportation Systems*, 7(2):167–174, 2006.
- [4] Osman Abul, Francesco Bonchi, and Mirco Nanni. Never walk alone: Uncertainty for anonymity in moving objects databases. *2008 IEEE 24th International Conference on Data Engineering*, pages 376–385, 2008.
- [5] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97, jan 2002.
- [6] Sander PA Alewijnse, Kevin Buchin, Maike Buchin, Andrea Kölzsch, Helmut Kruckenberg, and Michel A Westenberg. A framework for trajectory segmentation by stable criteria. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 351–360, 2014.
- [7] Hoda Allahbakhshi, Lindsey Conrow, Babak Naimi, and Robert Weibel. Using accelerometer and gps data for real-life physical activity type detection. *Sensors*, 20(3), February 2020.
- [8] Hoda Allahbakhshi, Lindsey Conrow, Babak Naimi, and Robert Weibel. Using accelerometer and gps data for real-life physical activity type detection. *Sensors*, 20(3), 2020.
- [9] Estela Almanza, Michael Jerrett, Genevieve Dunton, Edmund Seto, and Mary Ann Pentz. A study of community design, greenness, and physical activity in children using satellite, gps and accelerometer data. *Health Place*, 18(1):46 – 54, 2012. Active Living Research.
- [10] W Alonso. A theory of movements: Introduction. *Working Paper 266*, 1976.

- [11] Gamallo Alvaro, Gonzalez and Fraile-Ardanuy. Estimation of electric vehicles' consumption based on their mobility. Section 3.3.3. of Deliverable D6.1 of the DataSIM Project, 2013.
- [12] Theodoros Anagnostopoulos et al. Mobility prediction based on machine learning. In *MDM*, volume 2, pages 27–30. IEEE, 2011.
- [13] Alex Anas, Richard Arnott, and Kenneth Small. Urban spatial structure. *Journal of Economic Literature*, 36:1426–1464, 02 1998.
- [14] James Anderson. The gravity model. NBER Working Papers 16576, National Bureau of Economic Research, Inc, 2010.
- [15] Gennady Andrienko et al. (so) big data and the transformation of the city. *International Journal of Data Science and Analytics*, 2020.
- [16] Gennady L. Andrienko, Natalia V. Andrienko, Chiara Liliana Boldrini, Guido Caldarelli, Paolo Cintia, Stefano Cresci, Angelo Facchini, Fosca Giannotti, Aristides Gionis, Riccardo Guidotti, Michael Mathioudakis, Cristina Ioana Muntean, Luca Pappalardo, Dino Pedreschi, Evangelos Pournaras, Francesca Pratesi, Maurizio Tesconi, and Roberto Trasarti. (so) big data and the transformation of the city. *International Journal of Data Science and Analytics*, pages 1–30, 2020.
- [17] Mihael Ankerst, Markus Breunig, Hans-Peter Kriegel, and Joerg Sander. Optics: Ordering points to identify the clustering structure. volume 28, pages 49–60, 06 1999.
- [18] Valerio Arnaboldi, Marco Conti, Andrea Passarella, and Robin IM Dunbar. Online social networks and information diffusion: The role of ego networks. *Online Social Networks and Media*, 1:44–55, 2017.
- [19] Andreas Artmeier, Julian Haselmayr, Martin Leucker, and Martin Sachenbacher. The shortest path problem revisited: Optimal routing for electric vehicles. pages 309–316, 09 2010.
- [20] Yutao Ba et al. Crash prediction with behavioral and physiological features for advanced vehicle collision avoidance system. *Transportation Research Part C: Emerging Technologies*, 74:22–33, 2017.
- [21] Sungwoo Bae and Alexis Kwasinski. Spatial and temporal model of electric vehicle charging demand. *IEEE Transactions on Smart Grid*, 3(1):394–403, 2012.
- [22] Sambaran Bandyopadhyay, Harsh Kara, Aswin Kannan, and M. Narasimha Murty. FSCNMF: fusing structure and content via non-negative matrix factorization for embedding information networks. *CoRR*, abs/1804.05313, 2018.

- [23] Jie Bao, Yu Zheng, and Mohamed F Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *SIGSPATIAL*, pages 199–208. ACM, 2012.
- [24] Fateha Khanam Bappee, Amílcar Soares, Lucas May Petry, and Stan Matwin. Examining the impact of cross-domain learning on crime prediction. *J. Big Data*, 8(1):96, 2021.
- [25] Albert-László Barabási and Márton Pósfai. *Network science*. Cambridge University Press, Cambridge, 2016.
- [26] Hannah Bast, Daniel Delling, Andrew Goldberg, Matthias Müller-Hannemann, Thomas Pajor, Peter Sanders, Dorothea Wagner, and Renato Werneck. *Route Planning in Transportation Networks*, volume 9220, pages 19–80. 11 2016.
- [27] Anirban Basu, Anna Monreale, Juan Camilo Corena, Fosca Giannotti, Dino Pedreschi, Shinsaku Kiyomoto, Yutaka Miyake, Tadashi Yanagihara, and Roberto Trasarti. A privacy risk model for trajectory data. In Jianying Zhou, Nurit Gal-Oz, Jie Zhang, and Ehud Gudes, editors, *Trust Management VIII*, pages 125–140, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.
- [28] Michael Batty. Spatial entropy. *Geographical Analysis*, 6(1):1–31.
- [29] Moritz Baum, Julian Dibbelt, Andreas Gemsa, Dorothea Wagner, and Tobias Zündorf. Shortest feasible paths with charging stops for battery electric vehicles. *Transportation Science*, 53, 07 2019.
- [30] Moritz Baum, Julian Dibbelt, Lorenz Hübschle-Schneider, Thomas Pajor, and Dorothea Wagner. Speed-consumption tradeoff for electric vehicle route planning. In *ATMOS*, 2014.
- [31] Moritz Baum, Julian Dibbelt, Thomas Pajor, and Dorothea Wagner. Energy-optimal routes for electric vehicles. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL’13, page 54–63, New York, NY, USA, 2013. Association for Computing Machinery.
- [32] Hassan Bazzi, Dino Ienco, Nicolas Baghdadi, Mehrez Zribi, and Valérie Demarez. Distilling before refine: Spatio-temporal transfer learning for mapping irrigated areas using Sentinel-1 time series. *IEEE Geoscience and Remote Sensing Letters*, 17(11):1909–1913, November 2020.
- [33] Vahid Behzadan and Arslan Munir. Vulnerability of deep reinforcement learning to policy induction attacks. In *IAPR International Conference on Machine Learning and Data Mining in Pattern Recognition*, 2017.

- [34] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001.
- [35] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(85):2399–2434, 2006.
- [36] Michele Berlingerio, Bissan Ghaddar, Riccardo Guidotti, Alessandra Pascale, and Andrea Sassi. The graal of carpooling: Green and social optimization from crowd-sourced data. *Transportation Research Part C: Emerging Technologies*, 80:20–36, 2017.
- [37] Aude Bernard, Martin Bell, and Elin Charles-Edwards. Life-course transitions and the age profile of internal migration. *Population and Development Review*, 40(2):213–239, 2014.
- [38] F. Bertini, S. G. Rizzo, and D. Montesi. Can information hiding in social media posts represent a threat? *Computer*, 52(10):52–60, 2019.
- [39] C. Bharatiraja, P. Sanjeevikumar, Pierluigi Siano, K. Ramesh, and Raghu Selvaraj. Real time foresting of ev charging station scheduling for smart energy system. *Energies*, 10, 03 2017.
- [40] Reinaldo A.C. Bianchi, Luiz A. Celiberto, Paulo E. Santos, Jackson P. Matsuura, and Ramon Lopez de Mantaras. Transferring knowledge as heuristics in reinforcement learning: A case-based approach. *Artificial Intelligence*, 226:102–121, 2015.
- [41] Simona Bigerna and Silvia Micheli. Attitudes toward electric vehicles: The case of perugia using a fuzzy set analysis. *Sustainability*, 10:3999, 11 2018.
- [42] Vincent Bindschaedler and R. Shokri. Synthesizing plausible privacy-preserving location traces. *2016 IEEE Symposium on Security and Privacy (SP)*, pages 546–563, 2016.
- [43] Ella Bingham. Finding segmentations of sequences. In *Inductive Databases and Constraint-Based Data Mining*, pages 177–197. Springer, 2010.
- [44] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.
- [45] Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017.



- [46] Frederik Zuiderveen Borgesius. *European Data Protection Law Review 2017*, 3, 06 2017.
- [47] Ronald Bremer. *Outliers in statistical data*. Taylor & Francis, 1995. <http://www.deeplearningbook.org>.
- [48] Maike Buchin, Anne Driemel, Marc Van Kreveld, and Vera Sacristán. An algorithmic framework for segmenting trajectories based on spatio-temporal criteria. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 202–211. ACM, 2010.
- [49] Jacques Bughin, Eric Hazan, Sreenivas Ramaswamy, Michael Chui, Tera Allas, Peter Dahlstrom, Nicolaus Henke, and Monica Trench. *Artificial intelligence: the next digital frontier?* 2017.
- [50] H Bussemaker, H Li, and E Siggia. Regulatory element detection using a probabilistic segmentation model. *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology*, 8:67–74, 02 2000.
- [51] Chen Cai and Yusu Wang. A simple yet effective baseline for non-attributed graph classification. *arXiv preprint arXiv:1811.03508*, 2018.
- [52] Hongyun Cai, Vincent Zheng, and Kevin Chang. A comprehensive survey of graph embedding: Problems, techniques and applications. *IEEE Transactions on Knowledge and Data Engineering*, 30, 09 2017.
- [53] Corrado Canali. *Auto elettrica, tutto quello che c'è da sapere sulla ricarica a casa*, 2020.
- [54] Shaosheng Cao, Wei Lu, and Qiongkai Xu. Grarep: Learning graph representations with global structural information. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, CIKM '15*, page 891–900, New York, NY, USA, 2015. Association for Computing Machinery.
- [55] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, M. Sturm, and Noémie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015.
- [56] Oded Cats and Francesco Ferranti. Unravelling individual mobility temporal patterns using longitudinal smart card data. *Research in Transportation Business & Management*, 43:100816, 2022.
- [57] Joana Cavadas, Gonçalo Homem de Almeida Correia, and João Gouveia. A mip model for locating slow-charging stations for electric vehicles in urban areas accounting for driver tours. *Transportation Research Part E: Logistics and Transportation Review*, 75:188–201, 2015.

- [58] J.; Gullón P.; Bilal U.; Franco M.; Escobar F. Cebrecos, A.; Díez. Characterizing physical activity and food urban environments: A gis-based multicomponent proposal. pages 15–35, 2016.
- [59] Chakravarti, Laha, and Roy (1967). *Handbook of Methods of Applied Statistics, Volume I*. John Wiley and Sons, Hoboken, 1967.
- [60] JUDGMENT OF THE COURT (Second Chamber). "request for a preliminary ruling under article 267 tfeu from the bundesgerichtshof (federal court of justice, germany)", "2016".
- [61] JUDGMENT OF THE COURT (Second Chamber). "case t-557/20 : Single resolution board (srb) vs european data protection supervisor (edps)", 04 "2023".
- [62] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.
- [63] Cynthia Chen, Jingtao Ma, Yusak Susilo, Yu Liu, and Menglin Wang. The promises of big data and small data for travel behavior (aka human mobility) analysis. *Transportation Research Part C: Emerging Technologies*, 68:285–299, 2016.
- [64] Hong Chen and Hisashi Koga. Gl2vec: Graph embedding enriched by line graphs with edge features. In *International Conference on Neural Information Processing*, pages 3–14. Springer, 2019.
- [65] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. pages 785–794, 08 2016.
- [66] Ke Ching and Manmeet (Mandy) Mahinderjit Singh. Wearable technology devices security and privacy vulnerability analysis. *International Journal of Network Security Its Applications*, 8:19–30, 05 2016.
- [67] Eunjoon Cho et al. Friendship and mobility: user movement in location-based social networks. In *SIGKDD*, pages 1082–1090. ACM, 2011.
- [68] Glenn Cich, Luk Knapen, Tom Bellemans, Davy Janssens, and Geert Wets. Threshold settings for trip/stop detection in gps traces. *Journal of Ambient Intelligence and Humanized Computing*, 7(3):395–413, 2016.
- [69] Kristien Clement-Nyns, Edwin Haesen, and Johan Driesen. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. *IEEE Transactions on Power Systems*, 25(1):371–380, 2010.
- [70] P. Cocron et al. Methods of evaluating electric vehicles from a user’s perspective – the mini e field trial in berlin. *IET Intelligent Transport Systems*, 5:127–133(6), June 2011.

- [71] Giovanni Comandé. Opinions the rotting meat error: From galileo to aristotle in data mining? *European Data Protection Law Review*, 4:270–277, 01 2018.
- [72] European Commission. *The working party on the protection of individuals with regard to the processing of personal data*. 2007. [https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2007/wp136\\_en.pdf](https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2007/wp136_en.pdf).
- [73] European Commission. *Guidelines on Automated individual decision-making and Profiling for the purposes of Regulation*. 2016. [https://ec.europa.eu/newsroom/article29/item-detail.cfm?item\\_id=612053](https://ec.europa.eu/newsroom/article29/item-detail.cfm?item_id=612053).
- [74] Kristof Coninx and Tom Holvoet. A microscopic traffic simulation platform for coordinated charging of electric vehicles. In Yves Demazeau, Franco Zambonelli, Juan M. Corchado, and Javier Bajo, editors, *Advances in Practical Applications of Heterogeneous Multi-Agent Systems. The PAAMS Collection*, pages 323–326, Cham, 2014. Springer International Publishing.
- [75] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi. Demon: a local-first discovery method for overlapping communities. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 615–623, 2012.
- [76] Tom Croonenborghs, Kurt Driessens, and Maurice Bruynooghe. Learning relational options for inductive transfer in relational reinforcement learning. pages 88–97. International Conference on Inductive Logic Programming, 02 2008.
- [77] Livia Almada Cruz et al. Trajectory prediction from a mass of sparse and missing external sensor data. In *2019 20th IEEE International Conference on Mobile Data Management (MDM)*, pages 310–319. IEEE, 2019.
- [78] Elizabeth Daly and Mads Haahr. Social network analysis for information flow in disconnected delay-tolerant manets. *Mobile Computing, IEEE Transactions on*, 8:606–621, 05 2009.
- [79] Wile Daryl, Ranjit Ranawaya, and Zelma Kiss. Smart watch accelerometry for analysis and diagnosis of tremor. *Journal of neuroscience methods*, 230, 04 2014.
- [80] Rahul Deb Das and Stephan Winter. Automated urban travel interpretation: A bottom-up approach for trajectory segmentation. *Sensors*, 16(11):1962, 2016.
- [81] Francisco De Arriba Pérez, Juan Santos, and Manuel Caeiro Rodriguez. Analytics of biometric data from wearable devices to support teaching and learning activities. *Journal of Information Systems Engineering Management*, Vol. 1, 03 2016.

- [82] Michele De Gennaro, Elena Paffumi, Giorgio Martini, and Harald Scholz. A pilot study to address the travel behaviour and the usability of electric vehicles in two italian provinces. *Case Studies on Transport Policy*, 2(3):116–141, 2014.
- [83] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [84] Joseph L. Doob. The brownian movement and stochastic equations. *Annals of Mathematics*, 43:351–369, 1942.
- [85] Finale Doshi-Velez and Been Kim. Towards a rigorous science of interpretable machine learning. *arXiv: Machine Learning*, 2017.
- [86] Áine Driscoll, Seán Lyons, Franco Mariuzzo, and Richard S.J. Tol. Simulating demand for electric vehicles using revealed preference data. *Energy Policy*, 62(C):686–696, 2013.
- [87] EDPB. Edpb guidelines 04/2020 on the use of location data and contact tracing tools in the context of the covid-19 quoted, 2020.
- [88] EDPS. *Online manipulation and personal data*. 2018. [https://edps.europa.eu/data-protection/our-work/publications/opinions/online-manipulation-and-personal-data\\_en](https://edps.europa.eu/data-protection/our-work/publications/opinions/online-manipulation-and-personal-data_en).
- [89] Eiman Elbanhawy. The spatio-temporal analysis of the use and usability problems of ev workplace charging facilities. In Luis Romeral Martínez and Miguel Delgado Prieto, editors, *New Trends in Electrical Vehicle Powertrains*, chapter 9. IntechOpen, Rijeka, 2019.
- [90] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996.
- [91] George E. P. Box et al. *Time Series Analysis: Forecasting and Control*. John Wiley and Sons, 2015.
- [92] Mohammad Etemad, Amílcar Soares Júnior, Arazoo Hoseyni, Jordan Rose, and Stan Matwin. A trajectory segmentation algorithm based on interpolation-based change detection strategies. In *EDBT/ICDT Workshops*, 2019.
- [93] Mohammad Etemad, Amilcar Soares, Elham Etemad, Jordan Rose, Luis Torgo, and Stan Matwin. Sws: an unsupervised trajectory segmentation algorithm based on change detection with interpolation kernels. *GeoInformatica*, pages 1–21, 2020.
- [94] Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii Academiae Scientiarum Imperialis Petropolitanae*, 8:128–140, 1736.
- [95] R. Feynman. *The brownian movement*. 1964.

- [96] Data Flair. *Transfer Learning for Deep Learning with CNN*. 2018. <https://data-flair.training/blogs/transfer-learning/>.
- [97] Robin Flowerdew and Murray Aitkin. A method of fitting the gravity model based on the poisson distribution. *Journal of Regional Science*, 22(2):191–202, 1982.
- [98] Organisation for Economic Co-operation and Development. *The oecd privacy framework*. MIT Press, 2013. [https://www.oecd.org/sti/ieconomy/oecd\\_privacy\\_framework.pdf](https://www.oecd.org/sti/ieconomy/oecd_privacy_framework.pdf).
- [99] World Economic Forum. *The impact of digital content: Opportunities and risks of creating and sharing information online*. 2016. [https://www3.weforum.org/docs/GAC16/Social\\_Media\\_Impact\\_Digital.pdf](https://www3.weforum.org/docs/GAC16/Social_Media_Impact_Digital.pdf).
- [100] Thomas Franke, Nadine Rauh, Madlen Günther, Maria Trantow, and Josef F. Kreams. Which factors can protect against range stress in everyday usage of battery electric vehicles? toward enhancing sustainability of electric mobility systems. *Human Factors*, 58(1):13–26, 2016. PMID: 26646301.
- [101] Cheng Fu, Haosheng Huang, and Robert Weibel. Adaptive simplification of gps trajectories with geographic context – a quadtree-based approach. *International Journal of Geographical Information Science*, 35(4):661–688, 2021.
- [102] W. A. Fuller. *Introduction to Statistical Time Series*. John Wiley and Sons, 1976.
- [103] Benjamin Fung, ke Wang, Rui Chen, and Philip Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Comput. Surv.*, 42, 06 2010.
- [104] A Furnas. *Everything you wanted to know about data mining but were afraid to ask*. 2012. "<https://www.theatlantic.com/technology/archive/2012/04/everything-you-wanted-to-know-about-data-mining-but-were-afraid-to-ask/258538/>".
- [105] Andre Furtado, Areli Santos, Luis Alvares, Nikos Pelekis, and Vania Bogorny. Inferring relationships from trajectory data. *Simpósio Brasileiro de GeoInformática - GeoInfo*, 01 2015.
- [106] Lang Gabriel, Eric Marcon, and Florence Puech. Distance-based measures of spatial concentration: introducing a relative density function. *The Annals of Regional Science*, 64, 04 2020.
- [107] Alexis Galland and Marc Lelarge. Invariant embedding for graph classification. In *ICML 2019 Workshop on Learning and Reasoning with Graph-Structured Data*, Long Beach, United States, June 2019.

- [108] Francis Galton. *Vox populi*. Nature Publishing Group, 1907. <https://www.nature.com/articles/075450a0>.
- [109] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. 09 2014.
- [110] Feng Gao, Guy Wolf, and Matthew Hirn. Geometric scattering for graph data analysis. In *International Conference on Machine Learning*, pages 2122–2131. PMLR, 2019.
- [111] Mehrnaz Ghamami, Ali Zockaie, and Yu (Marco) Nie. A general corridor model for designing plug-in electric vehicle charging infrastructure to support intercity travel. *Transportation Research Part C: Emerging Technologies*, 68:389–402, 2016.
- [112] F. Giannotti and D. Pedreschi. *Mobility, Data Mining and Privacy: A Vision of Convergence*, pages 1–11. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [113] Fosca Giannotti et al. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *The VLDB Journal*, 20(5):695–719, October 2011.
- [114] Fosca Giannotti, G Giannotti, and Dino Pedreschi. *Mobility, data mining and privacy: Geographic knowledge discovery*. Springer, 01 2008.
- [115] Fosca Giannotti, Mirco Nanni, Dino Pedreschi, Fabio Pinelli, Chiara Renso, Salvatore Rinzivillo, and Roberto Trasarti. Unveiling the complexity of human mobility by querying and mining massive trajectory data. *The VLDB Journal*, 20:695–719, 2011.
- [116] Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 330–339, 2007.
- [117] Lei Gong, Toshiyuki Yamamoto, and Takayuki Morikawa. Identification of activity stop locations in gps trajectories by dbscan-te method combined with support vector machines. *Transportation Research Procedia*, 32:146–154, 2018.
- [118] Marta C. Gonzalez, Cesar A. Hidalgo, et al. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, June 2008.
- [119] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [120] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. *CoRR*, abs/1607.00653, 2016.

- [121] Qihang Gu, Dimitris Sacharidis, Michael Mathioudakis, and Gang Wang. Inferring venue visits from gps trajectories. In *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, SIGSPATIAL '17, New York, NY, USA, 2017. Association for Computing Machinery.
- [122] Riccardo Guidotti and Paolo Cintia. Towards a boosted route planner using individual mobility models. In *International Conference on Software Engineering and Formal Methods*, pages 108–123. Springer, 2015.
- [123] Riccardo Guidotti, Anna Monreale, Salvatore Rinzivillo, Dino Pedreschi, and Fosca Giannotti. Unveiling mobility complexity through complex network analysis. *Social Network Analysis and Mining*, 6:1–21, 2016.
- [124] Riccardo Guidotti, Anna Monreale, Salvatore Ruggieri, Dino Pedreschi, Franco Turini, and Fosca Giannotti. Local rule-based explanations of black box decision systems. *ArXiv*, abs/1805.10820, 2018.
- [125] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. A survey of methods for explaining black box models. *ACM Computing Surveys (CSUR)*, 51:1 – 42, 2018.
- [126] Riccardo Guidotti and Mirco Nanni. Crash prediction and risk assessment with individual mobility networks. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 89–98. IEEE, 2020.
- [127] Riccardo Guidotti, Mirco Nanni, Salvatore Rinzivillo, Dino Pedreschi, and Fosca Giannotti. Never drive alone: Boosting carpooling with network analysis. *Information Systems*, 64:237–257, 2017.
- [128] Riccardo Guidotti, Roberto Trasarti, and Mirco Nanni. Tosca: two-steps clustering algorithm for personal locations detection. In *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 38. ACM, 2015.
- [129] Riccardo Guidotti, Roberto Trasarti, Mirco Nanni, Fosca Giannotti, and Dino Pedreschi. There’s a path for everyone: A data-driven personal model reproducing mobility agendas. In *2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 303–312. IEEE, 2017.
- [130] David Gunning and David Aha. Darpa’s explainable artificial intelligence (xai) program. *AI Magazine*, 40:44–58, 06 2019.
- [131] Sini Guo, Xiang Li, Wai-Ki Ching, Ralescu Dan, Wai-Keung Li, and Zhiwen Zhang. Gps trajectory data segmentation based on probabilistic logic. *International Journal of Approximate Reasoning*, 103:227–247, 2018.

- [132] Xiawei Guo, Quanming Yao, Yuqiang Chen, Wenyuan Dai, and Qiang Yang. Privacy-preserving transfer learning for knowledge sharing. 11 2018.
- [133] Armitage H. *Stanford Medicine scientists hope to use data from wearable devices to predict illness, including COVID-19.* Stanford Medicine, 2020. <https://med.stanford.edu/news/all-news/2020/04/wearable-devices-for-predicting-illness-.html>.
- [134] Xiao-Pu Han, Qiang Hao, Bing-Hong Wang, and Tao Zhou. Origin of the scaling law in human mobility: hierarchy of traffic systems. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 83 3 Pt 2:036117, 2011.
- [135] Gabriel Handler and Israel Zang. A dual algorithm for the constrained shortest path problem. *Networks*, 10:293 – 309, 10 2006.
- [136] Scott Hardman, Alan Jenn, Gil Tal, Jonn Axsen, George Beard, Nicolò Daina, Erik Figenbaum, Niklas Jakobsson, Patrick Jochem, Neale Kinnear, Patrick Plötz, Jose Pontes, Nazir Refa, Frances Sprei, Tom Turrentine, and Bert Witkamp. A review of consumer preferences of and interactions with electric vehicle charging infrastructure. *Transportation Research Part D: Transport and Environment*, 62:508–523, 07 2018.
- [137] Samiul Hasan, Xianyuan Zhan, and Satish V. Ukkusuri. Understanding urban human activity and mobility patterns using large-scale location-based data from online social media. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing, UrbComp '13*, New York, NY, USA, 2013. Association for Computing Machinery.
- [138] Fang He, Yafeng Yin, and Jing Zhou. Deploying public charging stations for electric vehicles on urban road networks. *Transportation Research Part C: Emerging Technologies*, 60:227–240, 2015.
- [139] Mireille Hildebrandt. *Defining Profiling: A New Type of Knowledge?*, pages 17–45. 01 2008.
- [140] Johan Himberg, Kalle Korpiaho, Heikki Mannila, Johanna Tikanmaki, and Hannu TT Toivonen. Time series segmentation for context recognition in mobile devices. In *Proceedings 2001 IEEE International Conference on Data Mining*, pages 203–210. IEEE, 2001.
- [141] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.
- [142] Horizon2020. Track and know- in horizon project.
- [143] Shima Hosseinpour, Hongyi Chen, and Hua Tang. Barriers to the wide adoption of electric vehicles: A literature review based discussion. In *2015 Portland International Conference on Management of Engineering and Technology (PICMET)*, pages 2329–2336, 2015.



- [144] Jiabo Huang and Shaogang Gong. Unsupervised transfer learning with self-supervised remedy. *ArXiv*, abs/2006.04737, 2020.
- [145] Zheng Huo, Xiaofeng Meng, Haibo Hu, and Yi Huang. You can walk alone: Trajectory privacy-preserving through significant stays protection. In Sang-goo Lee, Zhiyong Peng, Xiaofang Zhou, Yang-Sae Moon, Rainer Unland, and Jaesoo Yoo, editors, *Database Systems for Advanced Applications*, pages 351–366, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [146] Sungsoo Hwang, Christian Evans, and Timothy Hanke. Detecting stop episodes from gps trajectories with gaps. In *Seeing Cities Through Big Data*, pages 427–439. Springer, 2017.
- [147] Chidubem Iddianozie and Gavin McArdle. A transfer learning paradigm for spatial networks. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, SAC '19, page 659–666, New York, NY, USA, 2019. Association for Computing Machinery.
- [148] Alexander T Ihler and Padhraic Smyth. Learning time-intensity profiles of human activity using non-parametric bayesian models. *NIPS*, 19, 2007.
- [149] Masahiko Itoh, Daisaku Yokoyama, Masashi Toyoda, Yoshimitsu Tomita, Satoshi Kawamura, and Masaru Kitsuregawa. Visual exploration of changes in passenger flows and tweets on mega-city metro network. *IEEE Transactions on Big Data*, 2:85–99, 2016.
- [150] Sergey Ivanov and Evgeny Burnaev. Anonymous walk embeddings. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2191–2200, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.
- [151] Zahedeh Izakian, M Saadi Mesgari, and Robert Weibel. A feature extraction based trajectory segmentation approach based on multiple movement parameters. *Engineering Applications of Artificial Intelligence*, 88:103394, 2020.
- [152] L Kirchner J Larson, S Mattu and J Angwin. How we analyzed the compas recidivism algorithm. In *Propublica*, 2016.
- [153] Karim Keramat Jahromi, Matteo Zignani, Sabrina Gaito, and Gian Paolo Rossi. Simulating human mobility patterns in urban areas. *Simulation Modelling Practice and Theory*, 62:137–156, 2016.
- [154] M. K. Jat, P. K. Garg, and D. Khare. Modelling of urban growth using spatial analysis techniques: a case study of ajmer city (india). *International Journal of Remote Sensing*, 29(2):543–567, 2008.

- [155] N. Jewell, M. Turner, J. Naber, and M. McIntyre. Analysis of forecasting algorithms for minimization of electric demand costs for electric vehicle charging in commercial and industrial environments. In *2012 IEEE Transportation Electrification Conference and Expo (ITEC)*, pages 1–6, June 2012.
- [156] Bin Jiang, Junjun Yin, and Sijian Zhao. Characterizing the human mobility pattern in a large street network. *Physical review. E, Statistical, nonlinear, and soft matter physics*, 80:021136, 08 2009.
- [157] Weiwei Jiang and Jiayun Luo. Big data for traffic estimation and prediction: A survey of data and tools. *Applied System Innovation*, 5(1), 2022.
- [158] Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey. *Expert Syst. Appl.*, 207(C), nov 2022.
- [159] Derick A. Johnson and Mohan M. Trivedi. Driving style recognition using a smartphone as a sensor platform. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1609–1615, 2011.
- [160] Dan Jones. Conflict resolution: Wars without end. *Nature*, 519:148–50, 03 2015.
- [161] Woo-Sung Jung, Fengzhong Wang, and H. Stanley. Gravity model in the korean highway. *EPL (Europhysics Letters)*, 81, 10 2007.
- [162] Amílcar Soares Júnior et al. Grasp-uts: an algorithm for unsupervised trajectory segmentation. *International Journal of Geographical Information Science*, 29(1):46–68, 2015.
- [163] Amílcar Soares Júnior et al. A semi-supervised approach for the semantic segmentation of trajectories. In *19th IEEE International Conference on Mobile Data Management (MDM)*, pages 145–154, 2018.
- [164] Haik Kalantarian, Nabil Alshurafa, and Majid Sarrafzadeh. Detection of gestures associated with medication adherence using smartwatch-based inertial sensors. *IEEE Sensors Journal*, 16:1–1, 01 2015.
- [165] Chaogui Kang, Xiujun Ma, Daoqin Tong, and Yu Liu. Intra-urban human mobility patterns: An urban morphology perspective. *Physica A: Statistical Mechanics and its Applications*, 391(4):1702–1717, 2012.
- [166] KIA Corporation. How to extend EV battery life? <https://www.kia.com/dm/discover-kia/ask/how-to-extend-ev-battery-life.html>. Retrieved on Jan 2023.
- [167] Young-Jun Kweon et al. Development of crash prediction models with individual vehicular data. *Transportation research part C: emerging technologies*, 19(6):1353–1363, 2011.

- [168] Victor Lavrenko, Matt Schmill, Dawn Lawrie, Paul Ogilvie, David Jensen, and James Allan. Mining of concurrent text and time series. In *KDD-2000 Workshop on Text Mining*, volume 2000, pages 37–44, 2000.
- [169] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014.
- [170] Scikit Learn. *Keras’s documentation*. <https://scikit-learn.org/stable/>,.
- [171] Kenneth Lebeau, Philippe Lebeau, Olivier Mairesse, Cathy Macharis, and Joeri Van Mierlo. Consumer attitudes towards battery electric vehicles: A large-scale survey. *Int. J. Electric and Hybrid Vehicles*, 5:28–41, 04 2013.
- [172] Chris Lee, Bruce Hellinga, and Frank Saccomanno. Real-time crash prediction model for application to crash prevention in freeway traffic. *Transportation Research Record*, 1840(1):67–77, 2003.
- [173] Jae-Gil Lee et al. Trajectory clustering: A partition-and-group framework. In *ACM SIGMOD*, page 593–604. ACM, 2007.
- [174] Seungmin Lee, Dongwan Kim, Namil Kim, and Seong-Gyun Jeong. Drop to adapt: Learning discriminative features for unsupervised domain adaptation, 2019.
- [175] Sungjun Lee et al. Next place prediction based on spatiotemporal pattern mining of mobile device logs. *Sensors*, 16(2):145, 2016.
- [176] Luis Leiva and Enrique Vidal. Warped k-means: An algorithm to cluster sequentially-distributed data. *Information Sciences*, 237:196–210, 07 2013.
- [177] AA Leman and B Weisfeiler. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968.
- [178] Jundong Li, Liang Wu, and Huan Liu. Multi-level network embedding with boosted low-rank matrix approximation. *CoRR*, abs/1808.08627, 2018.
- [179] Shengyin Li, Yongxi Huang, and Scott J. Mason. A multi-period optimization model for the deployment of public electric vehicle charging stations on network. *Transportation Research Part C: Emerging Technologies*, 65:128–143, 2016.
- [180] Wentian Li. Dna segmentation as a model selection process. In *Proceedings of the fifth annual international conference on Computational biology*, pages 204–210. ACM, 2001.
- [181] Xiao Liang, Jichang Zhao, Li Dong, and Ke Xu. Unraveling the origin of exponential law in intra-urban human mobility. *Scientific reports*, 3:2983, 05 2013.

- [182] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Attributed social network embedding. *CoRR*, abs/1705.04969, 2017.
- [183] Yuan Liao. Understanding human mobility with emerging data sources: Validation, spatiotemporal patterns, and transport modal disparity. 2020.
- [184] Chris Lilly. *EV charging connector types*. 2022. <https://www.zap-map.com/charge-points/connectors-speeds/>.
- [185] Zachary Chase Lipton. The mythos of model interpretability. *Communications of the ACM*, 61:36 – 43, 2016.
- [186] Yixin Liu, Ming Jin, Shirui Pan, Chuan Zhou, Yu Zheng, Feng Xia, and Philip S. Yu. Graph self-supervised learning: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 35(6):5879–5900, 2023.
- [187] Z. Liu, Z. Li, K. Wu, and M. Li. Urban traffic prediction from mobility data using deep learning. *IEEE Network*, 32(4):40–46, 2018.
- [188] Zhaoyang Liu, Yanyan Shen, and Yanmin Zhu. Where will dockless shared bikes be stacked? — parking hotspots detection in a new city. In *Proc. of the 24th ACM SIGKDD, KDD '18*, page 566–575, New York, NY, USA, 2018. ACM.
- [189] Eu Law Live. "european data protection supervisor appeals general court judgment in data protection case against single resolution board", 08 "2023".
- [190] Juliana Lockman, Robert S. Fisher, and Donald M. Olson. Detection of seizure-like movements using a wrist accelerometer. *Epilepsy Behavior*, 20(4):638 – 641, 2011.
- [191] Leonardo Longhi and Mirco Nanni. Car telematics big data analytics for insurance and innovative mobility services. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–11, 2019.
- [192] Dominique Lord and Fred Mannering. The statistical analysis of crash-frequency data: a review and assessment of methodological alternatives. *Transportation research part A: policy and practice*, 44(5):291–305, 2010.
- [193] Massimiliano Luca, Gianni Barlacchi, Bruno Lepri, and Luca Pappalardo. A survey on deep learning for human mobility. *ACM Comput. Surv.*, 55(1), nov 2021.
- [194] Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *ArXiv*, abs/1705.07874, 2017.
- [195] Kees Maat, Bert van Wee, and Dominic Stead. Land use and travel behaviour: Expected effects from the perspective of utility theory and activity-based theories. *Environment and Planning B: Planning and Design*, 32(1):33–46, 2005.

- [196] Amin Mahmoudzadeh Andwari, Apostolos Pesiridis, Srithar Rajoo, Ricardo Martinez-Botas, and Vahid Esfahanian. A review of battery electric vehicle technology and readiness levels. *Renewable and Sustainable Energy Reviews*, 78:414–430, 2017.
- [197] Gianclaudio Malgieri and Giovanni Comandé. Why a right to legibility of automated decision-making exists in the general data protection regulation. *International Data Privacy Law*, 7:243–265, 2017.
- [198] Gianclaudio Malgieri and Giovanni Comandé. Sensitive-by-distance: quasi-health data in the algorithmic era. *Information Communications Technology Law*, 26:1–21, 06 2017.
- [199] Matteo Manca, Ludovico Boratto, Victor Morell Roman, Oriol Martori i Gallissà, and Andreas Kaltenbrunner. Using social media to characterize urban mobility patterns: State-of-the-art survey and case-study. *Online Social Networks and Media*, 1:56–69, 2017.
- [200] Richard Mann, Allan D Jepson, and Thomas El-Maraghi. Trajectory segmentation using dynamic programming. In *Object recognition supported by user interaction for service robots*, volume 1, pages 331–334. IEEE, 2002.
- [201] Fred L Mannering and Chandra R Bhat. Analytic methods in accident research: Methodological frontier and future directions. *Analytic methods in accident research*, 1:1–22, 2014.
- [202] Cesare Marchetti. Anthropological invariants in travel behavior. *Technological Forecasting and Social Change*, 47:75–88, 1994.
- [203] Zvika Marx, Michael Rosenstein, Leslie Kaelbling, and Thomas Dietterich. Transfer learning with an ensemble of background tasks. 01 2022.
- [204] A Paolo Masucci, Joan Serras, Anders Johansson, and Michael Batty. Gravity versus radiation models: On the importance of scale and heterogeneity in commuting flows. *Physical Review E*, 88(2):022812, 2013.
- [205] Laszlo Matyas. Proper econometric specification of the gravity model. *The World Economy*, 20(3):363–368, 1997.
- [206] F. McSherry. *Statistical Inference considered harmful*. MIT Press, 2016. <https://github.com/frankmcsherry/blog/blob/master/posts/2016-06-14>.
- [207] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., USA, 1 edition, 1997.
- [208] Yves-Alexandre Montjoye, Cesar Hidalgo, Michel Verleysen, and Vincent Blondel. Unique in the crowd: The privacy bounds of human mobility. *Scientific reports*, 3:1376, 03 2013.

- [209] P. A. P. Moran. Notes on continuous stochastic phenomena. *Biometrika*, 37(1/2):17–23, 1950.
- [210] G. Morton. A computer oriented geodetic data base and a new technique in file sequencing. In *IBM Research Report*, 1966.
- [211] Mirco Nanni, Agnese Bonavita, and Riccardo Guidotti. City indicators for mobility data mining. In *Big Mobility Data Analytics (BMDA)*. CEUR, 2021.
- [212] Mirco Nanni, Riccardo Guidotti, Agnese Bonavita, and Omid Isfahani Alamdari. City indicators for geographical transfer learning: an application to crash prediction. *Geoinformatica*, 2022.
- [213] Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*, 2017.
- [214] Elahe Nasiri, Kamal Berahmand, Mehrdad Rostami, and Mohammad Dabiri. A novel link prediction algorithm for protein-protein interaction networks by attributed graph embedding. *Computers in Biology and Medicine*, 137:104772, 2021.
- [215] Mehmet Nergiz, Maurizio Atzori, and Yucel Saygin. Towards trajectory anonymization: A generalization-based approach. *Proceedings of the SIGSPATIAL ACM GIS 2008 International Workshop on Security and Privacy in GIS and LBS, SPRINGL'08*, pages 52–61, 01 2008.
- [216] Newmotion - Shell EV Charging Solutions UK Ltd. EV Driver Survey Report 2020, 2020.
- [217] Paul Newson and John Krumm. Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 336–343. ACM, 2009.
- [218] Ming Ni, Qing He, and Jing Gao. Forecasting the subway passenger flow under event occurrences with social media. *IEEE Transactions on Intelligent Transportation Systems*, 18:1623–1632, 2017.
- [219] Michael A Nicholas, Gil Tal, and Justin Woodjack. California statewide charging assessment model for plug-in electric vehicles: Learning from statewide travel surveys. 2013.
- [220] Anastasios Noulas, Salvatore Scellato, Renaud Lambiotte, Massimiliano Pontil, and Cecilia Mascolo. Correction: A tale of many cities: Universal patterns in human urban mobility. *PLOS ONE*, 7(9):null, 09 2012.

- [221] Directorate General of Human Rights and Rule of Law (Council of Europe). Guidelines on the protection of individuals with regard to the processing of personal data in a world of big data. 2017.
- [222] Government Digital Service Cabinet Office. Data ethics framework. 2020.
- [223] Keith Ord. Outliers in statistical data: V. Barnett and T. Lewis, 1994, 3rd edition, (John Wiley Sons, Chichester), 584 pp., [UK Pound]55.00, ISBN 0-471-93094-6. *International Journal of Forecasting*, 12(1):175–176, 1996.
- [224] Mingdong Ou, Peng Cui, Jian Pei, Ziwei Zhang, and Wenwu Zhu. Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1105–1114, New York, NY, USA, 2016. Association for Computing Machinery.
- [225] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2009.
- [226] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.
- [227] S.J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.*, 22(10):1345–1359, 2010.
- [228] Cecilia Panigutti, Dino Pedreschi, and Alan Perotti. Doctor xai: an ontology-based approach to black-box sequential data classification explanations. *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 01 2020.
- [229] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. 2017.
- [230] Luca Pappalardo. *Human Mobility, Social Networks and Economic Development: a Data Science perspective*. PhD thesis, 12 2014.
- [231] Luca Pappalardo et al. Returners and explorers dichotomy in human mobility. *Nature communications*, 6:8166, 2015.
- [232] Luca Pappalardo, Salvatore Rinzivillo, Zehui Qu, Dino Pedreschi, and Fosca Giannotti. Understanding the patterns of car travel. *The European Physical Journal Special Topics*, 215(1):61–73, Jan 2013.
- [233] Luca Pappalardo and Filippo Simini. Modelling individual routines and spatio-temporal trajectories in human mobility. *arXiv:1607.05952*, 2016.

- [234] Luca Pappalardo, Filippo Simini, S Rinzivillo, Dino Pedreschi, Fosca Giannotti, and Albert-Laszlo Barabasi. Returners and explorers dichotomy in human mobility. *Nat Commun*, 6, 09 2015.
- [235] Adam Pavliček, Jan Pačes, Oliver Clay, and Giorgio Bernardi. A compact view of isochores in the draft human genome sequence. *FEBS letters*, 511(1-3):165–169, 2002.
- [236] Judea Pearl. Theoretical impediments to machine learning with seven sparks from the causal revolution. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018.
- [237] Karl Pearson. Notes on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, 58:240–242, 1895.
- [238] Dino Pedreschi et al. Meaningful explanations of black box ai decision systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9780–9784, 2019.
- [239] Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *ICML*, 2000.
- [240] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. *CoRR*, abs/1403.6652, 2014.
- [241] Bryan Perozzi, Vivek Kulkarni, and Steven Skiena. Walklets: Multiscale graph embeddings for interpretable network classification. *CoRR*, abs/1605.02115, 2016.
- [242] Laurent Perron and Vincent Furnon. *OR-Tools*. Google, 2022. <https://developers.google.com/optimization/>.
- [243] Robert Pietracho, Christoph Wenge, Stephan Balischewski, Pio Lombardi, Przemyslaw Komarnicki, Leszek Kasprzyk, and Damian Burzyński. Potential of using medium electric vehicle fleet in a commercial enterprise transport in germany on the basis of real-world gps data. *Energies*, 14(17), 2021.
- [244] Omer Tene Jules Polonetsky. Big data for all: Privacy and user control in the age of analytics. *J. TECH. INTELL. PROP*, pages 239–270, 2013.
- [245] S. Porta, P. Crucitti, and V. Latora. Centrality measures in spatial networks of urban streets. *Physical Review E*, 73(3, part 2):036125–1, 3 2006.
- [246] Francesca Pratesi, Anna Monreale, R. Trasarti, Fosca Giannotti, Dino Pedreschi, and T. Yanagihara. Prudence: A system for assessing privacy risk vs utility in data sharing ecosystems. *Transactions on Data Privacy*, 11:139–167, 08 2018.



- [247] Pollution Probe. Assessment of the consumer electric vehicle charging experience in canada – final project report, 2022.
- [248] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. Knock knock, who’s there? membership inference on aggregate location data. *CoRR*, abs/1708.06145, 2017.
- [249] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. Network embedding as matrix factorization. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, feb 2018.
- [250] Azizur Rahim, Tie Qiu, Zhaolong Ning, Jinzhong Wang, Noor Ullah, Amr Tolba, and Feng Xia. Social acquaintance based routing in vehicular social networks. *Future Generation Computer Systems*, 93:751–760, 2019.
- [251] Rajat Raina, Andrew Y. Ng, and Daphne Koller. Constructing informative priors using transfer learning. In *Proceedings of the 23rd International Conference on Machine Learning, ICML ’06*, page 713–720, New York, NY, USA, 2006. Association for Computing Machinery.
- [252] Vasily E Ramensky, V Ju Makeev, Mikhail A. Roytberg, and Vladimir G Tumanyan. Dna segmentation through the bayesian approach. *Journal of Computational Biology*, 7(1-2):215–231, 2000.
- [253] Rationality. *Oxford Dictionary of English*. Oxford University Press, 2015.
- [254] Samaneh Rezaei, Jafar Tahmoresnezhad, and Vahid Solouk. A transductive transfer learning approach for image classification. *Int. J. Mach. Learn. Cybern.*, 12:747–762, 2021.
- [255] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. *arXiv*, 1602.04938, 2016.
- [256] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, 2018.
- [257] Salvatore Rinzivillo, Lorenzo Gabrielli, Mirco Nanni, Luca Pappalardo, Dino Pedreschi, and Fosca Giannotti. The purpose of motion: Learning activities from individual mobility networks. In *2014 International Conference on Data Science and Advanced Analytics (DSAA)*, pages 312–318. IEEE, 2014.
- [258] P.A. Rogerson. *Statistical Methods for Geography: A Student’s Guide*. SAGE Publications, 2010.
- [259] Benedek Rozemberczki, Carl Allen, and Rik Sarkar. Multi-scale attributed node embedding. *CoRR*, abs/1909.13021, 2019.

- [260] Benedek Rozemberczki, Oliver Kiss, and Rik Sarkar. Karate Club: An API Oriented Open-source Python Framework for Unsupervised Learning on Graphs. In *Procs. of the 29th ACM Int. Conf. on Information and Knowledge Management (CIKM '20)*, page 3125–3132. ACM, 2020.
- [261] Benedek Rozemberczki and Rik Sarkar. Characteristic functions on graphs: Birds of a feather, from statistical descriptors to parametric models. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1325–1334, 2020.
- [262] Benedek Rozemberczki and Rik Sarkar. Fast sequence-based embedding with diffusion graphs. *CoRR*, abs/2001.07463, 2020.
- [263] Alan Rubel. The black box society: The secret algorithms that control money and information, by frank pasquale. cambridge: Harvard university press, 2015. 320 pp. isbn 978–0674368279. *Business Ethics Quarterly*, 26:568–571, 10 2016.
- [264] Sebastian. Ruder. *Transfer Learning - Machine Learning's Next Frontier*. 2017. <http://ruder.io/transfer-learning/>.
- [265] Meead Saberi, Hani S. Mahmassani, Dirk Brockmann, and Amir Hosseini. A complex network perspective for characterizing urban travel demand patterns: graph theoretical analysis of large-scale origin–destination demand networks. *Transportation*, 44(6):1383–1402, November 2017.
- [266] Martin Sachenbacher, Martin Leucker, Andreas Artmeier, and Julian Haselmayr. Efficient energy-optimal routing for electric vehicles. In *AAAI*, 2011.
- [267] Hachem Saddiki and Laura B. Balzer. A primer on causality in data science. *arXiv: Applications*, 2018.
- [268] Hamid Safi, Behrang Assemi, Mahmoud Mesbah, and Luis Ferreira. Trip detection with smartphone-assisted collection of travel data. *Transportation Research Record*, 2594(1):18–26, 2016.
- [269] Flora Dilys Salim, Seng Wai Loke, Andry Rakotonirainy, Bala Srinivasan, and Shonali Krishnaswamy. Collision pattern modeling and real-time collision detection at road intersections. In *2007 IEEE Intelligent Transportation Systems Conference*, pages 161–166. IEEE, 2007.
- [270] Salvatore Scellato, Anastasios Noulas, Renaud Lambiotte, and Cecilia Mascolo. Socio-spatial properties of online location-based social networks. In *ICWSM*, 2011.
- [271] Jochen Schiller and Agnès Voisard. *Location-based services*. Elsevier, 2004.

- [272] Raz Schwartz and Germaine R Halegoua. The spatial self: Location-based identity performance on social media. *New Media & Society*, 17(10):1643–1660, 2015.
- [273] Maximilian Schücking, Patrick Jochem, Wolf Fichtner, Olaf Wollersheim, and Kevin Stella. Charging strategies for economic operations of electric vehicles in commercial applications. *Transportation Research Part D: Transport and Environment*, 51:173–189, 2017.
- [274] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 7 1948.
- [275] Sulochana Shekhar. Urban sprawl assessment entropy approach. *GIS Development*, 2004, Vol 8 issue 5, Page ., 6 Pages:43 – 48, 05 2004.
- [276] Robin Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *Comput. J.*, 16:30–34, 1973.
- [277] Katarzyna Siła-Nowicka, Jan Vandrol, Taylor Oshan, Jed A Long, Urška Demšar, and A Stewart Fotheringham. Analysis of human mobility patterns from gps trajectories and contextual information. *International Journal of Geographical Information Science*, 30(5):881–906, 2016.
- [278] Filippo Simini, Marta C. Gonzalez, Amos Maritan, et al. A universal model for mobility and migration patterns. *Nature*, 484(7392):96–100, 2012.
- [279] Filippo Simini, Marta C. González, Amos Maritan, and Albert-László Barabási. A universal model for mobility and migration patterns. *Nature*, 484(7392):96–100, Feb 2012.
- [280] Katarzyna Siła-Nowicka, Jan Vandrol, Taylor Oshan, Jed A. Long, Urška Demšar, and A. Stewart Fotheringham. Analysis of human mobility patterns from gps trajectories and contextual information. *International Journal of Geographical Information Science*, 30(5):881–906, 2016.
- [281] Chaoming Song, Tal Koren, Pu Wang, and Albert-László Barabási. Modelling the scaling properties of human mobility. *Nature Physics*, 6(10):818–823, Sep 2010.
- [282] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.
- [283] T. Stough, N. Cressie, E.L. Kang, et al. Spatial analysis and visualization of global data on multi-resolution hexagonal grids. *Japanese Journal of Statistics and Data Science*, 3:107–128, 2020.

- [284] Martin Strehler, Sören Merting, and Christian Schwan. Energy-efficient shortest routes for electric and hybrid vehicles. *Transportation Research Part B: Methodological*, 103:111–135, 2017. Green Urban Transportation.
- [285] Steven Strogatz. Strogatz, s.h.: Exploring complex networks. nature 410, 268. *Nature*, 410:268–76, 04 2001.
- [286] Camilo Suarez and Wilmar Martinez. Fast and ultra-fast charging for battery electric vehicles – a review. In *2019 IEEE Energy Conversion Congress and Exposition (ECCE)*, pages 569–575, 2019.
- [287] Charles Sutton and Andrew McCallum. Composition of conditional random fields for transfer learning. Technical report, MASSACHUSETTS UNIV AMHERST DEPT OF COMPUTER SCIENCE, 2005.
- [288] Timothy Sweda, Irina Dolinskaya, and Diego Klabjan. Adaptive routing and recharging policies for electric vehicles. *Transportation Science*, 51, 03 2017.
- [289] Timothy Sweda and Diego Klabjan. Finding minimum-cost paths for electric vehicles. pages 1–4, 03 2012.
- [290] Vasileios Syrris, Ondrej Pesek, and Pierre Soille. Satimnet: Structured and harmonised training data for enhanced satellite imagery classification. *Remote Sensing*, 12:3358, 10 2020.
- [291] Tan, Pang-Ning, Michael Steinbach, Michael Adeyeye Oshin, Vipin Kumar, and Vipin. *Introduction to Data Mining*. 05 2005.
- [292] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In Věra Kůrková, Yannis Manolopoulos, Barbara Hammer, Lazaros Iliadis, and Ilias Maglogiannis, editors, *Artificial Neural Networks and Machine Learning – ICANN 2018*, pages 270–279, Cham, 2018. Springer International Publishing.
- [293] Pang-Ning Tan et al. *Introduction to data mining*, 2005.
- [294] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2018.
- [295] Mingyue Tang, Pan Li, and Carl Yang. Graph auto-encoder via neighborhood wasserstein reconstruction. In *International Conference on Learning Representations*, 2022.
- [296] K. Tehrani and A. Michael. Wearable technology and wearable devices: Everything you need to know. *Wearable Devices Magazine*, 2014.
- [297] Evimaria Terzi and Panayiotis Tsaparas. Efficient algorithms for sequence segmentation. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 316–327. SIAM, 2006.

- [298] Benoit Thierry, Basile Chaix, and Yan Kestens. Detecting activity locations from raw gps data: a novel kernel-based algorithm. *International journal of health geographics*, 12(1):1–10, 2013.
- [299] Edison Thomaz, Irfan Essa, and Gregory D. Abowd. A practical approach for recognizing eating moments with wrist-mounted inertial sensing. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '15*, page 1029–1040, New York, NY, USA, 2015. Association for Computing Machinery.
- [300] Christian Tilk, Ann-Kathrin Rothenbächer, Timo Gschwind, and Stefan Irnich. Asymmetry matters: Dynamic half-way points in bidirectional labeling for solving shortest path problems with resource constraints faster. *European Journal of Operational Research*, 261(2):530–539, 2017.
- [301] Eran Toch, Boaz Lerner, Eyal Ben Zion, and Irad Ben-Gal. Analyzing large-scale human mobility data: a survey of machine learning methods and applications. *Knowledge and Information Systems*, 58:501–523, 2019.
- [302] J Topping. Investigations on the theory of the brownian movement. *Physics Bulletin*, 7(10):281–281, oct 1956.
- [303] Leo Torres, Kevin S. Chan, and Tina Eliassi-Rad. Geometric laplacian eigenmap embedding. *CoRR*, abs/1905.09763, 2019.
- [304] Lisa A. Torrey and Jude W. Shavlik. Chapter 11 transfer learning. 2009.
- [305] R. Trasarti, Riccardo Guidotti, Anna Monreale, and Fosca Giannotti. Myway: Location prediction via mobility profiling. *Information Systems*, 64, 11 2015.
- [306] Roberto Trasarti, Riccardo Guidotti, Anna Monreale, and Fosca Giannotti. Myway: Location prediction via mobility profiling. *Information Systems*, 64:350–367, 2017.
- [307] Roberto Trasarti, Fabio Pinelli, Mirco Nanni, and Fosca Giannotti. Mining mobility user profiles for car pooling. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1190–1198. ACM, 2011.
- [308] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alex Bronstein, and Emmanuel Müller. Netlsd: Hearing the shape of a graph. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '18*, 2018.
- [309] Wei Tu, Qingquan Li, Zhixiang Fang, Shih lung Shaw, Baoding Zhou, and Xiaomeng Chang. Optimizing the locations of electric taxi charging stations: A spatial–temporal demand coverage approach. *Transportation Research Part C: Emerging Technologies*, 65:172–189, 2016.

- [310] Juan Van Roy, Niels Leemput, Sven De Breucker, Frederik Geth, Peter Tant, and Johan Driesen. An availability analysis and energy consumption model for a flemish fleet of electric vehicles. In *EEVC European Electric Vehicle Congress*, 2011.
- [311] M. van Steen. *Graph Theory and Complex Networks: An Introduction*. Maarten van Steen, 2010. <https://books.google.it/books?id=V7bMbwAACAA>.
- [312] Saurabh Verma and Zhi-Li Zhang. Hunt for the unique, stable, sparse and fast feature learning on graphs. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [313] Paul Voigt and Axel von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer Publishing Company, Incorporated, 1st edition, 2017.
- [314] Sandra Wachter and B. Mittelstadt. A right to reasonable inferences: Re-thinking data protection law in the age of big data and ai. *Columbia Business Law Review*, 2019:494–620–494–620, 2018.
- [315] Dong Wang, Sven Thunéll, Ulrika Lindberg, Lili Jiang, Johan Trygg, and Mats Tysklind. Towards better process management in wastewater treatment plants: Process analytics based on shap values for tree-based machine learning methods. *Journal of Environmental Management*, 301:113941, 2022.
- [316] Hao Wang et al. Location recommendation in location-based social networks using user check-in data. In *SIGSPATIAL*, pages 374–383, 2013.
- [317] Jinjun Wang, Wei Xu, and Yihong Gong. Real-time driving danger level prediction, November 23 2010. US Patent 7,839,292.
- [318] Jinzhong Wang, Xiangjie Kong, Feng Xia, and Lijun Sun. Urban human mobility: Data-driven modeling and prediction. *ACM SIGKDD Explorations Newsletter*, 21:1–19, 05 2019.
- [319] Lili Wang, Chenghan Huang, Weicheng Ma, Xinyuan Cao, and Soroush Vosoughi. Graph embedding via diffusion-wavelets-based node feature distribution characterization. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 3478–3482, 2021.
- [320] Xin Wang, Asad J. Khattak, Jun Liu, Golnush Masghati-Amoli, and Sanghoon Son. What is the level of volatility in instantaneous driving decisions? *Transportation Research Part C: Emerging Technologies*, 58:413–427, 2015. Big Data in Transportation and Traffic Engineering.

- [321] Yaqing Wang, Chunyan Feng, Ling Chen, Hongzhi Yin, Caili Guo, and Yunfei Chu. User identity linkage across social networks via linked heterogeneous network embedding. *World Wide Web*, pages 1–22, 2018.
- [322] Yibo Wang, Wei Xu, Yiqun Zhang, Yu Qin, Wenping Zhang, and Xue Wu. Machine learning methods for driving risk prediction. In *Proceedings of the 3rd ACM SIGSPATIAL Workshop on Emergency Management using*, page 10. ACM, 2017.
- [323] Steven E. Whang. *Data Analytics: Integration and Privacy*. Computer Science Department, Stanford University, 06 2012.
- [324] Wikipedia contributors. Analytics — Wikipedia, the free encyclopedia. <https://en.wikipedia.org/w/index.php?title=Analytics&oldid=1059577376>, 2021. [Online; accessed 10-December-2021].
- [325] Wikipedia contributors. General data protection regulation — Wikipedia, the free encyclopedia. [https://en.wikipedia.org/w/index.php?title=General\\_Data\\_Protection\\_Regulation&oldid=1062419012](https://en.wikipedia.org/w/index.php?title=General_Data_Protection_Regulation&oldid=1062419012), 2021. [Online; accessed 2-January-2022].
- [326] Wikipedia contributors. Lévy flight — Wikipedia, the free encyclopedia, 2021. [Online; accessed 27-December-2021].
- [327] Els Kindt Vanfleteren Michaël Wim Jan Schreurs, Mireille Hildebrandt. Cogitas, ergo sum. the role of data protection law and non-discrimination law in group profiling in the private sector. *Springer*, pages 241–270, 2008.
- [328] Feng Xia, Jinzhong Wang, Xiangjie Kong, zhibo wang, Jianxin Li, and Chengfei Liu. Exploring human mobility patterns in urban scenarios: A trajectory data perspective. *IEEE Communications Magazine*, 56, 01 2018.
- [329] Xiaoyang Xie, Zhiqing Hong, Zhou Qin, Zhihan Fang, Yuan Tian, and Desheng Zhang. Transrisk: Mobility privacy risk prediction based on transferred knowledge. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 6(2), jul 2022.
- [330] Mengjia Xu, David Lopez Sanz, Pilar Garces, Fernando Maestu, Quanzheng Li, and Dimitrios Pantazis. A graph gaussian embedding method for predicting alzheimer’s disease progression with meg brain networks, 2020.
- [331] Erotokritos Xydias, Charalampos E. Marmaras, Liana M. Cipcigan, A. S. Hassan, and Nicholas Jenkins. Forecasting electric vehicle charging demand using support vector machines. *2013 48th International Universities’ Power Engineering Conference (UPEC)*, pages 1–6, 2013.

- [332] Zhixian Yan, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. Semantic trajectories: Mobility data computation and annotation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(3):49, 2013.
- [333] Hong Yang, Shirui Pan, Peng Zhang, Ling Chen, Defu Lian, and Chengqi Zhang. Binarized attributed network embedding. In Dacheng Tao and Bhavani Thuraisingham, editors, *2018 IEEE International Conference on Data Mining (ICDM 2018)*, pages 1476–1481, United States of America, 2018. IEEE, Institute of Electrical and Electronics Engineers. IEEE International Conference on Data Mining 2018, ICDM 2018 ; Conference date: 17-11-2018 Through 20-11-2018.
- [334] Shuang Yang and Bo Yang. Enhanced network embedding with text information. *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 326–331, 2018.
- [335] Zhiyu Yao, Yunbo Wang, Mingsheng Long, and Jianmin Wang. Unsupervised transfer learning for spatiotemporal predictive networks. *ArXiv*, abs/2009.11763, 2020.
- [336] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *NIPS*, 2014.
- [337] Usman Zafar, I. Safak Bayram, Sertac Bayhan, and Raka Jovanovic. Analysis of gps-based high resolution vehicle mobility data towards the electrification of transportation in qatar. July 2022. 48th Annual Conference of the IEEE Industrial Electronics Society, IECON2022 ; Conference date: 17-10-2022 Through 20-10-2022.
- [338] Richard S. Zemel, Ledell Yu Wu, Kevin Swersky, Toniann Pitassi, and Cynthia Dwork. Learning fair representations. In *ICML*, 2013.
- [339] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. SINE: scalable incomplete network embedding. *CoRR*, abs/1810.06768, 2018.
- [340] Yan Zhang, Lin Wang, Yi-Qing Zhang, and Xiang Li. Towards a temporal network analysis of interactive WiFi users. *EPL (Europhysics Letters)*, 98(6):68002, jun 2012.
- [341] Chen Zhao, An Zeng, and Chi Ho Yeung. Characteristics of human mobility patterns revealed by high-frequency cell-phone position data. *EPJ Data Science*, 10, 2019.
- [342] Fang Zhao, Ajinkya Ghorpade, Francisco Câmara Pereira, Christopher Zegras, and Moshe Ben-Akiva. Stop detection in smartphone-based travel surveys. *Transportation research procedia*, 11:218–226, 2015.



- [343] Vincent W Zheng et al. Collaborative location and activity recommendations with gps history data. In *WWW*, pages 1029–1038. ACM, 2010.
- [344] Yu Zheng, Lizhu Zhang, Zhengxin Ma, et al. Recommending friends and locations based on individual location history. *ACM Transactions on the Web (TWEB)*, 5(1):5, 2011.
- [345] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, PP:1–34, 07 2020.
- [346] George Kingsley Zipf. The p1 p2/d hypothesis: On the intercity movement of persons. *American Sociological Review*, 11(6):677–686, 1946.