eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

Department of Computer Architecture and Technology

Mobility Mining for Time-Dependent Urban Network Modelling

Thesis submitted in fulfilment of the requirements for the degree of *Doctor in Computer Science*

Harbil Arregui

Donostia, March 2021

Supervised by:
Dr. Olatz Arbelaitz Gallego - *Department of Computer Architecture and Technology EHU/UPV*
Dr. Oihana Otaegui Madurga - *Vicomtech, Basque Research and Technology Alliance (BRTA)*

– Etxekoei. Aitorri.

Eskerrak

Ez nioke lan honi itxiera eman nahi esker gutxi batzuk eman gabe.

Lehenik eta behin Olatz Arbelaitz eta Oihana Otaeguiri, egindako lana gidatu eta errazteagatik. Oihanari, bereziki, duela dagoeneko 11 urte lan-talde ikaragarri baten parte izatea ahalbidetzeagatik. Mila esker, urte hauetan beti emandako konfiantzagatik.

Vicomtech-eko lankideei, eta batez ere departamentukoei, egunerokotasuna xamurtzeagatik. Aipamen berezia merezi duzue azken urteetako Smart City-etako proiektuetan kide izan zaretenok. Lankide eta lankide-ohi guztietatik nere bigarren lagun-kuadrila osatzen duzuen guztioi, beti zaretelako umore-iturri eta bizitzaren ikuspuntu ezberdin askoren adibide. Esti, en especial, gracias por ser un apoyo y confidente inigualable desde el primer día.

Ama, Aita, Maite eta Isaak, hazi eta hezteko elkarte onena sortzeagatik. Familia eta Zarauzko lagunei, tesiarekin nola nindoan gehiegi ez galdetzeaz gain, hor egoteagatik, nire zati handi bat zarete.

Eta amaitzeko, Aitor, beti ondoan izan eta bide luze honek iraun duen bitartean gauza askori uko egiteagatik, lan hau zurea ere bada. Eskerrik asko zure mundua niri zabaltzeagatik eta bizitzak dakarzkigun erronka berriei aurrera egiteko eduki nezaken bidelagun onena izateagatik.

Harbil

Abstract

Mobility planning, monitoring and analysis in such a complex ecosystem as a city are very challenging. Advances in current sensing capabilities and computational intelligence are able to facilitate new seamless decision making solutions for city managers, transport operators and citizens themselves. However, an integrated overview of urban mobility as a whole is still a chimera. Our contributions in this dissertation are expected to be a small step forward towards a more integrated vision of mobility management.

The topics addressed in this dissertation link two main research areas: Machine Learning and Geographic Information Systems for Transportation. The main hypothesis behind this thesis is that the transportation offer and the mobility demand are greatly coupled, and thus, both need to be thoroughly and consistently represented in a digital manner so as to enable good quality data-driven advanced analysis. Data-driven analytics solutions rely on measurements. However, sensors do only provide a measure of movements that have already occurred (and associated magnitudes, such as vehicles per hour). For a movement to happen there are two main requirements: i) the demand (the need or interest) and ii) the offer (the feasibility and resources). In addition, for good measurement, the sensor needs to be located at an adequate location and be able to collect data at the right moment. All this information needs to be digitalised accordingly in order to apply advanced data analytic methods and take advantage of good digital transportation resource representation.

Our main contributions, focused on mobility data mining over urban transportation networks, can be summarised in three groups. The first group consists of a comprehensive description of a digital multimodal transport infrastructure representation from global and local perspectives. The second group is oriented towards matching diverse sensor data onto the transportation network representation, including a quantitative analysis of map-matching algorithms. The final group of contributions covers the prediction of short-term demand based on various measures of urban mobility.

Laburpena

Hiria bezalako ekosistema konplexuetan mugikortasunaren antolakuntza, jarraipena eta analisia egitea erronka handia da. Gaur egungo sentsoreetan eta konputazio adimentsuan eman diren aurrerapenei esker gai gara hiriaren kudeatzaile, operadore eta bizilagunei erabakiak hartzen lagun diezaieken soluzioak eskaintzeko. Dena dela, hiri barruko mugikortasunaren ikuspegi bateratu bat lortzetik urrun gaude oraindik. Lan honetako gure ekarpenek pauso txiki bat gehiago izan nahi dute mugikortasuna modu bateratuan kudeatzeko saiakera horretan.

Tesi honetan landutako gaiak ikasketa automatikoa eta garraiorako informazio-sistema geografikoen ikerketen artean kokatzen dira. Garraio-sarearen eskaintza eta mugikortasun eskaria oso lotuta daudela da gure hipotesi nagusia, eta beraz, biak adierazpide digital egoki eta trinkoen bidez azaldu behar direla kalitatezko datuen analisi aurreratuak egin nahi badira. Analisi aurreratu hauek egindako neurketetan dute oinarria. Baina sentsoreek neurtu ditzaketen desplazamenduak (eta loturiko magnitudeak), gertaturikoak besterik ezin dira izan. Desplazamendu bat gertatu ahal izateko bi baldintza nagusi betetzea ezinbestekoa da: i) desplazatzeko beharra edo nahia (eskaria) eta ii) desplazamendua ahalbidetuko duen azpiegituraren eskaintza edukitzea. Gainera, neurketa egoki batentzat, sentsore edo neurgailuak leku eta une egokian martxan egon beharko du. Informazio hau guztia digitalki jaso behar da metodo analitikoetatik ahalik eta ondorio zehatzenak atera ahal izateko eta garraio-sarearen eskaintzaren digitalizazioari probetxua atera ahal izateko.

Gure ekarpenak, hiriko garraio sareen gaineko mugikortasun datuen meatzaritzan kokaturikoak, batez ere hiru multzo handitan bana daitezke. Lehen multzoa modu-anitzeko garraioaren azpiegitura eta zerbitzu eskaintzaren adierazpide digital bati dagokio, ikuspegi global zein lokaletik aztertutakoa. Bigarrena, sentsore mota heterogeneoetatik lortutako datuak azpiegituraren adierazpidearekiko lotzearekin lotuta dago, map-matching algoritmoen gaineko analisi kuantitatibo batekin batera. Hirugarren multzoa, azkenik, epe-laburreko eskariaren aurreikuspenarekin lotuta dago, hiriko mugikortasunari dagozkion metrika ezberdinak banaka edo modu bateratuan aztertuz.

Resumen

La planificación, monitorización y el análisis de la movilidad en un ecosistema complejo como es la ciudad es un reto. Los avances en la capacidad de sensorización y la inteligencia computacional son capaces hoy en día de facilitar nuevas soluciones para la toma de decisiones para gestores/as, operadores de transporte y la ciudadanía. Sin embargo, la gestión integral de la movilidad como un todo es aún lejana. Nuestras contribuciones en esta tesis esperan poder ser un paso más hacia esa visión más integrada de la gestión de la movilidad.

Los temas principales que se abordan en esta tesis se encuentran entre las áreas de investigación del aprendizaje automático y los sistemas de información geográfica para el transporte. Nuestra hipótesis principal es que la oferta y la demanda de la movilidad están fuertemente acoplados, y por tanto, ambos deben ser representados digitalmente de una manera consistente para un análisis de datos avanzado de calidad. Los análisis basados en datos se basan en mediciones. Pero, los sensores que realizan estas mediciones de desplazamientos (y magnitudes asociadas) sólo pueden medir desplazamientos, en efecto, realizados. Para que ese desplazamiento se dé, es imprescindible que se cumplan dos condiciones: i) la existencia de la demanda y ii) la existencia de la infraestructura de transporte que la habilite. Además, para que la observación de la medición sea adecuada, es importante que el sensor esté localizado en un lugar e instante apropiados. Toda esta información deberá representarse de forma acorde para poder aplicar métodos avanzados de analítica que a su vez puedan sacar provecho de la representación digital de la red de transporte ofertada.

Nuestras principales contribuciones, enfocadas en la minería de datos de movilidad sobre redes de transporte urbanas, se pueden resumir en tres grupos. El primero consiste en una descripción de la representación digital de los recursos de transporte multimodales desde dos puntos de vista. El segundo grupo, está orientado a la asociación de los datos provenientes de sensores heterogéneos a dicha representación de la red de transporte, incluyendo un análisis cuantitativo de algoritmos de map-matching. El tercer grupo, para finalizar, aborda las contribuciones relativas a la estimación de la demanda de

movilidad a corto plazo sobre varias métricas que representan la movilidad urbana.

Contents

List of Tables

List of Figures

Introduction 1

1.1 Background and motivation

Over the last few years, the Intelligent Transport Systems (ITS) sector has met several new paradigms, which have lead to novel challenges. The most important paradigms are digitalisation, which enables digital twinning, and the almost full connectivity between remote locations at any given moment, which enables new Internet of Things (IoT) architectures. Now, almost every citizen or vehicle movement can be digitally recorded instantaneously. Combining this data availability with advances in Computational Intelligence, another significant paradigm, has opened new opportunities to tackle ITS problems from different perspectives. This study will focus on passenger transport and mobility at the city level. The concept of what a *city* is, as well as its spatial and socioeconomic delimitation, can be quite abstract, but we rely on a harmonised definition given by the Organisation for Economic Cooperation and Development (OECD) and the European Comission (EC) [1]:

*"This definition of **city** works in four basic steps and is based on the presence of an 'urban centre', a new spatial concept based on high-density population grid cells.*

- *Step 1: All grid cells with a density of more than 1 500 inhabitants per sq km are selected.*
- *Step 2: The contiguous high-density cells are then clustered, gaps are filled and only the clusters with a minimum population of 50 000 inhabitants are kept as an 'urban centre'.*
- *Step 3: All the municipalities (local administrative units level 2 or LAU2) with at least half their population inside the urban centre are selected as candidates to become part of the city.*
- *Step 4:The city is defined ensuring that 1) there is a link to the political level, 2) that at least 50 % of city the population lives in an urban centre and 3) that at least 75 % of the population of the urban centre lives in a city."*

Short-, mid- and long-term mobility decision making in future cities will rely on more and more efficient digital representations of the heterogeneous transportation resources and intelligent data management towards a digital twinning of the city. We expect that this study may provide some useful contributions towards this scenario.

This dissertation is a consequence of multiple years of research at the Intelligent Transport Systems and Engineering Department at Vicomtech (Donostia, Spain). Participation in a wide range of research projects focused on the mobility of people

and goods has brought us face to face with issues regarding how transportation resources and mobility demand are digitally represented.

Diverse challenges and problems have arisen during this period, ranging from the time-varying characteristics of transport networks when computing route optimizations, to the uncertainty of limited or low quality data. They have always appeared together with complex interoperability issues given different indexing techniques, data models and naming conventions when handling geospatial data. In parallel, the ever increasing amount of measurable data requires efficient storage and processing methods to manage the digital representation of transportation resources and movement data.

Now, technological capabilities such as distributed computing systems have become key enablers for pushing large scale processings in highly sensed environments. Big Data approaches are widely presented as current cutting-edge solutions to obtain conclusions and learning from what is being sensed. In this scenario, cities, factories and the automotive sector are big players with strong economic investments for innovation in these lines of research. Scientific research efforts over the last few years show prolific literature on data mining and machine learning.

The motivation of the present study, however, is to enhance the current literature dealing with network based spatio-temporal data mining and its real world applications. In the following pages, we would like to present a deep understanding of what a network-based structure implies when tackling data analysis under spatially and temporally correlated facts.

Therefore, our study is **Mobility Mining for Time-Dependent Urban Network Modeling**. We focus on *Network Modelling*, because we try to work towards a representational model to be able to store, process and query network constrained instead of open space movements. We also focus on the *Urban* context, because we set the research in a complex ecosystem, the city. Moreover, we focus on *Time-Dependent* modelling, since this city ecosystem is alive, thus, variable at different time scales. Finally, we consider the concept of *Mobility Mining*, because our last objective is to be able to apply coherent data mining and machine learning techniques over our representational model.

At this stage, we would like to state that throughout the study we will use the concept of movement repeatedly. With this concept we will refer to a wide range of observable actions derived from a journey or trip. For instance, among others, we consider as diverse examples as: a vehicle or pedestrian journey from point A to point B; the observation of a journey as the vehicle or pedestrian passes through a sensorised location (even if the origin and destination are unknown); or the beginning of a motorised journey where a parked vehicle leaves its parking space.

As previously mentioned, the dissertation resides within a particular application case: the urban mobility. Understanding how citizens move making use of the

transportation resources available, both private and public, is of great interest for city planners, transportation operators, users themselves and new stakeholders that begin to appear in parallel with novel transportation business models such as peer-to-peer schemes, Mobility as a Service (MaaS) or even autonomous driving. Nevertheless, many of the methodological and implementational contributions are expected to be applicable in other scenarios that share similar constraints .

1.2 Key contributions

The objective of this thesis, is to present the interrelation between the transportation resources and the movement that happens inside a city. Data-driven analytics solutions rely on measurements obtained from multiple and heterogeneous sensors. However, sensors are only able to record movements that have happened, and for a movement to happen, apart from the demand, the offer needs to exist. In addition, the sensor needs to be located at an adequate location and be able to collect data at the right moment. All this information needs to be digitalised accordingly in order to apply advanced data analysis methods and take advantage of good digital transportation resource representation.

Below we state the main outcomes that our study has sought to contribute to the science of mobility data mining over urban transportation networks, and we refer to the published articles when applicable.

- A description of what type of information is available for city mobility decision makers both regarding digital transportation infrastructure representation and movement data itself, from the experience extracted from various research projects. In these research projects, interaction with several city managers has helped to understand some main challenges and difficulties they face in any attempt towards integrated urban mobility analysis. Chapter 3.
- A proposal of the concept named Urban Movement Space, a combination of multiple types of information layers that are capable of considering the main heterogeneities found when representing resources offered (infrastructure assets) for mobility purposes [2]. Chapter 3.
- A generalised classification of types of mobility-related sensor data. Chapter 4.
- A quantitative analysis of the digital representation of the transportation network with regard to the accuracy of map-matching algorithms that locate moving positions onto a digital map [3]. Chapter 4.
- A traffic density representation model for localised areas such as roundabouts [4]. Chapter 5.
- A Random Forest-based short-term prediction model evaluation with novel positioning data sources and use-case applications [4]. Chapter 5.

- A city-wide study of interaction between multiple types of data sources, combining traffic counters and parking occupancies, for short-term prediction of parking occupancy with real data. Chapter 5.

1.3 Thesis outline

This thesis is structured in the following six chapters, see the graphical representation of the structure in Figure 1.1:

1. Introduction: We summarise the motivation and main contributions of the dissertation.
2. Background and Related work: The main foundations and a literature review of related scientific works are collected in order to better understand the topics related to this dissertation.
3. Representing the urban movement space: Our proposal to represent the transportation network resources deployed in cities.
4. Mapping movement data over the transportation resources: Contributions on what kind of mobility data is collected and how their representation can be related to the digital representation proposed in the previous chapter.
5. Estimating short-term mobility demand and applications: Exploitation of mobility data for advanced analytics and contributions on two scenario cases for short-term demand prediction.
6. Conclusion: Final summary and revision of the outcomes obtained from the dissertation, as well as identification of discussion and future work topics.

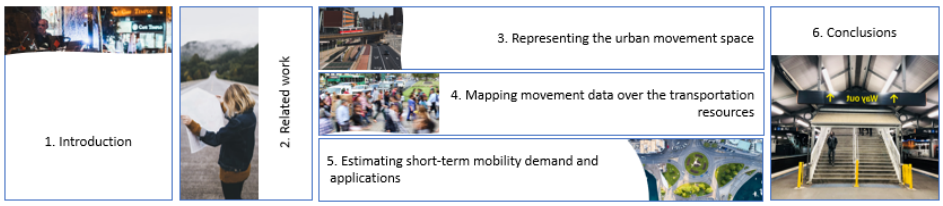


Figure 1.1: Chapter structure of the document

Background and Related work 2

When considering the need, the feasibility and the opportunity of the design and development of a time-dependent urban network model, allowing efficient mobility data mining applications, the concepts we need to understand are diverse. Due to the complexity and heterogeneity of the different concepts introduced, we will combine the description of the main background of the relevant topics together with the study of the related work published in the literature. This compilation has been divided into three main areas. This chapter aims to collect and classify the relevant literature related to these three main aspects, which are developed in three subsequent sections.

- How mobility infrastructure and resources are, or can be, digitally modelled (Section 2.1 Digital Transport Network Modelling)
- The collection and representation of spatial, temporal and spatio-temporal data, given that movement data is a particular case, as well as data mining models (Section 2.2 Spatio-Temporal Data Modelling and Mining)
- Different approaches for knowledge extraction based on movement data, especially in cities (section 2.3 Urban Scale Data Driven Mobility Analysis)

At the end of the chapter we have included a short list of some promising tools for related research, in Section 2.4, and a summary relating the concepts presented throughout the chapter to how they have motivated us to carry out our study, in Section 2.5.

2.1 Digital Transport Network Modelling

Transport networks are one of the application domains of digital maps. In this section, we will introduce 1) some generic features of digital maps, 2) describe some models used to digitally represent the transportation infrastructure and locations relative to that infrastructure, 3) list some standardisation activities used in these kinds of maps and 4) explain some technologies used for storage and indexing of digital spatial data.

This section will serve us as an introduction to the main existing technological and semantical ways to represent and encode the information that we will later need, from a lower level information to a higher level.

2.1.1 Digital maps

Digital maps are a virtual representation of real world elements, describing their locations and spatial relations among them. To accomplish the objectives outlined in the present dissertation, we must create, process and read digital maps that describe city resources and infrastructures that host the movements in the urban ecosystem. Therefore, we need to understand how urban transport networks are usually digitally represented, and what options we have in order to represent any new data related to them.

In general, maps can be digitalised in raster or vector format (see Figure 2.1). In most cases map sources with road information are distributed in vector format. Vectors can represent point, line, and area features very accurately, and they are more efficient than raster data in terms of storage. Raster maps can be understood as images, where each pixel represents a value. They are easier to understand and analytical operations (interpolation, filtering) are also easier to perform. One of the drawbacks of raster formats, however, is that each cell can only represent one feature. Vector data requires less space to store than raster data, but often requires more processing in order to be used in a GIS application. In the rest of the dissertation, unless specified, we will use vector maps.

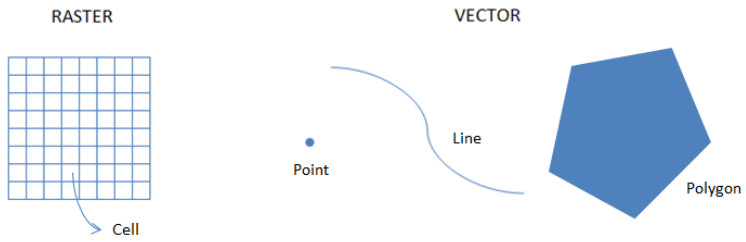


Figure 2.1: Raster vs. vector representation

Points are defined by a pair of latitude, longitude coordinates. Lines (also called arcs) are constructed by several consecutive pairs of coordinates, as well as polygons, but the latter represent closed areas. Elements in vector maps are sometimes processed to obtain the transportation network as a graph, resulting in a representation of nodes connected by directed or undirected links.

2.1.1.1 Digital map models

The explosion of digital maps for road navigation purposes was extended with solutions such as TeleAtlas Multinet [5], which included information on buildings, administrative areas and Points of Interest (POI). The newest mapping techniques are

able to generate accurate and highly detailed maps using satellite imagery sources, sensors, 360 view cameras mounted on cars, or through crowdsourcing. Spatial vector representation is enhanced by a more complete data model that includes the most relevant entities and their attributes and relations. For instance, for digital road maps, the accuracy and high level of detail such as lane information elevation and curvatures is vital when tackling new challenges that are brought on by, for example, autonomous vehicles and on-board systems [6]. They need high-precision, up-to-date and accurate maps, also known as high-definition maps. In this sense, format specifications such as the Navigation Data Standard (NDS) Open Lane Model* [7] and Association for Standardization of Automation and Measuring Systems (ASAM)† OpenDrive [8] must be mentioned. In addition, in Car-to-Cloud or V2X environments, there is a need to disseminate maps that are being sensed in real time. For this purpose, for example, HERE published an interface specification that defined how sensor data gathered by vehicles on the road could be ingested by a cloud infrastructure‡. This specification included connectivity models for simple and complex intersections and, in addition to lane center line geometries, lane boundary geometries and types of road markings, guardrails and walls were given.

The main exponent of crowdsourced digital maps is OpenStreetMap (OSM) [9]. The basic original data types are nodes, ways and relations and they are described by tags. A tag is a key/value pair and describes specific features of a map element under agreed conventions about their meaning and use. A way is an ordered list of nodes which normally also has at least one tag or is included within a relation. For example, *highway = residential* is a tag used on a way to indicate a road along which people live. Other tags may indicate access restrictions such as paths for pedestrian use or bike lanes.

2.1.2 Referencing location to transportation infrastructure

Locations over a transportation infrastructure can be modelled in several ways. The heterogeneity of map providers, each with a different model and specification, makes it difficult to match the same element onto two different maps.

Depending on the model selected, locations must then be referenced accordingly. The work of Jensen et al. [10] compiles and builds such relationship in the following ways:

- Kilometre-post: Distance markers on the road are used (kilometre posts). Locations are expressed in terms of the road, the distance marker code and the offset from the distance marker. The network topology is not considered.

* https://nds-association.org/
† https://www.asam.net/
‡ http://360.here.com/2016/06/28/here-standard-for-shared-car-data-wins-pan-european-backing/

- Link-node representation: Based on weighted directed or undirected graphs. A node is a road location with a significant change of traffic properties, e.g., an intersection. Geographical details are omitted and only the graph topology is used.
- Geographical representation: Uses coordinate-based representations to directly reference locations on the surface of earth rather than measure distances along roads from certain locations.
- Segment representation: Models infrastructure as a collection of segments that intersect at connections (locations where there is an exchange of traffic). This representation preserves the network topology and captures a complete set of roadways. This representation can act as an integrator of the previous representations. Usually, it is an internal model, and segment identifiers are not known for external applications or users.

The concept of linear referencing (LRS [11]), instead of referencing location features on the surface of earth, is widely used in the literature of Geographic Information Systems in Transportation (also known as GIS-T) [12]. Figure 2.2 describes this Linear Referencing concept.

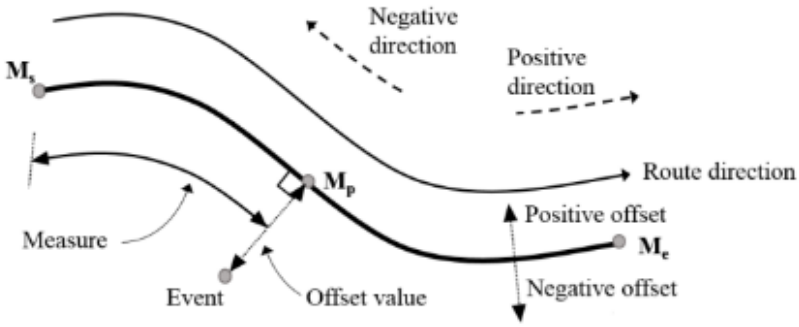


Figure 2.2: Linear referencing [13]

The location referencing in multimodal scenarios is limited in the literature, but the NCHRP Report on Guidelines for the Implementation of Multimodal Transportation Location Referencing Systems is a good reference [14].

2.1.3 Standardisation activities

In this subsection, we collate multiple standardisation activities promoted by different entities and associations in order to find common ground with respect to the representation of maps and how locations are referenced. As we will see, there is a long heterogeneous list of efforts and activites and the selection of the most suitable one according to the application is not always straightforward.

As previously mentioned, the use of different location referencing systems creates problems, especially when trying to share or reuse information. In general, different maps (from different providers) may use different location references. This fact makes the interoperability between maps difficult. We can name three well-known location referencing solutions:

1. TMC [15]: Traffic Message Channel uses tables of pre-coded locations. It is used as a digital information channel to broadcast coded traffic information via a radio signal
2. AGORA-C [16]: A proprietary dynamic location reference method, standardised through ISO to match locations between dissimilar maps effectively
3. OpenLR[§]: Open Compact and Royalty-free Dynamic Location Referencing. It includes an open-source Java implementation. Less complex than AGORA-C, it supports several reference types: Line, Geo-Coordinate, PointAlongLine, PoiWithAccessPoint, Circle, Rectangle, Grid, Polygon and ClosedLine [17].

The main drawback of TMC is that location codes are fixed, thus limiting the size of a code list that includes all countries (limited to 64k locations). Therefore, they only cover major roads and are not useful for other roads and streets. Both AGORA-C and OpenLR rely on dynamic codes, created when needed in the original map system, transmitted in the message, and then discarded after decoding at reception. OpenRL requires a lower bandwidth than AGORA-C, and some authors in the literature have proposed approaches in combination with OpenLR to match linear or circular routes between dissimilar maps based on geometric dissimilarities [18].

In Europe, some efforts have been made over the past few years to establish a common basis for representing and sharing information.

For instance, geospatial and digital maps-related progress is under the frame of the INSPIRE Directive, aimed to create a European Union spatial data infrastructure, whose full implementation by Member States is required by 2021[¶]. The INSPIRE Thematic Working Group Transport Networks (TWG-TN) is a multinational team of experts in the field. Their proposed model [19] includes some features such as:

- Network connection mechanisms for cross-border (between countries) and intermodal (between networks of different transport modes) connectivity.
- Object referencing to support the reuse of information
- Linear referencing
- Mechanisms to combine network elements into high-level semantic meanings

Figure 2.3 shows an overview of the most relevant objects described by this model.

Additionally, CEN/TC 278 - Intelligent transport systems is the technical body for the European Committee for Standardisation in charge of standardisation in the field

[§] http://www.openlr.org/index.html
[¶] https://inspire.ec.europa.eu/about-inspire/563

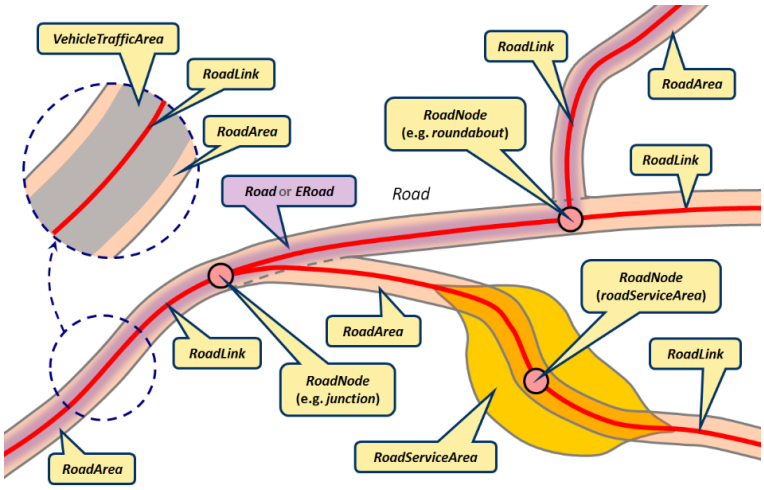


Figure 2.3: Overview of the main Road Transport Networks objects, INSPIRE [19]

of telematics to be applied to road traffic and transport, including those elements that need technical harmonisation for intermodal operation in the case of other means of transport ‖. It is comprised of 11 Working Groups, depicted in Figure 2.4.

The most relevant topics for the current dissertation that are addressed by these working groups are: WG3: Public Transport, WG7: ITS spatial data, focused on geographic road data and WG 17: Mobility integration. TN-ITS**, for instance, is a platform involved in the works of CEN WG7 regarding standardisation of static geographic road network data (ITS spatial data, CEN/TS 17268). Some European Commission funded projects, such as ROSATTE (ROad Safety ATTributes exchange infrastructure in Europe)††, generated significant input to the work of this platform.

Currently, several standards from the multiple working groups have already been published. Some of the most relevant are collated in Table 2.1.

In addition to the standardisation efforts in the European Union, we also find other approaches in the United States. In fact, the ITS Standards Program, established by the U.S. Department of Transportation (DOT), collates the current 92 standards in order to widen the use of ITS technologies surface transportation systems‡‡.

‖ https://standards.cen.eu/dyn/www/f?p=204:7:0::::FSP_ORG_ID:6259&cs=1EA16FFFE1883E02CD366E9E7EADFA6F7

** https://www.tn-its.eu/

†† https://cordis.europa.eu/project/id/213467

‡‡ https://www.standards.its.dot.gov/DevelopmentActivities/PublishedStandards

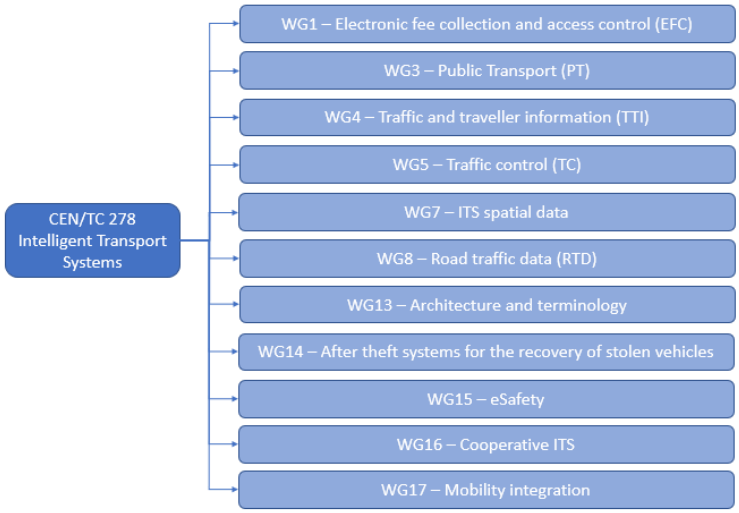


Figure 2.4: Working groups of CEN/TC 278

WG	Reference	Title	Publication date
3	EN 12896-2:2016	PT - Reference data model (TRANSMODEL). Part 2: Public transport network	2016-09-28
3	CEN/TS 16614-1:2020	PT - Network and Timetable Exchange (NeTEx) - Part 1: Public transport network topology exchange format	2020-04-29
7	CEN/TS 17268:2018	ITS - ITS spatial data - Data exchange on changes in road attributes	2018-12-12
17	CEN/TR 17143:2017	ITS - Standards and actions necessary to enable urban infrastructure coordination to support Urban-ITS	2017-10-04
17	CEN/TR 17297-1:2019	ITS - Location referencing harmonization for Urban ITS - Part 1: State of the art and guidelines	2019-05-29
17	CEN/TS 17297-2:2019	ITS - Location Referencing Harmonisation for Urban-ITS - Part 2: Transformation methods	2019-09-25
17	CEN/TR 17401:2020	ITS - Urban-ITS - Mixed vendor environment guide	2020-01-22
17	CEN/TS 17241:2019	ITS - Traffic management systems - Status, fault and quality requirements	2019-04-03

Table 2.1: Some of the most relevant published Intelligent transport systems (ITS) and Public transport (PT) standards by CEN/TC 278

The Open Geospatial Consortium (OGC), as well, covers transportation related objects inside its CityGML[§§] specification.

Despite the heterogeneous list of existing standards and specifications (those listed do not cover all of them), their applicability is not always direct. Some authors have solved their needs by selecting portions of one specification to solve some requirements and portions from other specifications to solve other requirements [20]. In addition, the existence of these specifications is not always reflected in an extended use by the industry and by local city administrations. For instance GTFS is a *de facto* standard in many cities to the detriment of TRANSMODEL (EN 12896). Therefore, in the work that has lead to this dissertation, we have been inspired and tried to follow the main basis established by these data models to some extent, whereas the idea of following a unique standard has been found to be unrealistic.

2.1.4 Storage and indexing

Digital representations of spatial data, generally speaking, and more specifically both transport networks and movement data collected by any sensing technique, need to be stored in efficient electronic formats if we hope to use them to apply data mining techniques. In fact, how they are stored and indexed will impact the read and write access times, as well as the execution of spatial or temporal joins of any kind.

Technological advances in spatial data storage allow great geospatial analysis capabilities through the development of tools to make data storage and processing scalable. In general, GIS data is known to be stored and processed in three fundamentally different ways [21]:

- Flat file: data is indexed for spatial searches but stored on a single filesystem and accessed using a simple API. For example: *Vector Cluster*. Vector Cluster can not edit or delete records, it is essentially a read-only format. A Vector Cluster can contain one or many feature layers, which can be found quickly by way of a B+ Tree index on the layers. Each layer has its own header and is independently indexed. The spatial index is a 2-dimensional R* Tree. Accordingly, nodes in the tree are ordered by the Minimum Bounding Rectangle (MBR) method. The Vector Cluster can support attributes with a schema, or use a schema-less key/value pair system.
- Relational database (RDBMS): general purpose databases are powerful tools for managing many types of data. Spatial extensions are added to handle particularities of spatial data. The main example is *PostGIS*[¶¶], a PostgreSQL database extension.

[§§] https://www.ogc.org/standards/citygml
[¶¶] https://postgis.net/

- Distributed-cloud key-value stores: built atop many levels of software platforms and a possibly large amount of computing hardware. This is the most complex paradigm, devoted to large-scale storage and processing, where the best known example is *GeoMesa*[22]. GeoMesa is an open-source, distributed, spatio-temporal database built on a number of distributed cloud data storage systems, including Accumulo, HBase, Cassandra, and Kafka. GeoMesa extends Accumulo (a Hadoop-based datastore) with spatio-temporal indexing and querying, based on OGC SimpleFeature data model. Accumulo is a distributed key-value datastore based on the BigTable [23] design from Google. The underlying use of HDFS, the Hadoop Filesystem, provides fault tolerance, parallel access and load balancing. As a NoSQL key-value store, a single index is built in Accumulo, on the lexicographically-ordered records. Indexing spatiotemporal data is a matter of finding a sensible way to flatten three dimensions of data (longitude, latitude, and time) into a single dimension: the list of Accumulo keys. The specific flattening is described by an index-schema format, a customizable space-filling curve that interleaves portions of the location's Geohash with portions of the datetime string. Many other approaches, similar to GeoMesa, can also be found for spatial data indexing over Hadoop. *Geospark* [24], for example, is an in-memory cluster computing framework for processing large-scale spatial data. At the same time, *GeoWave* [25] is an open source set of software intended to be a multidimensional indexing layer that can be added on top of any sorted key-value store. It adds multi-dimensional indexing capability to Apache Accumulo and Apache HBase; it adds support for geographic objects and geospatial operators to Apache Accumulo and Apache HBase; and it provides Map-Reduce input and output formats for distributed processing and analysis of geospatial data.

Other intermediate aproaches are also found, for example *GeoPackage*[***], cross-platform and compatible with many types of GIS software. This file format (defined by OGC in 2014 and becoming a powerful alternative to the propietary *ShapeFile*) is a single file, as well as the Vector Cluster, and it also uses an R Tree as a spatial index. Nevertheless, Geopackage considers itself a RDBMS.

Several studies have addressed the specific computational challenges of storing and querying records that represent locations, movements and trajectories[26]. Another relevant approach proposed in the literature is the Moving Objects Database (MOD) [27]. Their approach suggests storing a dynamic attribute such as vehicle speed, instead of its location, since it is expected to change less frequently. Therefore the updating overhead in the database is reduced. The objective of another study was to provide a comprehensive data model and query language for moving objects in networks[12], supporting the description and querying of complete histories of movement (past and future). Camossi et al. [28] address issues related to representation and storage of multi-granularity data in spatial and temporal models such as

[***] https://www.geopackage.org/

the need to convert data to common finer or coarser granularities in order to enable comparisons. A very recently published survey has collected research on spatial, temporal and spatio-temporal databases [29], specifying query models, indexing and ontologies.

At the same time, considering most transportation modes are, somehow, network constrained, the representation and indexing following graph paradigms must not be neglected. There are several ways in which a graph can be stored in a file (CSV, XML or JSON formats): GraphML and GraphSON are two of them.

For high performance, large-scale graph processing solutions are a valuable alternative, for instance:

- Gelly for Apache Flink [†††]: Gelly is a Graph API for Flink, an open source platform for distributed stream and batch data processing. It contains a set of methods and utilities which aim to simplify the development of graph analysis applications in Flink.
- GraphX for Spark [‡‡‡]: GraphX is Apache Spark's API for graphs and graph-parallel computation. In addition to a highly flexible API, GraphX comes with a variety of graph algorithms, many of which were contributed by the community.
- GraphFrames for Spark [§§§]: GraphFrames support general graph processing, similar to Apache Spark's GraphX library. However, GraphFrames are built on top of Spark DataFrames, resulting in some key advantages: First, uniform APIs for all 3 languages: Python, Java and Scala. Second, queries can be phrased in the familiar powerful APIs of Spark SQL and DataFrames. Finally, support for DataFrame data sources, allowing writing and reading graphs using many formats like Parquet, JSON, and CSV.

After introducing the main background on digital maps, location referencing, known standards and solutions for storage and indexing, we are ready to continue with the next steps, focused on the extraction of knowledge and advanced analysis of the represented data, characterised by a strong spatio-temporal nature.

2.2 Spatio-Temporal Data Modelling and Mining

The next important topic we need to describe is relative to the spatio-temporal data modelling and data mining. Even though we approach it from the transportation domain, this topic involves significant scientific challenges across multiple domains where time-varying geographic information is used. After the previous section, which

[†††] https://ci.apache.org/projects/flink/flink-docs-stable/dev/libs/gelly/
[‡‡‡] http://spark.apache.org/graphx/
[§§§] https://databricks.com/blog/2016/03/03/introducing-graphframes.html, http://graphframes.github.io

was more related to the representation of spatial information, this section covers the main concerns regarding knowledge extraction from spatial and spatio-temporal data. We can conceptualise this as a next step forward.

Spatio-temporal data mining (STDM) is a challenging lines of research. It is aimed at the extraction of implicit knowledge, structures, relationships, or patterns from data, in the same way as classical data mining.. However, dependencies in spatial and temporal data violate the stationarity assumption of classic statistical models [30], and need specialised modelling and forecasting techniques. There are important differences in the nature of spatial and temporal data. Camossi et al. [28] stated that, while the operations connected with the temporal domain rely strictly on its monotonic order, spatial operations mainly depend on topological relations. Temporal data also has characteristics such as periodicity, not common in spatial data.

2.2.1 Spatial Data modelling and mining

When we introduced digital maps, we already described the main basic data types used to model spatial information. The main purpose when modelling spatial data is to describe a representation of the real world where entities of given size, position or shape are located at different places, cover different extents or have a different type of relationships between them (for example, two spatial entities can be described as being near). In general, literature has coped with three spatial data models: an object model, which is generaly represented by vector data models; a field model, represented by raster data models; and spatial network models, represented by abstract, geometry or network graphs [31].

Once data is represented by using a suitable model, multiple types of analysis of varying levels of complexity can be tackled for advanced knowledge extraction, through spatial data mining techniques. Research regarding spatial data mining is very extensive and has been approached from perspectives such as geocomputation, geovisualisation and spatial statistics, among others.

Multiple data mining tasks can be found in the literature. A well-known classification [32] collects the following tasks:

- Spatial classification and prediction
- Spatial association rule mining
- Spatial clustering, regionalisation and point pattern analysis
- Geovisualisation

There is another relevant concept that we consider worth introducing here: spatial interaction. Spatial interaction models describe how different locations are functionally interdependent [33]. The main classical aproach for modelling this is gravity models. In gravity models, the degree of spatial interaction (that is, the strength of a

flow, such as migration, trade flow, travel, etc.) between two places is proportional to the sizes of the population in both places, and it is inversely proportional to the distance between them. More novel spatial interaction model examples are the radiation model, the rank-distance model, the single-parameter model and the population-weighted opportunity model [34].

The spatial interaction described by the introduced models can also be numerically quantified. Here, we aim to mention Moran's I metric, in Eq. 2.1, which is a very well-known statistical metric used to quantify the spatial autocorrelation.

$$I = \frac{N}{\sum_i \sum_j w_{ij}} \frac{\sum_i \sum_j w_{ij} \left(X_i - \overline{X}\right) \left(X_j - \overline{X}\right)}{\sum i \left(X_i - \overline{X}\right)^2} \tag{2.1}$$

where N is the number of spatial units, X_i and X_j are, respectively, the measured value of the variable of interest in each location i and j, $\overline{X}$ is the average value for all X and w_{ij} represents the spatial weights matrix.

The reasoning behind the Moran's I metric can be explained as follows: values near to 0 are obtained when there is a random relation, as far as they deviate from 0, positive values are obtained when positive spatial autocorrelation exists, and negative values are obtained with negative autocorrelation.

2.2.2 Time Series modelling and mining

After a short introduction to the spatial domain, we now introduce the traditional data analysis in the time domain.

A time series is a series of data points indexed in time order. In general, data is taken at successive, equally spaced points in time. Time Series Analysis provides a robust statistical framework for assessing the behaviour of time series, in order to, for example, predict future behaviour.

One of the most important features in time series is the serial correlation. It represents how sequential observations in a time series affect each other. In general, serial correlation can be studied after removing deterministic trends as well as seasonal variations. It is relatively straightforward to identify deterministic trends as well as seasonal variation and decompose a series into these components. However, once such a time series has been decomposed we are left with a random component.

In time series statistics, the concept known as stationarity is frequently used. Stationarity means that the time series properties do not depend on the time of observation. A time series that is stationary in the mean and stationary in the variance is known as second order stationarity. In second order stationary time series the correlation

between sequential observations is only a function of the number of time steps separating each sequential observation[¶¶¶]. This number of steps is also known as the lag.

A time series model is a mathematical model that attempts to explain the serial correlation present in a time series. The time series analysis process considers a wide variety of models and chooses the simplest one that can explain the serial correlation of data under analysis. To do so, when fitting an appropriate model, the objective is to reduce the serial correlation in the residuals of the model and its time series. A useful utility is the correlogram representation. The fit needs to be refined until no correlation is present in these residuals. After that, forecasts about future values are made, using the model and its second-order properties. The quality of the forecasts is assessed using statistical methods (confusion matrices, ROC curves for classification, regressive metrics like MSE, MAPE). The process is iterative until optimal accuracy is obtained.

Some basic time series models are Random Walks and Discrete White Noise.

2.2.3 Spatio-temporal phenomena and movement data

After understanding the spatial and time domains, the spatiotemporal phenomena can be incrementally presented. In this case, three main classifications of data types are given by Shekkar et al. [35]: temporal snapshot models, which can be represented as time series of locations; temporal change models, where a snapshot at a certain time is given together with incremental changes; and event/process models. Authors propose developing statistics and techniques to incorporate spatial and temporal information into the data mining process, instead of materialising the spatial and temporal relationships into traditional data input columns.

The fact is that the fields of time series analysis and spatial analysis have largely developed separately from one another [30]. The nature of space and time domains are different. While time has a clear ordering of past, present and future, space does not. Moreover, temporal data has characteristics like periodicity (seasonalities), which is very rare in spatial data.

The complexity of data collected from movements, as a particular case of spatio-temporal data, is high: objects can be represented by geometry, a position at a given time and other (non-spatiotemporal) attributes. Higher levels of reasoning are needed when dealing with these kinds of objects than with traditional data.

Space–time prisms models are worth mentioning, even though they do not model actual movements. They model the potential movement based on a known speed limit on the object's movement, which limits the part of space–time where a moving

[¶¶¶] https://www.quantstart.com/articles/Serial-Correlation-in-Time-Series-Analysis

object possibly could have been between two measured space–time locations [36]. They somehow limit the space-time boundaries of feasible movements.

2.2.4 Spatio-temporal modelling and forecasting

In the previous section, we mentioned some studies that address forecasting applications. The most classical approaches are parametric.

Some of the most well known time-series modelling families are ARIMA models and their variants. ARIMA models aim to describe the autocorrelations in the data. The name ARIMA is an acronym for Auto-Regressive Integrate Moving Average.

In multiple regression models, the variable of interest is forecasted by the use of a linear combination of predictors. Similarly, in an autoregression model, $AR(p)$, the variable of interest is forecasted using a linear combination of past values of the variable. The term autoregression indicates that it is a regression of the variable against itself.

The term MA comes from Moving Average. Rather than use past values of the forecast variable in a regression, a moving average, $MA(q)$, model uses past forecast errors in a regression-like model.

The combination of autoregression and moving average results in a non-seasonal ARIMA (AR-I-MA, I of integration) model.

Seasonality of the data can also be modelled. These models are called SARIMA, from Seasonal ARIMA. A seasonal ARIMA model is formed by including additional seasonal terms in the ARIMA models. It is written as shown in Figure 2.5 (where m represents the number of periods per season):

$$\text{ARIMA}\ \underbrace{(p,d,q)}_{\substack{\uparrow\\ \text{Non-seasonal part}\\ \text{of the model}}}\ \underbrace{(P,D,Q)_m}_{\substack{\uparrow\\ \text{Seasonal part}\\ \text{of the model}}}$$

Figure 2.5: SARIMA model

Moreover, we can find a new variant, where Space and Time are accounted for simultaneously: Space-Time ARIMA or STARIMA, which can also be seasonal or not.

Kamarianakis and Prastacos [37] discussed the application of space-time autoregressive integrated moving average (STARIMA) methodology to represent traffic flow patterns. In this paper, traffic flow data is represented in the form of a spatial time series, collected at specific locations at constant time intervals. Furthermore, the

model can be used to assess the impact of traffic-flow changes on other parts of the network through the use of weight matrices estimated on the basis of the distances among the various locations [38].

Another study, published by Lin et al. [39], demonstrated the application of the STARIMA model in an urban network considering the space and time sites, which can greatly improve the accuracy of short-term traffic flow prediction. Non-linear functions could be estimated by this model and accurate forecasts in traffic flow network simulation studies were given. The authors suggested an additional application to the prediction of traffic flow: complementing missing values.

Cheng et al. [30] state that STARIMA model families have not yet been adequately adapted to deal with spatial heterogeneity and parameter estimates are global. This implies that the space-time process needs to be stationary (or made stationary through differencing/transformation) for STARIMA modelling to be effective.

2.2.5 Spatio-temporal analysis in network space

Research in spatio-temporal statistic in the case of network space (STN), is very limited in the literature. The unique challenges of the network space include directionality and anisotropy of spatial dependency, connectivity and high computational cost [35]. Anisotropy means that weights depend on the geographical direction as well, not only on the geographical distance. Qi et al. [40] propose a Spatial Network Algebra but in this case, time dimension is not accounted for. Zhang and Wang [41] use network distances to forecast average weekday ridership in the New York City subway system. They calculate the before and after situation of the launch of new services according to spatial dependencies, applying Kriging statistics (classically used for spatial interpolation in euclidean space). Gunturi and Shekhar [42] apply STN concepts to consider the effect of arterial signalling; the analysis is, however, only made for road traffic vehicles.

We have now described multiple perspectives found in the literature to tackle the spatio-temporal data modelling and mining. More specifically, our study will focus on the spatio-temporal representation and understanding of mobility in urban scenarios and predictive capabilities, which can be seen as a particular use case. Therefore, in the next section, we will go deeper into the background and literature review of mobility analysis in urban settings, from a data-driven perspective, with a specific emphasis on short-term mobility prediction.

2.3 Urban-scale Data-driven Mobility Analysis

The applications of the analysis of urban mobility are diverse. Extending from citizen mobility, a significant area of study and many lines of research focus on the analysis for city planning and understanding internal functional city structures [43]. In general, even when using short-term data, the outcomes of these studies are for quasi-static or very long-term changes of city features.

An interesting line of thought found in the literature presents the concepts of the low frequency city and the high frequency city [44]. According to this study, most thinking about cities has been about how cities evolve in the long term: cities over years to decades to centuries to epochs. However, most control, but not planning, has been about the short term: cities over minutes, hours, days and years. Sensing what is going on automatically in cities is clearly something that happens in real time and this focus is changing from long to the short term, from years to seconds-minutes-hours-days-months.

The contributions of our dissertation are mainly focused on the high frequency city. Therefore, in the following section, we are going to focus on related work on urban-scale knowledge extraction from mobility data. The foundations of what we describe here are related to the topics addressed in the previous two sections. First, we will collate different existing methods to obtain spatio-temporal location data and to track real high frequency user movements inside a city. Afterwards, we will collate several data models and specifications constructed specifically for managing multiple transport modes available in cities. These models partly rely on digital maps that are enhanced by other data structures. These data structures encode and share the information that can be obtained from sensing utilities. In essence, they encode and share the expert knowledge on how that mode of transport works and how travellers in that mode of transport behave. Finally, we will describe methods and strategies for short-term traffic and mobility prediction. In this case, methods have evolved from the classic parametric spatio-temporal forecasting methods of the ARIMA family to more data-driven approaches.

2.3.1 Means of tracking location data for mobility analysis

A comprehensive survey published in 2015 collated several previous studies focused on data-driven human mobility modelling [45]. The mobility trace data classified in that survey differentiates: geo-location data from WLAN-, cellular-, and Bluetooth networks; and location check-in data offered by social LBS applications. The tracking system that promises the most significant results for the objective of multimodal urban mobility analysis is the use of mobile phone tracking data. Some authors gave an interesting overview of the outcomes and limitations of the exploitation of this data source from the Big Data perspective [46]. This is a passive location-based

tracking data source (users do not actively participate in sharing the location), and can be separated into two types of measurements: sightings and Call Detail Record (CDR). In sightings, localisation is computed using triangulation. They give higher temporal and spatial resolution than CDR. In CDR, each phone call is represented by a CDR. The resolution is lower, but user interactions are observable. Some authors have worked on the classification of the transport mode (automobile vs. transit) analyzing smartphone traces [47]. Another example of using mobile phone data is in combination with social media check-in data (with active user participation, much more sparse in space and time) [48]. The aim of the paper was to reveal urban functions and their diurnal dynamics. More concrete urban dynamics of pedestrians were treated by Maeda et al. [49] from mobile wireless networks. During the SARS-Cov-2 pandemic, mobile network operator data has been used, for instance, in studies that relate human mobility to the expansion of the virus, after a request from the European Commision to these operators to fully share anonymised aggregate mobility data for this purpose [50]. Nevertheless, mobile phone data, held by private network operators, is not usually available for city authorities in real time.

A study published in 2018 analyses the use of Google Location History as a novel source of information [51]. These data consist of geographic coordinates routinely recorded by Android phones, and are associated with a consolidated user account, allowing for location data that are recorded across all mobile devices that an individual has owned. It can provide a link between fine scale mobility and long distance and international trips, during long term observations of individual movements. However, for any study using such data, users need to download their associated data and provide it to researchers via surveys that include an appropriate informed consent process. Thus, real time availability of data is not possible either.

What is usually available is data from transportation operational systems deployed in the urban scenario, for example, traffic loop counters, cameras, public open wireless hot spots, smart card information and transit operator systems. These are very heterogeneous systems and offer data with very diverse spatial and temporal granularities, cadence, and nature. Examples of the use of this kind of location data are transit smart-card data [41, 52], bike-share rental data [53], tracked bike positions [26] or traffic travel time estimation, with probe data [54], or comparing connected vehicles vs. loop detectors [55]. Some other researchers [56] have studied the interrelation between modes (bike-train, in their study), but using user questionnaires instead of real tracked data.

2.3.1.1 Simulation models and tools for generating synthetic data

In some situations, it is not possible to get real data and thus, simulation tools can be used. Most existing simulation tools are built for road traffic.

Simulations can be categorised into *microscopic*, *mesoscopic* (vehicles can be grouped in packets and are treated as one entity) and *macroscopic* models, according to their level of detail. A new category can be found recently in the literature: *nanoscopic* models, which are extensions of microscopic models. Nanoscopic models are based on the combination of models of the vehicle, models of vehicle movement and driver behaviour. Two main mechanisms are found in the design of the flow of a traffic simulation: *event-driven* and *time-stepped* mechanisms.

Microscopic simulations take each vehicle as a single element with a full entity representing its position, speed, acceleration and heading. The most well known models of this kind of simulation are *car-following* [57], *cellular-automaton* [58], *lane-change* [59] and *route-choice* [60] models.

SUMO [61, 62] is a purely microscopic open-source traffic simulation. Each vehicle is given explicitly, defined at least by an identifier (name), the departure time, and the vehicle's route through the network. SUMO performs a time-discrete simulation with a default step length of 1s. Internally, time is represented in microseconds, stored as integer values. The simulation model is space-continuous and internally, each vehicle's position is described by the lane the vehicle is in and the distance from the beginning of this lane. SUMO uses an extension of the stochastic car-following model [63] per default. Car-following models usually compute the speed of the observed vehicle (ego) by looking at this vehicle's speed, its distance to the leading vehicle (leader), and the leader's speed.

AIMSUM is a simulator based on a modified Gipps' car-following model [64]. The Java Urban Traffic Simulator (JUTS[17]) is a pseudoparallel Java based simulation tool with graphical output, and it is based on the Nagel-Schreckenberg's cellular automaton model [65].

Large-scale simulations are computationally costly, especially microscopic ones, and therefore the use of parallelisation and distributed computation approaches are considered in some implementations. When using these approaches, two decisions have to be made:

1. How to split the simulation.
2. How to allow communication among processes.

For example, the work by Potuzak [66] presents a parallel/distributed traffic simulator created by an adaptation of the Distributed Urban Traffic Simulator (DUTS), and evaluates the improvement of the simulation performance using a parallel/distributed approach in comparison to a distributed-only approach. Other simulators that can be run using parallel approaches are SEMSIM [67] and TRANSIMS [68]. TRANSIMS is similar to JUTS but the main differences from the JUTS-based model are the larger traffic cells (7.5 meters, instead of 2.5) and only one length of the vehicles (corresponding to one cell).

[17] http://www.juts.zcu.cz

Apart from road traffic simulations, which are the most well known, other transport modes can also be found, for example, pedestrian dynamics [69]. Among more general multi-modal simulation and modelling software, we need to mention commercial EMME[18] or VISSIM[19] frameworks.

2.3.2 Urban Mobility data models and specifications

The following data models and specifications are some examples used in different cities around the world to structure and share mobility data from real time information from multiple sensors.

2.3.2.1 SharedStreets

SharedStreets[20] includes a street referencing system, non-proprietary data standards, aggregation, anonymisation and encryption utilities as well as data visualisation.

Launched in 2018, it is a project managed by the Open Transport Partnership, focused especially on traffic safety, real time traffic monitoring, and curb management, to digitally connect the public and private sectors.

We can set it between digital network map models (described in Section 2.1) and models used for encoding movement data.

2.3.2.2 General Transit Feed Specification (GTFS)

The General Transit Feed Specification[21] (GTFS) is a data format for agencies to publish their public transport information and allow developers to create third party applications. It is a set of text files, describing routes, stops, timetables and schedules of public transport services for each day. Established by Google in 2005, it changed its name from Google Transit Feed Specification to the current name in 2009, and it is a *de facto* standard nowadays.

There is an extension, called GTFS Realtime[22], which is designed to distribute up-to-date arrival and departure times.

[18] https://www.inrosoftware.com/en/products/emme/emme-modeller/
[19] http://vision-traffic.ptvgroup.com/en-uk/products/ptv-vissim/
[20] https://sharedstreets.io/
[21] https://developers.google.com/transit/gtfs
[22] https://developers.google.com/transit/gtfs-realtime

2.3.2.3 General Bikeshare Feed Specification (GBFS)

The General Bikeshare Feed Specification[23], known as GBFS, is an open data standard for bike-sharing. Its purpose is to provide the status of the sharing system at each moment (e.g. the number of available bikes or docks), through read-only data; it does not provide historical data services. GBFS shares real-time data feeds online with a uniform format.

Under the North American Bikeshare Association's leadership, since 2015, GBFS has been developed by public, private sector and non-profit bike-sharing system owners and operators, application developers, and technology vendors. Currently, it is used by 290 systems deployed in cities worldwide.

2.3.2.4 Mobility Data Specification (MDS)

The Mobility Data Specification[24] is a data standard and a set of three APIs for mobility for cities and service providers of dockless bike-sharing, e-scooters and shared rides. It was created by the Los Angeles Department of Transportation, but now, as of October 2020, it is managed by the Open Mobility Foundation[25].

This specification is inspired by GTFS and GBFS. Specifically, the goals of the MDS are to provide API and data standards for municipalities to help ingest, compare and analyse mobility as a service provider data.

MDS was, at first, comprised of two distinct components:

- The Provider API to be implemented by mobility as a service providers, for data exchange and operational information that a municipality will query.
- The Agency API to be implemented by municipalities and other regulatory agencies, for providers to query and integrate with during operations.

In 2019 a third component, the Policy API, was included, in order to allow providers query information about local rules.

At the moment, more than 90 cities and public agencies use these APIs, most of them located in the United States.

[23] https://github.com/NABSA/gbfs
[24] https://github.com/openmobilityfoundation/mobility-data-specification
[25] https://www.openmobilityfoundation.org/

2.3.3 Short-term mobility prediction

The most widely used application of mobility prediction by far is on vehicle traffic prediction, especially in interurban roads. However, the methodologies and algorithms used are transferable to other transportation modes. Traffic prediction is a highly researched area, road transportation being a transversal sector with high operational and socio-economic impact.

In traffic theory literature, the measures that raise the most interest can be grouped as [70]: rates of flow (vehicles per unit time), speeds (distance per unit time), travel time over a known length of road (or sometimes the inverse of speed, *tardity* is used), occupancy (percent of time a point on the road is occupied by vehicles), density (vehicles per unit distance), time headway between vehicles (time per vehicle), spacing, or space headway between vehicles (distance per vehicle) and concentration (measured by density or occupancy). In general, they can all be classified using three fundamental macroscopic traffic parameters: flow, occupancy and speed. These measures can be obtained based on observations at a point, over a short section, along a length of road or by the use of a moving observer. Using these measurements and their derived spatial or temporal aggregations, three main groups of approaches can be found in the literature: 1) building models that represent the behaviour of traffic, 2) detecting repeatable patterns in the measured data, and 3) forecasting future situations based on past evidence. When talking about prediction, we mostly focus on this third group.

With regard to the final approach, we can consider different prediction time-horizons:

- Long-term: Trends in metrics such as the Annual Average Daily Travels (AADT) may indicate that the travel demand of users of given road system is likely to exceed the available capacity. Decisions could be made about building a new road or increasing the capacity of an existing one by adding new lanes.
- Mid-term: Increase or decrease in travel demand may suggest changes in signalling and control of the infrastructure. For example, varying traffic light plans. A wide range of time-horizons, from hours to weeks or months, can be categorised here.
- Short-term: in the order of minutes.

Several studies have stressed the importance of making use of aggregated measurements, even though many systems are able to collect data in short intervals (few seconds) to overcome the strong variability of traffic parameters. Some authors use 5 minutes [71], others state the 15-min interval as the best prediction interval [72], and others recommended intervals not shorter than 10 minutes [73].

In addition to the selection of an appropriate time-horizon and aggregations, the selection of which measurement or variable to predict can sometimes be open. The

choice of the best measure to use is mainly related to the final purpose of the analysis or the prediction objective, but other concerns such as stability might be relevant. For example, Levin and Tsao [74] found flow-based forecasts much more stable than those based on occupancy. According to Lin et al. [75], flow data are inappropriate because the same flow level may correspond to either a congested or a free-flow traffic state. Thus, they suggested using occupancy, which is proportional to density, as a better indicator of traffic condition. Flow and occupancy estimations can be helpful in traffic control applications. On the contrary, travel time and speed predictions are more valuable for drivers [76].

One of the deepest literature analyses on short-term traffic prediction is the one published by Vlahogianni et al. [77]. The authors stated that the effort in most previous studies had gone into 1) using data from motorways and freeways, 2) employing univariate statistical models, 3) predicting traffic volume or travel time, and 4) using data collected from single point sources. The authors presented a list of 10 challenges in traffic prediction, related relevant literature and future research directions for each of these challenges. From their list of 10 challenges, the contributions of our dissertation are mainly related to the following two:

- Handling temporal characteristics and spatial dependencies
- Using new technologies to collect and fuse data

From the methodological point of view, two big separate approach groups are found in the traffic prediction literature:

- Classical statistical methods, lead by the use of Auto-Regressive Integrated Moving Average (ARIMA) family of models and their variants, as presented in the previous section.
- Data-driven methods making use of computational intelligent approaches where large datasets are required.

Karlaftis and Vlahogianni [78] summarise the bibliography from both approaches, with neural network (NN) techniques as the main representatives of the second group. The authors state that the common approach in literature to compare NN performance for time-series prediction is to test their accuracy against classic ARIMA models, but very few provide clear evidence and the results are *'confusing'* (some studies suggest NN are more accurate that ARIMA and others suggest the opposite).

Fusco et al. [79] propose short-term forecasts on urban road traffic networks by exploiting ubiquitous big data, focusing on congestion situations. They base their research on individual floating car data point speeds from a large fleet of private cars to compare data-driven models (Neural Networks and Bayesian Networks) with a Seasonal Auto Regressive Moving Average (SARMA) time-series model. The objective of Guo et al. [80] was to detect outliers, with 15-min aggregated flow series, following a Seasonal Autoregressive Integrated Moving Average plus Generalized Autoregressive Conditional Heteroscedasticity (SARIMA + GARCH) structure. The authors split

this structure into three components: a Seasonal Integrated Moving Average, a short term Kalman Filter and a GARCH filter. Connected Vehicle technology with Artificial Intelligence is used by other authors [55] and the generated data is sent to the edge devices (e.g., roadside units) for further processing: headway, number of stops, and speed data are used to estimate traffic density. However it is set in freeway scenarios.

Another study which is set in freeways is the one published by Zhang et al. [81]. They present a hybrid approach to explain the different components of traffic flow, combining a spectral analysis for intra-day periodic trends, an ARIMA model to predict the deterministic component in the residual part, and a GJR-GARCH model (an enhanced GARCH model first proposed by Glosten, Jagannathan, and Runkle) for the volatibility part of the traffic flow.

Habtemichael and Cetin [82] use 15-minute aggregated data to predict freeway traffic flow, based on a non-parametric method: an enhanced K-NN algorithm with Weighted Euclidean distance metric. Their method provides multiple time step forecasts (one hour and a half in steps of 15 minutes), but spatial features are not taken into consideration. They compare the performance against the methods proposed by Guo et al. [83]. One of their conclusions is that the accuracy of K-NN-based approach depends on the size of the search space, and thus, models based on time series analysis are prefered if the size of the available dataset is small.

These studies and many others, follow the univariate time series approach. However, some researchers have suggested the simultaneous forecasting of more than one variable to model the traffic dynamics [76]. For example, some authors [84] proposed a multivariate analysis based on three dimensions: flow, speed and occupancy (in motorways). Others [85] considered a spatio-temporal dimension of traffic, discovering that the more distant the used locations from the location of interest, the longer predicting horizons could be achieved. Nevertheless, they stated that this is not the case in urban areas, where the information from previous sites (located far away from the site of interest) is probably strongly distorted, given the interrupted nature of the flow in urban scenarios.

Other types of mobility data predictions follow similar approaches. For example, the parking prediction methods found in the literature are diverse. For instance, Tiedemann et al. [86] proposed a neural gas machine learning (a type of artificial neural network [87]) combined with data threads. The study described by Vlahogianni [88] is based on fine-grain data of individual parking space sensors, providing information on each on-street parking space every minute. Two types of predictions were provided in this case: 1) a short-term parking occupancy prediction in a given region, and 2) the probability of each free space to remain free in subsequent time intervals. Among the limitations they present, one is the lack of spatio-temporal traffic demand data use, which would enhance the predictability of parking occupancy. In other studies, parking occupancy is predicted by Long Short Term Memory (LSTM)

recurrent neural networks [89]. The authors focused on the capabilities of cloud computing and the ease of including new data into model without the need to retrain the whole model again from the beginning.

2.4 Short compilation of useful resources

In this section, we compile different resources that have been found promising for urban mobility research, even though we have not tested them all. For each, if used, we include a short description of its purpose and in which chapter it has been used.

- Kepler.gl[26]: Open Source python library for large scale geospatial data analysis and visualisation. We have not used this tool but the quality of visualisations and the interaction capabilities that it provides are very interesting for web-based visual representations of movement data. We envision to test it in future works.
- MobilityDB [90]: Open Source moving object database system implementation on top of PostGIS. Even though we have not used it for the purposes of this study, according to some preliminar tests, it seems to be a very valuable solution for urban mobility data management software development to handle moving elements and to provide an abstraction layer on top of a relational database.
- MovingPandas[27]: Open geospatial library based on Pandas and GeoPandas, that provides utilities for movement feature and trajectory computation such as length, duration, speed and direction. We have not tested this tool, but it is a project which is very active currently and the author promotes and disseminates many examples and tutorials.
- osm2po[28]: Converter and routing engine to read from OSM data and enable routable database creation over PostGIS. It has been especially used in Chapter 4 to build the routable road structure out of OSM raw data in order to enable the shortest path calculations needed for some map-matching algorithms.
- OSMNx [91]: A python tool for easy street networks analysis and management from data sources such as OSM. It has been used in Chapter 5 to build the network from OSM data and compute network distances among multiple locations.
- r5r[29]: Rapid Realistic Routing with R5 in R, for fast multimodal transport routing. It has not been tested but it is a recent project with routing capabilities to handle multiple transport modes.

[26] https://kepler.gl/
[27] https://anitagraser.github.io/movingpandas/
[28] https://osm2po.de/
[29] https://github.com/ipeaGIT/r5r

- Scikit-mobility [92]: a python library for the analysis, generation and risk assessment of human mobility data. We have not used it, but it is a very generalistic tool to operate with different kinds of mobility data.
- SharedStreets Trip simulator[30]: tool for generating realistic raw GNSS telemetry data. We have not tested this tool, but the capability of obtaining realistic geolocated data is very interesting. In Chapters 4 and 5, we have used the SUMO simulator to generate synthetic geolocated positions, adding random deviations to simulate positioning errors. Using this simulator could have been a good alternative.

From the beginning of the research covered in this thesis, there has been a significant evolution and emergence of software tools that simplify the various steps needed when handling geographical and spatial datasets for transportation applications. This small collection aims to list some of the relevant active tools and projects in the latest stages of the research.

2.5 Summary and main gaps

In the previous sections, we compiled multiple background concepts and related work published in the literature. From the analysis we have performed, we can summarise the following outcomes and gaps that lead towards our efforts in providing some new contributions to the body of knowledge on mobility mining for urban network modelling:

- Digital Transport Network Modelling: The analysis has collected multiple types of digital transport network models, depending on the data/map provider. There is a significant amount of (commercial) providers focused on vehicle traffic, which are mostly oriented towards navigation applications. Efforts on location referencing and the associated standardisation of maps and spatial information related to transportation infrastructure, are mostly focused on interurban traffic, which is not applicable in urban settings (interrupted traffic and high-level of multimodality and interaction among transport modes). This presents some challenges and specific solutions are still needed to extend the applicability of digital map representations to cities. In addition, as previously mentioned, the vast list of standards does not imply the existence of homogeneous or harmonised approaches that are widely adopted. Therefore, our contributions in Chapters 3 and 4 will be framed within this setting. Regarding data formats and technical approaches to store and index spatial information, the existing solutions are found flexible and capable enough to be used in our research, and thus, we select the use of relational database management systems (PostgreSQL with PostGIS) and graph databases (Neo4j) to implement

[30] https://github.com/sharedstreets/trip-simulator

the necessary developments. We have also collected some alternatives for large-scale graph processing that, although not used in this study in the end, open a path for future studies extending and evolving our contributions (particularly derived from Section 3.3).

- Spatio-Temporal Data Modelling and Mining: We first identified the characteristics of spatial and temporal data. Keeping their peculiarities in mind, both when handling them separately and jointly, is key to understanding some differences with other types of data. Then, in compiling the background and related work on this topic, we identified the main modelling approaches, derived from parametric ARIMA-family models. These spatio-temporal statistical models have been widely used and are still being used for demand prediction solutions and it is worth understanding their basics before going into more data-driven approaches, which are more widespread in current research trends. We have identified that the efforts in modelling spatio-temporal data over networks, which adds additional particularities and constraints, are limited in the literature, and this motivates us to work on data mining over transportation networks.
- Urban-scale Data-driven Mobility Analysis: We have collected different types of mobility data sources used in the literature, and existing software solutions to generate synthetic data. In fact, the availability of real data with sufficient granularity, in real time, is very limited, even for city managers. One of the contributions of this study will be to describe such experiences. "In addition, during the literature review, we analysed studies published on short term traffic prediction in urban mobility. In particular, we looked at one of the key aspects of great help in everyday resource management, from parametric models such as those presented for generic spatio-temporal data to data-driven models. The main efforts in the literature have gone into vehicle traffic flow or speed prediction (with urban traffic under-representated against interurban traffic), with limited studies combining multimodal data, particularly on multivariate time-series. In this sense, in Chapter 5, we contribute towards different perspectives on data-driven short-term predictions in cities by: 1) proposing a novel grid-based representation approach for small localised areas inside cities and 2) studying combined short-term predictions with traffic and parking data together with an evaluation of the interaction of the multiple measurement points deployed in remote locations.

Therefore, building on this analysis, and the gaps identified, we will now proceed to develop our main contributions in the following three chapters.

Representing the urban movement space 3

In this chapter, we focus on the digital representation of the transportation resources deployed in cities, which is the basis for Chapter 4 and Chapter 5 of this thesis.

After a first introductory section describing the main concepts and concerns regarding urban mobility infrastructure (3.1), in the following two sections, we will address the representation of what we call the Urban Movement Space, which is one of our contributions, from two points of view: using an external reference system (3.2) and taking the moving element as the reference system (3.3). Finally, we end the chapter with some conclusions in Section 3.4.

3.1 The urban mobility infrastructure

Cities are a complex set of interconnected mobility resources. Internal movements of citizens inside the city and their in-and-out movements to and from other municipalities for a wide variety of activities frame, to a great extent, everyday urban life. Urban planning has a significant impact on citizens mobility, since how residential, working or leisure areas are distributed, affects the demand for transportation. To address this demand successfully, providing efficient transportation infrastructure and services is essential from the point of view of city managers. It must be noted that the social and economic cohesion of cities and surrounding regions relies on a good-quality transportation network.

Transportation is usually classified into modes, according to the vehicle used to perform each movement, or the lack of any vehicle for pedestrians. We can make additional classifications separating transportation modes into private or public, collective or individual, motorised or non-motorised. However, over the last few years, the boundaries between these different classifications have become more and more blurred. There are several reasons for this including the growing use of shared modes and the new forth-coming Mobility-as-a-Service (MaaS) paradigms. On the one hand, shared modes avoid the need to own a vehicle (of any kind, e.g., bike, scooter, car) and the need to have a place to store the vehicle when not in use, such as a garage; however journeys are made individually. On the other hand, MaaS is slowly being tested in some cities, as a seamless flat-rate end-to-end mobility service that may combine multiple modes of transport for a single journey, according to the best alternative computed. The line between motorised and non-motorised modes is getting blurred by electro-mobility, given the significant increase of electric bikes and scooters in many countries. Other significant changes that need to be taken into

account when understanding the diverse urban mobility ecosystem, are the adoption of new ride-hailing services (in some places competing with traditional taxi services) and the unprecedented increase of last-mile goods delivery traffic, with a variety of vehicles, mainly due to e-commerce.

In addition to the vehicles themselves, it is also worth mentioning the physical space dedicated to various modes of transport. A significant part of the spatial extension of a city is dedicated to transportation and mobility resources, such as roads, parking lots, side-walks, train tracks, and stations of all kinds, among others. These physical resources have different functions and are sometimes shared by several modes of transport.

3.1.1 Decision making by urban mobility managers

Management of mobility in cities is often handled by a mobility department belonging to the city council, with several technicians responsible for different aspects. Everyday operation of many transport systems is handled under concessions contracted to private operators. What is most common nowadays is to have silos of information containing data obtained by each contractor individually and hosted in their software management systems. Technicians responsible for each transport system will usually have varying degrees of access to the management system data.

Many cities have begun investing in the implementation of software and hardware solutions as part of their evolution towards more integrated information acquisition schemes, and promoting data sharing requirements in concessions to allow for better management with real time monitoring and service accounting. This shift also expands the possibilities for citizens and third-party service providers to have access to open data. However, full integrated digital management of mobility data is still far from being achieved.

Many long-term mobility decision making strategies in cities, across the European Union, are framed under Sustainable Urban Mobility Plans (SUMP), oriented to urban transport planning. A widely accepted definition of a SUMP is, both in Europe and internationally, the following: "A Sustainable Urban Mobility Plan is a strategic plan designed to satisfy the mobility needs of people and businesses in cities and their surroundings for a better quality of life. It builds on existing planning practices and takes due consideration of integration, participation, and evaluation principles." [93]. SUMPs rely on eight main principles (see Figure 3.1). SUMP-based decision-making may involve operations such as street pedestrianisation, inclusion of new shared-mobility services, building new lanes or restricting some of them to public transport, changes at signalised intersections and many other operations some of them tightly linked with urban planning policies. The definition and monitoring of quantitative Key Performance Indicators (KPI) is one basis to determine if objectives established inside these sustainable mobility plans are achieved. This leads towards

Figure 3.1: Eight principles of Sustainable Urban Mobility Plans (SUMP)[a]

[a] https://www.eltis.org/mobility-plans/sump-concept

data-driven decision making, with improvements of city mobility with a global vision in mind.

Short-term decision-making is usually related to day-to-day operations, involving various procedures required to organise resources before planned events or to overcome unplanned incidents. These are the most well known uses and benefits of most intelligent transport systems, since they help detect anomalies, provide real-time information, and help automate prioritisation strategies. From the current use of prioritisation, understood as deciding who needs to go first at an intersection controlled by traffic lights, further evolution should be expected in the near future. This would involve prioritising any shared network resource to optimise the net flow of travellers. This is a very challenging topic, not only technically but also considering citizens' expectations and willingness to change. However, it is still far from being solved largely due to the fact that multiple modes of transport are involved.

3.1.2 Available transportation resource data in cities and main challenges

Based on the experiences of and discussions with several city council mobility departments in Spain during different research and innovation projects leading to this thesis, we summarise the type of information that is usually available and the

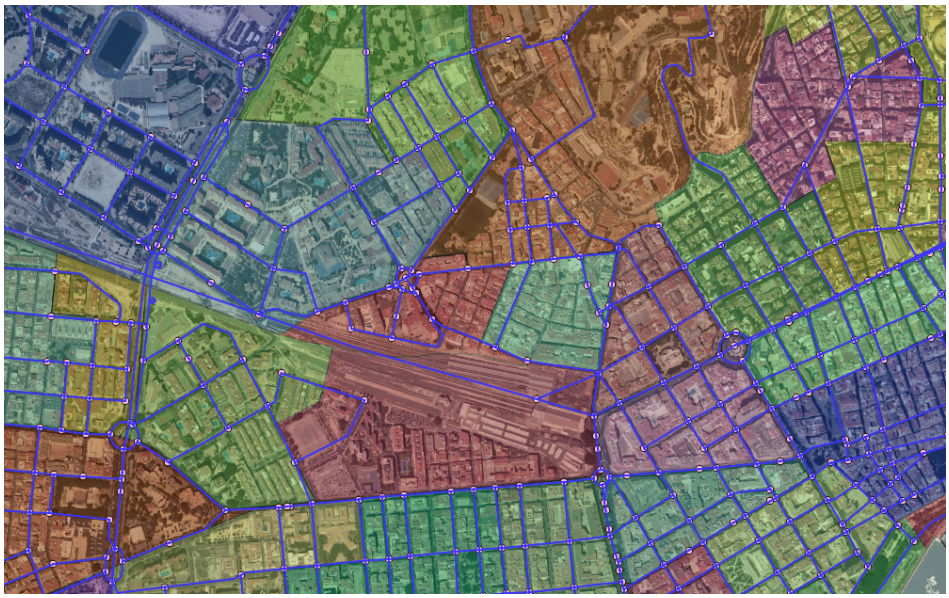


Figure 3.2: Example of transport network model exported from VISUM and processed into a set of interconnected ways, together with transport zones, shown on QGIS software.

main challenges when trying to integrate all the data into a common transportation spatial network representation.

- Street network: Cities that make use of mesoscopic modelling or simulation tools or have gone through an integrated mobility study have, in general, an accurate digital representation of the street network graph, although the smallest streets may not be included. In some cases, current public transport supply data is already incorporated. Depending on the simulation tool used and its format, this potentially powerful and complete dataset may or may not be eligible for being processed by other software tools. For instance, in one of the cities, we processed the VISUM* network, exported in .NET format, which is a plain text format, and loaded it into a PostGIS database (Figure 3.2). A significant challenge is the management of any changes that happen in the network. As far as we know, cities do not have any established procedure to handle these modifications.
- High definition street geometry: This information, containing detailed geometry and lane widths, can be key for microsimulation models. However, it is not usually available at the city level, as it is often used for specific construction work projects (e.g., building a new roundabout). This data can sometimes be obtained through High Definition digital map providers for navigation purposes or autonomous driving research, but only for vehicle traffic.

* https://www.ptvgroup.com/en/solutions/products/ptv-visum/

- Public transport resources: Public transport operators are usually managed with the use of a proprietary Intelligent Transport Aid System, handling and publishing the definition of stops, routes, stop-times, calendar information, as well as the geographical shapes of route itineraries. The main challenge is when multiple operators run in a city, for instance, when urban and interurban bus routes are operated by different agencies with different coding schemes. In some regions, a supra-municipal entity ensures a smoother integration.
- Parking locations: On-street and off-street parking data information is, in general, limited, at least with respect to being integrated spatially and topologically with the rest of the network. Off-street parking locations are, in general, represented as a single location point, but entry and exit accesses that link the parking lot to the street network are also very relevant in order to encode the relation between traffic and the parking situation. On-street parking locations are usually provided as PDF-based maps, and the relation to the road transportation network is not digitalised.
- Bike-lane network: Some cities have a significant bike network but the digital representation is sometimes limited to PDF-based maps, with no easy means of automatically importing or conducting further integrations. What is usually available is the location of the shared bike pick-up and drop-off points, since this is handled by contractors who often use a management system to monitor the use and availability of bikes.
- Pedestrian infrastructure: As far as we are concerned, digital representations of pedestrian areas for urban mobility management are rare, apart from 3D approaches using CityGML[†] or autonomous driving research.
- Zoning: Spatial subdivision inside cities is key to understanding mobility origin and destinations, also known as travel generation and attraction zones. Multiple administrative boundaries exist inside a city (e.g. neighbourhoods, census, districts), and usually they are not representative of the natural urban division when it comes to mobility. In mobility studies, the concept of Traffic Assignment Zones (TAZ), is well-known. There are multiple challenges and drawbacks when selecting a single zoning strategy, since it may not be representative for all modes of transport and, in addition, the assignment of data near these boundaries could also be inaccurate. However, the lack of a unique zoning strategy limits the possibility to address all mobility modes as a whole, in an integrated manner.

Rich crowdsourced data sources such as OpenStreetMap (OSM)[‡], must also be taken into account. They can be used as valid data providers to some extent, even though the quality and coverage of the transportation network by OSM is not always as complete or detailed as other proprietary solutions, in particular with regards to the road network (we refer the reader to studies that have reviewed some comparisons

[†] https://www.ogc.org/standards/citygml
[‡] https://www.openstreetmap.org

(a) VISUM network with common spatial geometries for road centre-lines and no bike-lane representation

(b) OSM network including separated geometries for two-way streets and detailed bike-lane

Figure 3.3: Sample comparison of transportation network sources

[9], [94]). However, in many situations, some mobility infrastructure elements that the original municipal source does not provide in digital format (for example, parking locations and bike-lanes) can be found in OSM. As a comparison, in Figure 3.3 we show the same given area in a city and the information about the transportation resource encoded by the VISUM network that the city council handles and OSM. Some substantial differences can be seen, for instance, when representing two-way streets; in the example given, OSM separates two different spatial geometries, one for each direction, while the network used in VISUM does not.

In order to accommodate different approaches to represent all the transportation resources, we propose a conceptual representation model that covers all the main heterogeneities and we set a common framework that will allow users to work with mobility data from multiple transport modes and input data sources.

3.1.3 The Urban Movement Space - Proposed concept

The representation model that we propose, called the 'Urban Movement Space' (UMS), is fed by the cartographic information of physical resources and planning data from non-physical resources (service schedules). This model can be seen as a black-box that provides infrastructure network information to another big black-box: the data mining process and algorithms. These processes and algorithms are being constantly fed by input data from several locations and time instants. Real time processes extract knowledge from current situations obtained from movement data, which is analysed in Chapter 4, and extend updated information to the Urban Movement Space, such as varying network edge costs (e.g. increase in travel time).

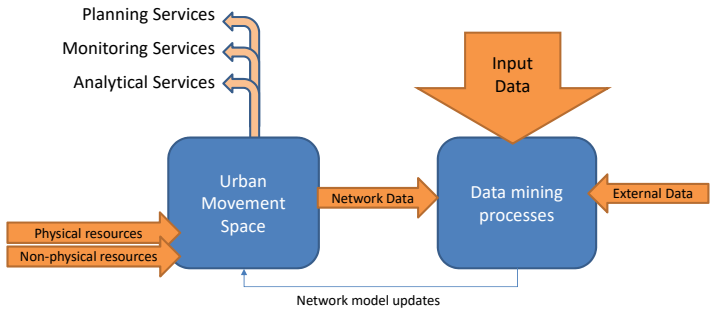


Figure 3.4: Conceptual diagram of the UMS model. It is conceived as a set of two main big functional blocks which feed one each other: the Urban Movement Space and Data Mining processing algorithms.

With this timely information, the Urban Movement Space records a history of the network situation, and is able to provide further services: planning, monitoring and analytical services. Our focus throughout this thesis aims to propose and discuss the most relevant representation aspects of this Urban Movement Space to enable further spatio-temporal data mining processes with a holistic view of multi-modal mobility in a city. The high level concept of what we propose is depicted in Figure 3.4.

We envision the UMS as a set of physical and non-physical resources devoted to the movement of citizens inside the administrative boundaries of a city. We use the concept of physical resources or infrastructure when we refer to fixed infrastructure such as streets, bike-lanes or side-walks. Access to these resources is not affected by any temporal constraints. On the contrary, we use the concept of non-physical resources to refer to scheduled services, such as public transportation routes.

The usefulness of UMS is having a representation model of a city's transportation supply into which we can match urban presence and movement events from a multi-modal perspective. Our approach considers the following transportation modes, which cover the main heterogeneities in urban mobility representation:

- Road traffic: Vehicles that move along the street network, whose journeys always begin and end in on-street or off-street parking.
- Transit (for the sake of simplicity, we limit our current approach to buses): Scheduled vehicles that move along the street network, stopping at predefined stops to pick-up and drop-off passengers.
- Bike sharing: Movements from origin to destination between predefined bike-sharing stations, limited to the availability of bikes (at origin) and free-slots (destination). They may use dedicated lanes, if existing, or share the road with other motorised traffic. Other shared modes (motorbikes or cars) may exist, but modelling would be equivalent.

- Pedestrians: Presence and flow of pedestrians in free-spaces, constrained mainly by obstacles such as buildings and roads.

3.2 The Urban Movement Space – the Eulerian point of view

Borrowing terminology from fluid mechanics, the flow of citizen movements can be eulerian, thus referenced to a fixed external reference system, or lagrangian, where the moving element (e.g. citizen, vehicle) is the initial point of the reference system (Section 3.3). In this section, we will focus on the eulerian reference system.

The infrastructure resources of each transport mode belonging to the UMS can be represented separately. For a given trip, citizens are able to perform modal change from one infrastructure to another. However, we stress the concept of *access to the infrastructure*. Network-based infrastructures can only be accessed through access nodes and by the corresponding allowed mode-vehicle. For instance, pedestrians are not allowed to access the road network without a car, and the other way round. A significant implication of this is that the transport modes that make use of a personal vehicle (private car or bike-sharing bicycle), always need a spatio-temporal transition for interchange: e.g., searching for parking to leave the car before getting on a bus, or locking the bike to the nearest available sharing station.

We have mentioned the boundaries of the city as the limits where the UMS is confined. However, a collection of several UMSs can be used to create a region that covers a larger area. Therefore, two UMS systems are said to be connected, if there exists at least a pair of inbound/outbound edges between them. Inbound and outbound movements may happen along inbound/outbound edges.

The first basic representational approach of the UMS is to describe it as a multilayer super-network combined with euclidean planes. As mentioned, transitions between networks must be done from an exit access node from the origin network to an entry access node in the destination network.

Figure 3.5 represents a holistic view of the urban transportation resources. In Figure 3.6, the separate layers are displayed.

With the representation model of the urban movement space, we aim to define how to measure the spatial heterogeneity of urban mobility inside the city. For instance, bike lanes may have similar speed limits along the whole network. However, speed limits or street capacities for vehicles may vary from edge to edge. Also, the existence or not of on-street parking, can greatly affect the behaviour of a road street edge, due to the effect of cruising for parking.

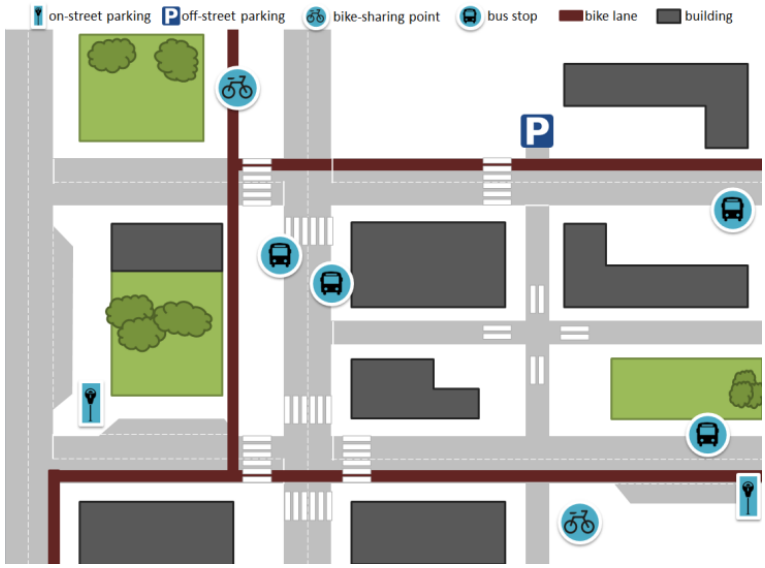


Figure 3.5: Schematic diagram of sample urban infrastructure resources for mobility. The same space is shared by several transport modes. The diagram depicts walkable side-walks and road-crossings for pedestrians, bike-sharing stations and exclusive bike-lanes, street road used by bicycles, private traffic and public transport vehicles, parking facilities, and public transport stops.

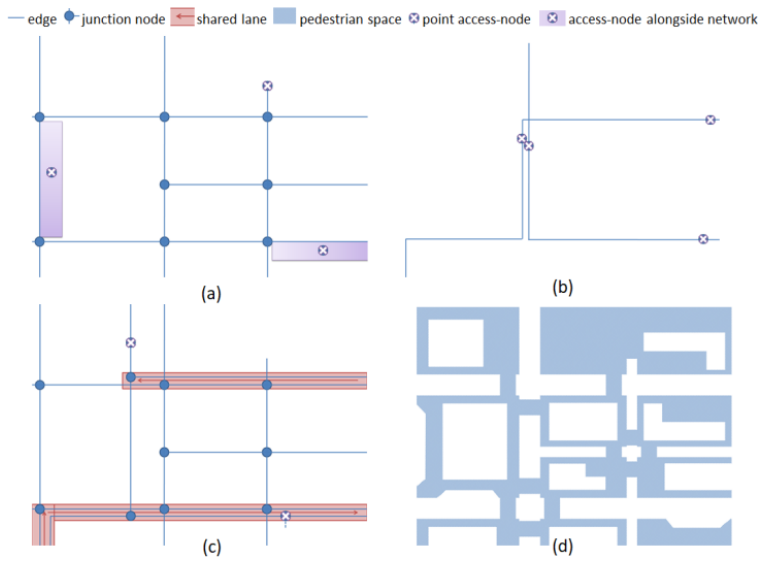


Figure 3.6: Networks that represent each transport mode: road traffic (a), transit (b), bike (c) and pedestrian side-walks space (d).

The same situation observed through the available measured data may represent an anomaly under certain circumstances, or a normal situation under others. To give an example, a downtown area where cars are banned on weekends, should not cause alarm when there is a drastic reduction in vehicle counts. This is why the city mobility resources supply must be digitally encoded to allow for a range of analytical models.

3.2.1 Modelling transportation resources

Our UMS approach makes use of different data models for each kind of physical resources.

Street networks have historically been modelled using a topological representation based on road arcs as edges and junctions as nodes, especially in GIS. More recently, the complex network theory approach is being expanded. In this case, a dual representation is used, where roads (or streets) are represented by nodes, and a link between two nodes implies that two streets are connected. In our proposal, the road traffic network is a non-planar geometric graph $G(V, E)$, consisting of directed network edges (E: e_1, e_2,... e_n) and nodes (V: v_1, v_2,... v_m). The main attributes of edges are the unique id of the edge id_{e_i}, road name or OSM identifier $osm\_id_{e_i}$, length of the edge l_{e_i}, nominal speed v_{e_i} and cost in travel time for both forward (cf_{e_i}) and backward directions (cb_{e_i}). Additionally, the spatial linestring geometry is also represented as $geom_{e_i}$. The main attributes of a node are its unique id_{v_j} and the spatial point geometry $geom_{v_j}$. Junctions are represented by nodes and are modelled as joint-nodes among at least three edges, where a driver has the choice of continuing towards one direction or towards another. Two edges can be connected by a continuity-node, when there is a need to separate a road segment into smaller segments to represent different non-spatial attributes (such as speed limit, or number of lanes). The road traffic network includes two types of access nodes in our schema: point access-node and access-node alongside network. The first one represents entries and exits for parking sites. The second one, represents on-street parking, where vehicles may park aside the road.

The shared bike network is also a non-planar geometric graph. It is peculiar because it can be made exclusively of bike-lanes, but at the same time, bikes can share the same road traffic network. Access nodes are, when referring to bike-sharing, point access-node stations where bikes can be locked and unlocked.

Access nodes that rely on offered resources need to be represented with an access-node status property that changes over time (ACTIVE/INACTIVE), which will be provided by the input data sensor associated with the location.

Pedestrian space is modelled using raster data. Each pixel represents if the geographical position is part of a pedestrian side-walk or other kind of open spaces available for citizens, such as public squares.

We recognise that public transportation has a dual nature. On the one hand, it relies on a physical resource infrastructure, made up of roads and stops. The existence of this physical resource is important, since passengers coming from other modes, need to join the public transport network at some point. However, movements between two locations only happen at predefined times. Any deviation from the expected arrival time at the entry access node, is non-linear with the arrival time deviation at the exit access node.

The definition and representation of schedules and service lines can be extracted from data provided in General Transit Feed Specification (GTFS) model format. It should be noted that schedule plans are prone to frequent minor changes (at week or month scales) and this needs to be taken into account when handling temporal variability. Stops and line connections among them are modelled using geometric graphs.

In classical transportation engineering space is usually divided into fixed Traffic Assignment Zones (TAZ). They are widely used to model demand by origin-destination matrices. In the UMS approach, our proposal is to use time-varying sets of vector Voronoi polygons, one set for each mode of transport, except in the case of public transport, where a different time-varying polygon set is proposed for each line. Access-nodes with the status INACTIVE are discarded for the generation of the polygon set. This more faithfully models the real accessibility of citizens to each of the resources. Data associated with Voronoi polygons from different transport modes in a given time frame, when spatially overlaid (Figure 3.7), are combined based on the proportional areas they share.

In addition, several levels of spatial administrative boundaries or zones are defined to match locations with different city regions. These are represented by vector polygons: city, district, neighbourhood or census section. The upward/downward relations between hierarchies might not be direct. Therefore, input data needs to be matched with the preferred one in each case.

3.2.2 Definition of time-referenced neighbourhood

One of the key representations of relations in space (both euclidean and network space) is the neighbourhood. It is usually modelled as an adjacency matrix, where pairs of entities are given a boolean or numerical value to address the level of proximity between them.

In the UMS frame, dynamic neighbourhood matrices are considered. The dynamism is the result of the time-varying situation of the infrastructure due to planning

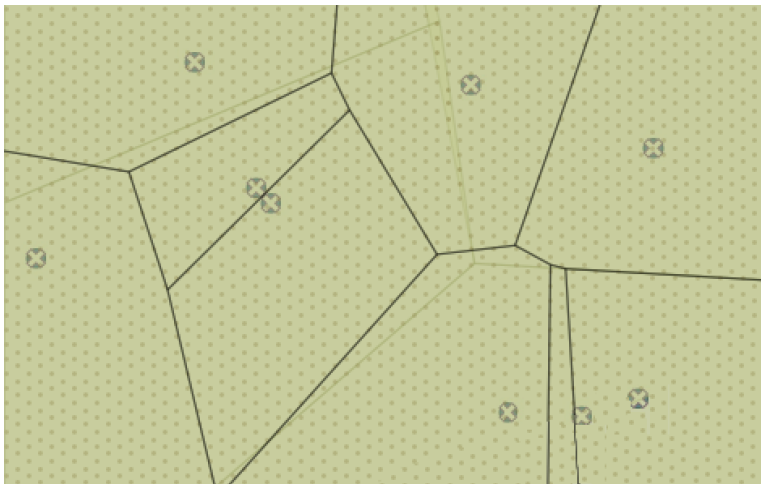


Figure 3.7: Two sets of Voronoi polygons, overlaid. The assignment of functional areas linked to each access-node, must be made according to the transport mode, and the real availability of the access-nodes.

(known in advance, such as the existence of an additional bus stop in certain location, that operates only at certain times) or due to operation (a road may get congested and therefore the distances among nodes nearby increase). Thus, we assign a time index to them.

We find the need to represent three main neighbourhood distances:

- Node neighbour spatial distance: based on the geographical graph representation, the computed distances are network distances.
- Node neighbour time distance: based on the geographical graph representation, the computed distances are network travel times.
- Edge neighbour distance: based on the graph representation, the computed distance is the minimum number of nodes in the shortest path.

3.2.3 Handling temporal variability and seasonality

A significant challenge is the representation of unstable and changing environments. Streets are assumed to be fixed networks, but construction/maintenance works may require non-periodic closures of some edges. In addition, planned periodic closures may exist in some areas of the city, limiting car access. Traffic management systems, as well, could change lane orientations to cope with variable demands throughout the day.

Scheduled transit services are also modifiable in time. Our approach considers the generation of planning versions. Each time a new change is received, a new version

tag is created and version history is kept. Any measurement from input data (GNSS position, or smart-card record) will be tagged with that version.

The representational model includes two separate entities for time:

- DATE: Day level time granularity is stored. Each day is represented by year, month, day properties, as well as week-of-year, quarter, specificities of the day inside a week (from Monday to Sunday) and class from Working-Day, Saturday or Holiday. A numerical index in YYYYMMDD format is used.
- TIME: Considers all minutes within a 24-hour day. Three grouping granularities are provided: time-frames of equal size (night, morning, afternoon-evening), 15-minute groups and 1-hour groups. A numerical index in 1HHMM format is used.

This means that the minimal time granule provided by the UMS in a consolidated model is 15 minutes.

The two time-entities allow for the representation and querying of temporal references into three perspectives:

- Linear time-line: Represents the ticking of time, as a sequence of snapshots when the graph is observed or modified.
- Hierarchical datetime-tree: Represents different date and time scales.
- Periodic time-cycles: Represents repeating time patterns.

3.2.4 Application use cases

We present three different application use cases where the UMS represents both the network and the tracked data in a suitable way for Spatio-temporal Data Mining.

3.2.4.1 Origin-destination flows

In public transport, many stops may act as a same conceptual stop. Some approaches found in the literature [41] make use of Thiessen (Voronoi) polygons to assign subway stops to population data. This implies that travellers always choose the closest stations, which may not always be the case. For example, as shown in Figure 3.8, depending on the bus line needed, a passenger may not have the choice of using the closest stop. This is overcome in the UMS representation by the use of multiple Voronoi polygon sets for access-nodes.

The same issue occurs with private vehicles. For similar pricing policies, the distance between a public parking lot (on-street or off-street) chosen by a driver to finish a trip, and their final destination is likely to be positively correlated with the overall demand for parking in the area. This is transferable directly to the bike-sharing scheme. The

Figure 3.8: Two separate transit stops that may act as a same conceptual stop. Since they belong to different lines, a passenger has no choice for selecting Stop A or Stop B.

presence of INACTIVE access-nodes, makes the polygons greater in size, and that way, we avoid data being unrealistically associated to a given origin/destination.

3.2.4.2 Spatio-temporal outliers

The discovery of spatio-temporal outliers inside the urban movement space is a relevant application use case of spatio-temporal data mining algorithms. Outliers represent locations where the measured non-spatial attributes have a significant discontinuity to what is being measured in the spatial and temporal *neighbourhood* locations. The approach consists of comparing measures against neighbour aggregates. Given that our representational model provides time-referenced spatial neighbour matrices, comparisons can be computed among sensor data that share the same time-frame (or a comparable time-frame in periodic patterns).

3.2.4.3 Co-locations and sequential patterns

Another kind of application provided by data mining algorithms is the detection of patterns where attributes measured in a certain location at a certain time imply a strong association with attributes measured in another location simultaneously (co-location) or with a certain time lag (sequential pattern). In urban mobility, the discovery of these patterns is very valuable to help with planning and decision

making for certain events, such as traffic congestion, the detection of bike-sharing stations that empty with no free bikes or an increase or decrease in parking lot occupancy. In Chapter 5, we will discuss the interaction among measured data at remote locations inside a city with a practical example of traffic and parking data for short-term estimations.

3.3 Local Dynamic Maps - the Lagrangian point of view

In this section we discuss some concerns to tackle the Urban Movement Space from a lagrangian point of view, where a moving element is the origin of the reference system, thus being able to enhance a Local Dynamic Map system.

The concept of Local Dynamic Maps (LDM) is expanded upon in in-vehicle navigation systems and autonomous driving research. The particularity of LDM is that each vehicle hosts one, and it relies on local map data of the surrounding area as it moves. ITS stations nearby may be involved in updating maps in connected vehicles for static or even dynamic data when very low communication latencies are available, such as those expected with 5G and beyond. LDM has already been considered part of some standardisation activities, such as:

- ETSI TR 102 863 (V1.1.1): Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Local Dynamic Map (LDM); Rationale for and Guidance on Standardization (2011).
- ISO/TS 17931:2013 Intelligent transport systems - Extensions of map database specifications for LDM for applications of Cooperative ITS. (Currently withdrawn)
- ISO/TR 17424:2015 Intelligent transport systems – Cooperative Systems – State of the art of Local Dynamic Maps concepts.

Four main layers make up the core of a LDM system. As given by ETSI TR 102 863, these layers have different objectives and are in charge of storing and serving digital map data according to their variability in time:

- *Type 1*: permanent static data, usually provided by a map data supplier.
- *Type 2*: transient static data, obtained during operation, e.g. traffic signs, gantries and changed static speed limits.
- *Type 3*: transient dynamic data, e.g. weather situation, traffic information, position, lane width, speed limits and incidents.
- *Type 4*: highly dynamic data, e.g. vehicles and dynamic traffic signs.

The UMS is particularly relevant to build and enhance Type 1, Type 2 and Type 3 layers. All measurements and sensor data that this *ego-vehicle* collects, such as surrounding vehicles and pedestrians, are able to feed the Type 4 layer of the LDM in real time.

One of the main benefits of a LDM is the ability to estimate the most probable trajectories of other actors (vehicles, pedestrians and obstacles) in near real-time in order to evaluate risks and decide on the safest and most optimal driving maneuvers. This is provided in particular by the use of the Type 4 layer. This kind of application is a combined use of digital maps and Advanced Driver Assistance Systems (ADAS) and it is sometimes called *eHorizon* [95].

The UMS approach, which arises from a global city mobility management perspective, is not expected to cover all the requirements of a LDM, but the synergies are significant and will be described in the following subsection.

3.3.1 LDM layers and the UMS model

In this section, we map each of the layers with the concepts that we have introduced when proposing our UMS representation model. The information handled by the Urban Movement Space is a valuable resource to feed local dynamic maps, with data otherwise not accessible from each moving vehicle. The UMS provides an integrated digital representation of the mobility network designed to enable mapping data from various city sensors, and this representation inherits knowledge from historical data that can be shared with the LDM.

3.3.1.1 Type 1: Permanent static data

The geometry and topology of physical urban mobility infrastructure resources of the UMS model belong to this layer. In particular roads and streets, pedestrian areas/side-walks, bike lanes and shared-bike stops as well as public transport stops.

3.3.1.2 Type 2: Transient static data

Public transport route schedules and shared-bike rental opening time are clear example of the data found in this layer. They enable or disable some important features of the infrastructure resources, according to the time of the day, limiting what we call the UMS access-node status.

3.3.1.3 Type 3: Transient dynamic data

Public transport stop times are the most relevant examples of transient dynamic data. These timetables are significant indicators of the potential presence of a vehicle at a stop location and, as a consequence, an increase of pedestrians in the surrounding areas, perhaps waiting for the bus or after being dropped off. As we will discuss in Chapter 4, we will feed UMS with mobility data. According to the data, patterns of

Layer	Group	Entities
Type 1	Roads	Way, Intersection, Lane, Parking
	Pedestrian	Area, Sidewalk
	Bike	Stop, Lane
	Transit	Stop, Route, Lane
	Other	Buildings
Type 2	Roads	Works (road closure), Parking hours
	Bike	Bike rental working hours, Works (lane closure)
	Transit	Route timetables
Type 3	General	Weather status, Environmental event
	Roads	Traffic light, Slippery road, Accident, Black Spot
	Transit	Stop timetables
Type 4	Ego-element	Ego-car, Ego-pedestrian, Ego-bike, Ego-bus
	Moving element	Vehicle, Pedestrian, Bicycle/electric scooter, Other
	Fixed element	Obstacle

Table 3.1: LDM layer types and modelled entities

demand that change throughout the day can be analysed and estimated in advance. These varying patterns can be another potential source of transient dynamic data.

3.3.1.4 Type 4: Highly dynamic data

The definition of the UMS itself does not provide any information that is able to feed the highly dynamic data. However, some mobility data that is collected in real time in cities, and mapped over the UMS model, can be representative of highly dynamic data to some extent: e.g., Floating Car Data positions.

The classification of the proposed model entities in each layer is depicted in Table 3.1.

3.3.2 Implementation concerns

The original purpose of the UMS representation is to handle the city transportation network as a whole. The resource elements for each mode of transport (used individually or shared by multiple transport modes), are initially built inside the model (the resource inventory), and subsequently updated if any changes occur. Information retrieval from the UMS can be done by requesting these inventoried resource elements using their unique codes, if known, or by location. To enable so, UMS storage is implemented over a relational database management system (RDBMS) with spatial capabilities: PostgreSQL + PostGIS.

The local LDM-oriented approach relies mostly on the relations among elements nearby, both in space and time. This is why the use of graph databases is an interesting option, since these types of databases are especially designed for efficient storage and retrieval of relations between entities. In existing LDM implementations, different options are found, with the newest ones using graph databases ([96–98]).

The transportation infrastructure network, as mentioned, is a directed graph, represented by a set of nodes and vertices, associated with a collection of numerical and categorical properties. Multiple alternatives have been found and tested to build this network, with Neo4j[§] selected as the graph database server, and importing data from OSM as the main road network data provider. Among the alternatives, we can mention the following:

- Osm2po, Moeller (2018), is a converter and routing engine. It allows filtering from OSM attributes to build a routable network. Then, we import its output manually into Neo4j. When representing this road network, we evaluated two options: 1) Intersections and ways, both as nodes, and add relationships among them to indicate heading, 2) Intersections as nodes, and add ways and their properties as relationships between intersections. The first variant adds complexity to the network since the number of nodes is significantly higher, but it is more flexible when querying the graph.
- Spatial plugin for Neo4j. This option avoids the need to pre-process the OSM file. Due to the spatial functions that the tool provides, the process is easier and faster. However,the way of representing the information has been found to be less intuitive. In addition, intersections are not easily identified, thus limiting some route analysis.
- Additional tools such as https://github.com/neo4j-contrib/osm

3.3.3 Time-varying representation

Context is variable in time. Time variability and granularity are greatly linked to each of the layer types. In general, at some point in time, an element may be located at a given position, and in another position at another point in time. Moreover, relations may change as well. An element that is known by another element now, may have not been known yesterday. Therefore, there is a need to handle the continuous progression of structural modifications of the graph.

From the four basic layers, three of them (Type 2, Type 3, Type 4) are, by definition, time-varying. Even Type 1 could somehow be represented by a given time point in history. We find several possible ways to model time. Depending on the data being represented, one option may be more suitable than other.

[§] https://neo4j.com/

In our model, we decided to consider three kinds of representations for time, the same way as in the Eulerian point of view. The only difference is that the implementation of the entities is different as we are using graph databases:

- Linear time-line: Represents the ticking of time, as a sequence of snapshots when the graph is observed or modified. It is represented by an ever-increasing integer. It could be an absolute value or relative value:
  - Example of absolute: Unix Timestamp, which represents seconds since Jan 01 1970 (UTC).
  - Example of relative: Seconds since the beginning of the current day; Seconds since the beginning of the vehicle journey
- Hierarchical datetime-tree: It represents different date and time scales:
  - The largest date unit is *Year*. Several years are linked sequentially. From each *Year*, a set of smaller time units is created: *Month*. From each *Month*, a set of smaller time units is created: *Day*. Several months are linked sequentially. Several days are linked sequentially.
  - Time is divided into *Hours* and *Minutes*.
  - Information attributes can be set, for instance in *Day* elements: *Working*, *Saturday*, *Holiday*
- Periodic time-cycle: Represents repeating time patterns:
  - It does not represent any concrete date in history. Valid for dates and for time of day. It is used to represent repeating seasons, and are a set of finite elements that are cyclic. Nevertheless, custom labels can be created:
    - *Q1*, *Q2*, *Q3*, *Q4*
    - *Monday*, *Tuesday*, *Wednesday*, *Thursday*, *Friday*, *Saturday*, *Sunday*
    - *January*, *February*, *March*, *April*, *May*, *June*, *July*, *August*, *September*, *October*, *November*, *December*
    - *Night*, *Morning*, *Noon*, *Afternoon*, *Evening*
    - *Night*, *Day*

Eventually, we may need to discard parts of the graph if their relevance in time has expired. For instance, in real time applications a vehicle may not need to handle information about Type 4 elements that were recorded yesterday. For long-term analytic use case applications, this may or may not be applicable.

3.3.4 Concerns on graph partitioning

In order to be locally useful and to reduce non-relevant distant data, graph partitioning strategies must be taken into account. A graph partition problem is defined by data represented in the form of a graph $G = (V, E)$, with V vertices and E edges, such that it is possible to partition G into smaller components with specific properties. Graph

partitioning can be of special interest to apply divide-and-conquer algorithms or for parallel computing. In our concrete problem of designing a LDM, graph partitioning must be allowed to enable the following possibilities:

- Memory requirements: In-vehicle applications embedded in limited hardware may benefit from using reduced data.
- Speed: many operations (e.g. map-matching) can be significantly faster if only a subgraph that represents the surrounding environment is being handled.

We must note that memory and speed are also to be considered in the Eulerian point of view, but the requirements are more restrictive in the Lagrangian version, especially when considering hardware limitations since the LDM might be located embedded in a vehicle or a mobile device. The main kind of partition to be considered in a LDM is the spatial partition. The objective is to serve data according to the nearest information as the ego-vehicle or user moves. There is an additional approach, however, which is not exactly graph partitioning: multitenancy. Multitenancy deploys multiple independent instances or multiple applications sharing a single environment. This is relevant in order to serve other kinds of user-specific data or information. We have collected different alternatives, which are dependent on the final deployment preferences. For instance, using Neo4j as a base implementation reference for LDM, we find that multitenancy is not supported as part of its core features at this moment. We proceed to present some work-arounds, divided into logical partitions and physical partitions.

3.3.4.1 Logical partitions

We define two lines in this group: first, partitions that are created by the user making use of specific queries; second, partitions created by external tools.

- User-created partitions:
  - Using labels: Graph partitions are created using a different unique identifying label for each different set.
- Partitions created by external tools:
  - Blueprints and PartitionGraph from Tinkerpop[¶]: Blueprints is a graph database interface that makes easier to build multi-tenant graph applications. There are several benefits to using Blueprints. For instance, it supports several graph databases, including Neo4j. Blueprints also includes a collection of functions (graph wrappers), that enable the creation of new features and implementations of existing graphs. One of them is the PartitionStrategy, which partitions the vertices and edges of a graph into String named partitions. This is very similar to the previous option

[¶] https://github.com/tinkerpop/blueprints/wiki/Partition-Implementation

of using labels in Neo4j. There are three primary configurations in PartitionStrategy: 1) Partition Key (the property key that denotes a String value representing a partition), 2) Write Partition (a String denoting what partition all future written elements will be in), and 3) Read Partitions (a Set<String> of partitions that can be read from). By writing elements to particular partitions and then restricting read partitions, the developer is able to create multiple graphs within a single address space. Moreover, by supporting references between partitions, it is possible to merge those multiple graphs (i.e. join partitions). There is another concept in Tinkerpop3: SubgraphStrategy, which is quite similar to PartitionStrategy in that it restrains a Traversal to certain vertices and edges as determined by a Traversal criterion defined individually for each. This strategy can be very useful to obtain subgraphs of elements according to their time-varying properties.

- Neo4j Graph algorithms (plugin)[‖]: There is a plugin named 'Neo4j Graph Algorithms' which provides several extra functions. Here, we can find some useful ones defined inside 'Community Detection algorithms', also known as 'Clustering algorithms' or 'Partitioning Algorithms'.
- Ineo[**]: An instance manager, aimed at providing several Neo4j instances at a time, each one at a different port.
- CAPS: Cypher for Apache Spark[††]. CAPS extends Apache Spark with Cypher. It allows for the integration of many data sources and supports multiple graph querying. It enables you to use your Spark cluster to run analytical graph queries. Queries can also return graphs to create processing pipelines.

3.3.4.2 Physical partitions

Neo4j is designed to run a single graph in each server, which means that, for a physical separation, each partition will need a custom server. A virtualised approach can be obtained using multiple server instances with different configuration files and base directories in each of them.

Another option is the use of Docker containers to handle the isolation and administration of resources.

[‖] https://neo4j.com/developer/graph-algorithms/
[**] https://github.com/cohesivestack/ineo
[††] https://github.com/conker84/cypher-for-apache-spark

3.3.5 Application use cases

Some applications of using a UMS-based LDM are described in the following subsections.

3.3.5.1 Risk assessment

A classic objective of a LDM system is to feed a risk detection module, in charge of estimating the most probable paths and maneuvers of other vehicles and pedestrians nearby, in order to assess a risk value for each of them. This needs to be fed by the sensor of the ego-vehicle itself, other cooperative vehicles or ITS stations. Using a UMS-based enhanced LDM can be of assistance in estimating risks under periodic pattern situations when traversing a complex road network area. To illustrate, we can consider the situation of a vehicle approaching a bus stop at a time of day when it is usually crowded with people. Even though an ego-vehicle is not able to observe situation in advance, historical physical and non-physical resource situation information owned by the UMS, as well as predictions, can notify the vehicle that special caution is needed.

3.3.5.2 Route optimisation

Knowing the transport network situation under periodic patterns as well as estimations about future status can be of assistance to enable the LDM to suggest better route alternatives. This is what navigation solutions are used for, but the benefit of using the integrated urban network representation model helps increase the types of data sources that feed the graph.

3.4 Summary

In this chapter we have described the main framework to explain what kind of transportation resources exist in a city, how they are usually handled by city mobility managers, and for what purpose. Currently, digital management of these assets from an integrated multimodal view is limited. Therefore, we have proposed a new concept called Urban Movement Space, which integrates the most well known frequent city passenger transport modes in order to consider the main concerns when digitally representing all that information. With this description complete, we are ready to head towards the next chapter, where the focus shifts from the representation of the mobility resources, to the representation of the everyday mobility data at its multiple variants. All these mobility data, in order to be analysed properly, will need to be framed within the spatial and temporal context of the urban movement space.

Mapping movement data over the transportation resources

4

In Chapter 3, we focused on our proposal for a digital representation of the multimodal resources that cities provide for mobility purposes, the Urban Movement Space (UMS). Now, we will focus on the kinds of mobility data collected, and how their representation can be related to the proposed digital representation for resources or assets.

This chapter is divided into three sections. In Section 4.1, we provide a categorised description of the main types of daily operation information sources found when handling urban mobility management, including the most significant challenges and concerns about them. Understanding the variety of ways data can be received is key in order to be able to combine and map them onto the UMS representation. The step of mapping the data into the transportation resource representation is needed so as to propose valid KPIs, advanced analytical or short-term prediction solutions, and thus, we will also introduce the concept of map-matching. In Section 4.2, we provide an experimental analysis of the map-matching process according to the features of the road network representation that are encoded in the UMS. For this analysis, we propose a set of quantitative metrics that are able to characterise the structure of the road network in cities, and evaluate their relation with the performance of map-matching algorithms. Finally, in Section 4.3, we summarise the main contributions.

4.1 Generalising a movement data representation model

Currently, there is a wide range of mobility data sensing capabilities. Movements across the city on multiple transportation modes can be recorded by sensors deployed on the physical infrastructure of the transportation resources, or by sensors on each moving citizen or vehicle. The type of information collected by these individual sensors, or by more complex information systems, is diverse. In the following subsections, we describe the diversity of these alternative collections of movement data in cities, attempt to categorise these alternatives and generalise how this input data can be classified.

4.1.1 Modelling input data

Input data is received from multiple locations across the city, at different frequencies (periodic sensor providers) and event processes.

A thorough characterisation of the types of available data is necessary to know what kind of processing we are able to do with them and to what extent we are able to combine the data to measure city mobility from an integrated view. Therefore, we describe a categorised classification of types of data sources, their main features and challenges, according to:

- Observation type: *Integration period* or *Event*.
- Sensor mobility: *Fixed* or *Mobile*.
- Type of detection: *Presence (binary)*, *Measure (numeric)*, *Count (numeric)*, *Location (spatial geometry)*, *Track (temporal sequence of spatial geometries)*
- Type of identification: *Unknown*, *Identification*, *Classify*
- Directionality: *Non-Directional*, *Directional*
- Influence: *Coverage (geometry)*, *No-Coverage*

With this generalised sensor model classification we are able cover the following types of mobility data sources: traffic loop detectors, computer vision traffic sensors such as vehicle classification or identification based on label-plate recognition (LPR), parking data (begin and end records, or a begin record with an estimated duration, as well as occupancy status), pedestrian flow counters, Wi-Fi hot-spot connections, GNSS traces coming from any of the transport modes, LBS check-in data, smart-card records at public transport facilities (pick-up and drop-off), and bike-sharing data (pick-up and returns). In this regard, Table 4.1 collates and classifies different types of mobility data sources that are usually used in city transportation management.

We would like to point out that the sensor technology and the data provided by the sensor/information system are not equivalent. For instance, even though a sensor's basic technology may be based on the detection of events of a particular physical magnitude, the sensor system in charge of providing the final data may produce aggregated counts (e.g., inductive traffic loop counters). An extended discussion on this, out of the scope of this thesis, could be started about more and more additional capabilities of computation and intelligence that can be deployed on sensor systems, sometimes referred to as *on edge* (for example, advanced computer vision). Additional capabilities can be inferred, as well, by the analysis of time series of the sensor systems' raw data or by a joint analysis of data coming from multiple sensors and centralised by complex information systems. Therefore, the boundaries of where and how mobility data can be obtained are diffuse.

Sensor type	Observation type		Sensor mobility		Type of detection					Type of identification			Directionality		Influence	
	Int. Period	Event	Fixed	Mobile	Presence	Measure	Count	Location	Track	Unknown	Identify	Classify	Directional	Non-Direct.	Coverage	Non-Cover.
Traffic loop detector	✓		✓				✓			✓				✓		✓
Camera (LPR)		✓	✓		✓						✓		✓			✓
Camera (vehicle count by class)	✓		✓				✓					✓	✓		✓	
On-board CV camera/LiDAR [a]		✓		✓	✓				✓			✓		✓	✓	
GNSS positioning		✓		✓				✓			✓			✓		✓
Parking occupancy		✓	✓			✓				✓				✓	✓	
Pedestrian counters	✓		✓				✓			✓			✓			✓
Wi-Fi hot-spot connections		✓	✓		✓						✓			✓	✓	
LBS check-in data [a]		✓	✓					✓			✓			✓		✓
Transit smart-card records		✓	✓		✓						✓			✓		✓
Bike-sharing pick-up/returns		✓	✓		✓						✓			✓		✓

Table 4.1: Classification of mobility data source types

[a] Not often used by council mobility managers

4.1.1.1 Observation type

When observing any movement-related data, we distinguish two different alternatives. This category describes whether the data needs to be recorded during a given time period (integration period) or not (event). For instance, traffic loop detectors provide count data on a periodic basis, with integration intervals of one minute, for example, and they provide the number of vehicles that pass across a traffic section during that interval. Other detectors, such as label plate recognition systems (LPR), provide the record of an identified vehicle instantaneously. Note that the difference is not the periodicity of the data: periodic event-based measurements can be received if the sensor or information system relies on periodic checks of the situation that is being monitored, based on a sampling rate (e.g. occupancy of a parking lot).

When integration periods are involved, the metrics obtained depend on the duration of these periods. If different sensors or information systems provide measurements under different integration periods, their values cannot be compared or aggregated, unless a previous transformation is done. The same problem exists if the integration

period of a given sensor varies in time.

4.1.1.2 Sensor mobility

Another category is the mobility of the sensor. Some movement data providing sensors are located at fixed locations while others are mobile. The information about the location, in both cases, is key when mapping the data recorded to the transportation network. The location required, in fact, may not be satisfied by having a pair of geo-coordinates, because each location needs to be associated with a unique directed network edge. However, it may not be possible, depending on the structure of the network, for example, when the road network is dense or streets are two-way.

If sensors are fixed, having first-time inventory data with a suitable set of relations against the network is sufficient, albeit difficult to automate in most cases.

Otherwise for moving sensors, each piece of data obtained, together with its real-time position, needs to be mapped using techniques such as map-matching. We will go deeper into map-matching later in this chapter. An exception to this is when the software system providing the data is able to handle and post-process additional information. For instance, when passengers get on a bus and their smart-card is recorded, the sensor located on the bus is mobile. However, the system is able to assign that passenger pickup to a given transit route, its direction and associated stop.

4.1.1.3 Type of detection

This category divides the type of information that is obtained. *Presence* only gives the information about the detection of any type of element without any additional information. *Measure* type of detection provides a numerical magnitude of any kind (such as speed or size). *Count* is unitless, giving an integer number of how many elements have been detected during an integration period or in an event (for instance when an image is processed). *Location* type of detection represents a spatial position of the detected element over a reference system, which can be extended to *Track* when this information is obtained during an extended interval in time.

4.1.1.4 Type of identification

Some sensors provide information that allows identifying the moving element to some extent. Others, are able to classify the moving element. Identification may persist for each unique sensor (computer vision based cameras that are able to track elements across video frames), or across sensors (e.g. licence plate, smart-card identifier). A significant concern is that the uniqueness of the identifications may or

may not remain forever. This is the case when data are pseudonimised, and identifiers are hashed and renewed periodically. This reduces the chances of representing the repetition of travel patterns of individuals, for instance.

Other sensors are not able to identify but are able to classify from a set of options. For example, they are able to distinguish a person from a vehicle, or a bike from a car.

4.1.1.5 Directionality

The directionality provided by some sensors gives information about the sense of movement of an observed entity. Some simple cases, are specifying the direction in a two-way street, or specifying if someone enters or exists from a given area, but it can be generalised as an angle based on a reference axis or set of axes.

4.1.1.6 Influence

This category consists of different options in which a sensor can observe the movement happening in its surroundings. In some sensors the area in which data can be observed, or the distance within which these data are measurable, is relevant and influences the measurements. For example, video or thermal cameras are only capable of processing what happens inside its field of view and the area of interest configured for further analysis. Similarly, a Wi-Fi network hotspot or Bluetooth beacon used to estimate the number of connected citizens is clearly dependent on the signal coverage.

4.1.2 Data acquisition techniques

The technical procedures in which movement data can be obtained from the original sources are diverse. Data can be provided by a centralised information system with a unique logical access point or by individual sensors remotely located (IoT). The principle means of accessing data that we have had to manage includes the following:

- File exchanges
- Database access
- API-based access
- Asynchronous messaging

In this regard, a relevant aspect is if data are received in a timely manner (periodically or at events) or if they need to be requested on demand. We use the term *cadence* to refer to the delay between the moment a movement record is observed and the moment this record is available for use. Additional delays may occur if any other third party system needs to request the data on demand. This is the case when data acquisition relies on file exchanges, API calls and database queries, and thus, for

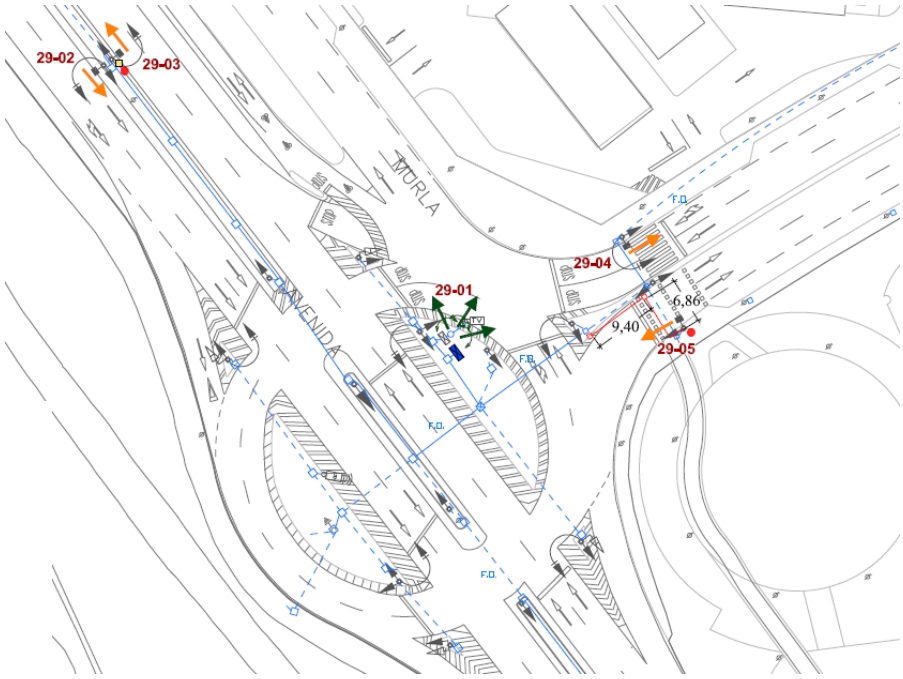


Figure 4.1: Sample PDF file with sensor locations. Context camera (29-01) and LPR cameras (29-02, 29-03, 29-04, 29-05). *Source: Alicante Se Mueve* project [a]

[a] https://www.alicante.es/es/area-tematica/alicante-se-mueve

real-time processing, it is necessary to know how and when new data is expected to be available in order to adjust the timing of a request. In applications where real time is not important, the implications of cadence are reduced. Some examples of relatively high cadence are sensor systems that bulk data periodically to a central server. We have seen this happens, for instance, with public transport travel records in some cities, since they are downloaded from each bus ticketing system at the end of the day.

Regardless of the acquisition technique, a list of the sensors involved and their locations (if fixed) is needed in order to associate data to the transportation network representation. In Figure 4.1 an example of five sensors and their exact location over the road network is depicted. However, since data is provided in PDF files, building automatic processes is not possible. Some information systems provide suitable tools and resources to acquire and maintain an up-to-date list of sensor codes and their locations through API methods, inventory files and so on. In other cases, a manual inventory of the sensing equipment must be built from scratch. A lack of a detailed digital inventory of sensors and their location to map them onto the transportation network is a generalised limitation in many cities.

4.1.3 Uncertainty sources

All the data that we are prepared to collect from the different data sources may provide several degrees of uncertainty; while this is not a unique problem of mobility-related data, it is important to understand what we face in order to assess the quality of the outcomes that we may obtain from further analysis that rely on uncertain data.

Uncertainty of movement data has several potential sources and it is difficult to manage them efficiently. In our representation model, we assume the following main sources: uncertainty in space, uncertainty in time, uncertainty in the availability of the data and uncertainty in the quality of the data. Under any form of these uncertainty types, notice should be given if direct action (such as applying a map-matching algorithm as presented later) is not taken.

- Uncertainty in space is very clear in mobile input types: positioning error (GNSS, wireless network). This error estimation needs to be reported by the system when it tries to match the location, or sequence of locations, to the network using a map-matching algorithm. We consider it appropriate to use confidence level ranks to discard data that does not satisfy a given threshold.
- Uncertainty in time may occur when data records are not tied to a corresponding timestamp in origin. In such cases, the third-party system that uses the data needs to attach the temporal information about when the records have been retrieved. This is the case of systems that provide the most recent record. Any communication failure or delay in requesting the information will lead to an error with the time. Similarly, moving sensors providing individual tracked data, for instance, are more prone to communication failures. Additionally, the main bias in time is due to temporal granularity which, according to each application, measures may need to be processed after observation over an integration period of several minutes (e.g. traffic count sensors).
- Uncertainty in the availability of the data: even though data acquisition is done in real time with a small time granularity, the final data processing platform may not be able to process it given that the service provider offers the data on a daily basis (e.g. at the end of the day, as previously mentioned). This cadence is usually fixed and known in advance. The availability of consolidated data for real-time operation cannot be assumed in this case, and only corrections *a posteriori* can be made.
- Uncertainty in the quality of the data: in this category, all the errors in the measurements of non-spatial attributes are included, from sensor resolution and accuracy (given by the sensor), to the lack of reliability of the data due to low penetration of the sensors in the city coverage.

4.1.4 Map-matching

We have already mentioned the concept of map-matching, required when exploiting data from mobile sensors that we need to associate with a digital map. In reality, with any fixed sensor, as well, we need to match its location to the digital representation of the transportation network, discussed in Chapter 3, at least once.

4.1.4.1 Matching fixed sensors

During the research projects leading to this thesis, most map-matching processes with the fixed sensors were carried out semi-automatically, due to the diversity (and sometimes lack) of inventory data that prevents fully automatised solutions. Fixed sensors need to be map-matched only once, as long as their location is permanent. Geometry-based *point-to-curve* map-matching methods (described below) are the only ones that are applicable for unique location points such as those coming from fixed sensor inventories. All the other map-matching techniques rely on sequences of locations points focused for mobile sensors.

However, geometric algorithms are not always sufficient due to the location not being specific enough to determine the edge of interest, and that is when manual work is needed. In some cases, two different edges of a two-way street share the same edge geometry or the sensor is located very close to another edge, and can be incorrectly associated. See Figure 4.2, where edges are not distinguishable from their spatial information, and therefore the LPR sensors cannot be automatically associated to the correct one. Some scenarios in which we are able to overcome this problem, from the best to the worst case scenarios are:

- The inventory of the sensor, together with the location coordinates, provides the unique id of the edge id_{e_i}.
- The inventory of the sensor, together with the location coordinates, provides the unique ids of the *from* and *to* nodes id_{v_j}
- The inventory of the sensor, together with the location coordinates, provides the location coordinates pairs of the *from* and *to* nodes id_{v_j}
- The inventory of the sensor, together with the location coordinates, provides some kind of textual description that could somehow be encoded (for instance, name of the street and direction, such as "Exit" or "Entry")

4.1.4.2 Matching mobile sensors

In general, the concept of map-matching is mostly used for sequences of mobile sensor locations in applications usually known as Floating Car Data (FCD), which act as moving sensors embedded in cars. In fact, map-matching arises as the first problem that needs to be solved, when the data obtained consists of geolocated

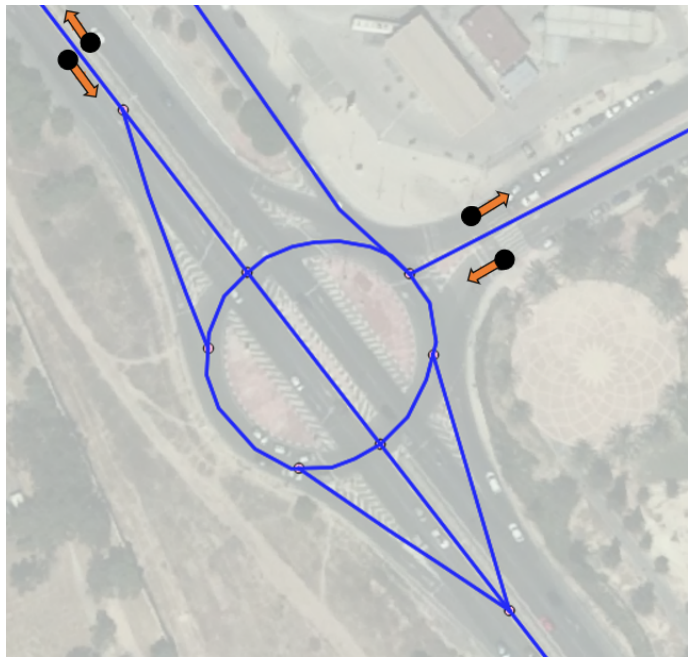


Figure 4.2: Examples of cases where geometric map-matching is not enough to associate fixed sensors to road network. Two different edges of a two-way street share the same edge geometry (blue lines). Black dots represent the spatial location of LPR sensors, and orange arrow represent their orientation

positions of moving elements. There is always an error of some meters between the geolocation given by positioning devices and the real position, so map-matching techniques are applied to overcome this imprecision.

The map-matching process usually consists of identifying the road link and defining the position of the object on this link. The result can be used to determine the physical location of the object and to improve the accuracy of positioning by correcting positioning errors. As mentioned, in most map-matching cases, positions are not analysed in isolation, but as part of a sequence of points that describe a path, which allows relation assumptions to be made between points. The map-matching process, in general, relies on a set of digital maps that represent the road or street network (with geographical and topological information) as a weighted directed graph, like the Urban Movement Space proposal we previously described. *Weighted* means that travelling along some edges is more expensive than along others, in terms of time or distance. *Directed* is used when the cost for one edge may be different travelling from source node to target node or the other way round, like in one-way roads. Roads are represented as edges with their shapes (a polyline describing the central line). Nodes are used to represent start and end points of edges, as well as junctions. Topology information is used to determine whether a pair of edges is connected or not. Each kind of map-matching technique uses this information at different levels, and some

techniques do not use it at all.

Several studies can be found in the literature, describing and evaluating different algorithms for map-matching in its multiple variants. Four main groups are found in the literature which try to solve the map-matching problem with different approaches:

- Geometry-based techniques that work with the shape of the links between nodes. The most representative examples are *point-to-point*, *point-to-curve* and *curve-to-curve* algorithms ([99, 100]).
- Topology-based algorithms which rely on vector-based street maps and take the relationship between entities into account. The use of topological information about the road networks means taking advantage of connectivity and contiguity constraints that are encoded in vector maps. This information allows the number of candidate arcs for each sample point to be reduced, and then using a weighting system to measure similarities between path portions and candidate arcs. These algorithms are usually fast and relatively easy to implement, however, performance might differ in terms of real-time capability and robustness. Among topology-based map-matching variants, the use of heading information is quite frequent, as well as making connections to previous or next sample points [101–108].
- Statistical or probabilistic techniques, which are based on confidence regions around observations. Many statistical techniques such as Kalman Filters and Hidden Markov Models (HMM) are used [109–112].
- Advanced map-matching algorithms that use more refined concepts. Kalman Filter and Extended Kalman Filter, Dempster-Shafer's mathematical theory of evidence, particle filters, fuzzy logic or applications of Bayesian inference should be mentioned [113]. More concrete examples are recursive Bayesian estimation methods [114] and the use of Bayesian Belief Networks [115].

Other groups classify map-matching algorithms into simple, weight-based and advanced map-matching algorithms [116]. The authors argue that the classic categorisation is not suitable to differentiate among recent algorithms because geometrical and topological information are used in most of them. Additionally, it must be noted that the same algorithm can sometimes be classified as advanced and probabilistic.

To understand the different variants of the map-matching problem, characteristics of the input data must be taken into consideration:

- Sampling frequency: sampling rate refers to how often a new observation is registered. The problems that arise when facing map-matching at 1 Hz sampling rate (quite frequent in navigation applications), are not the same as those faced at lower frequencies (periods up to 5 minutes, ~0.002 Hz). Low frequency FCD is aimed at being energy-efficient, but it suffers from lack of observed positions for many of the road segments traversed along the route.

- Available information: estimated location as a pair of coordinates is the minimum data required in each sample. Speed and heading information are available in many of the map-matching problems. According to the geolocation method and internal sensors used, other inertial information coming from accelerometers might be used, mainly when navigation is pursued. In this sense, it is quite frequent to use Deduced Reckoning ("Dead" reckoning or DR) sensors which consist of an odometer and a gyroscope, integrated with the GNSS sensor.
- On-line or off-line map-matching: in some cases, map-matching in post-processing or in batch is useful enough, but many applications need the results in real time.
- Localisation technique: the use of Wi-Fi and classic network-based localisation technologies offers less accurate positioning, but they allow unnecessary energy consumption to be avoided. Although in most cases GNSS offers more accurate location information than Wi-Fi and network-based localisation, the superiority of GNSS may decrease in some dense urban areas (GNSS may have significant outliers due to multipath effects). Additionally, future ultra-dense 5G networks will also improve positioning of moving nodes.
- Environment: the characteristics of the geographical area can influence the precision of positioning (unwanted GNSS signal reflections and shadows due to high buildings in urban areas, or tunnels, for example), and the road configuration may affect in the accuracy and speed of the map-matching technique.

4.1.4.3 Grid-based matching for localised data

We consider it relevant to describe a different map-matching approach, which is especially valid in small localised areas where the spatial granularity of the UMS is not sufficient. The local area of interest is discretised as a cartesian grid C, so that sensor observations and measurements can be spatially associated to smaller grid-cells. The following variables represent the local area in a 2-dimensional Euclidean Space:

- L_X: width (in metres) of each cell
- L_Y: height (in metres) of each cell
- N_X: number of cells in X axis
- N_Y: number of cells in Y axis

The local area, which covers $(N_X L_X) \times (N_Y L_Y)$ m^2, can be represented as a $N_X \times N_Y$ matrix of cells. Each cell can then be identified by a pair (x, y), where $x = 1, 2, ...N_X$ and $y = 1, 2, ...N_Y$. Each observation (e.g. vehicle positioning) is matched to a single cell at each snapshot.

In addition, we propose the concept of *neighbour-order* (Π) to represent the relationship between adjacent cells. The neighbour condition can be measured by the Neighbour-order parameter. A pair (x', y'), where $x' = 1, 2, ...N_X$ and $y' = 1, 2, ...N_Y$, is said a Π-order neighbour of pair (x, y), if and only if the following two conditions are met:

$$\begin{aligned} |x' - x| &\leq \Pi \\ |y' - y| &\leq \Pi \end{aligned} \tag{4.1}$$

When designing a grid-like subdivision of urban areas, we would also like to consider some other alternatives. For instance, we open the possibility of circular (radial) or hexagonal patterns instead of square grids. Nevertheless, matching positions onto these alternative representations is computationally more complex.

4.1.5 Main KPIs derived from movement data

After studying the principal aspects to consider when mapping of movement data onto the digital network, we now describe the main numerical values that we will be able to generate. In the following list we compile a collection of the most basic set of quantitative indicator types when describing mobility in cities.

- Volume: Number of entities observed during a given period. When extrapolated to a known fixed period length (such as one hour), this metric is known as flow.
- Speed: Relation between the time needed to traverse a finite segment and its known length (like a road segment). Instantaneous speed brings that relation to the limit of an infinitesimal segment length.
- Occupancy: Percentage of used slots or spaces from a given total number in a given time instant. For instance, parking occupancy represents the number of parking spaces that are being used at a given moment against the total number of spaces. Bike-sharing station occupancy means the number of slots that are occupied by bikes against the total number of spaces. The percentage of seats that are being used by passengers on a bus or train is equivalent as well. However, in traffic engineering, the occupancy is also defined as the percentage of time a point on the road is occupied by vehicles.
- Density: number of entities per unit of distance, almost exclusively used for road vehicles.
- Travel time: time needed to go from point A to point B. These points can be the origin node and destination node from a link, thus travel time represents the link travel time. However, the points can be located at a greater distance, the travel time representing the temporal length of a journey.

These indicators are basic observations. Multiple variants of greater complexity can be built on top of them when aggregated in space and time.

4.2 A quantitative analysis of the transportation network configuration on map-matching algorithms

In the previous section we introduced the concept of map-matching, as an unavoidable process needed to associate moving location data points over the digital transportation network representation.

Now, in this section, we aim to present a quantitative method we have developed to analyse the impact of map-matching algorithms' accuracy, especially in cities, according to a set of numerical attributes that characterise the road transportation network representation that we have already built. We will focus on the case of road transport networks for vehicles in FCD scenarios with low sampling rates.

The analysis is focused on the assumption that road configuration in cities is more complex. The implications of the worse quality of the GNSS signal, inherent in urban environments, have not been addressed. Some complexities of urban map-matching have been previously covered in the literature, e.g., the existence of multilevel streets [117]. However, our work presents an evaluation of which are the main parameters that make urban road configuration more complex for map-matching purposes. In the present study, three matching algorithms have been tested: *Point-to-Curve (PC)* [99], *ST-matching (ST)* [104] and *Hidden Markov Models (HMM)* [118].

Simulated vehicle data under five different real environments have been evaluated, in order to answer the following questions:

1. Is there any correlation among the matching accuracy and the road configuration parameters?
2. Which environment/road configurations most affect accuracy in each algorithm?
3. How does each algorithm behave when the sampling period is higher, that is, when observations are more sparse in time?

Due to urban planning and building density in cities, road configuration differs from that found in suburban/interurban areas and links between cities. The study published by Dowling [119] described a classification of different features or road configuration (see Table 4.2, columns 1-4). Map-matching algorithms do not achieve the same performance in every type of scenario, and therefore, we added a new column here with the qualitative impact expected in map-matching (Table 4.2, last column). Throughout the rest of the section, we will develop the quantitative analysis that supports these expectations.

Map-matching algorithms for road traffic in urban scenarios have several limitations for following reasons:

- Poor GNSS signal because of occlusions made by buildings and multipath reflections.

Criterion	Suburban	Intermediate	Urban	Impact on map-matching
Driveway/ access density	Low density	Moderate density	High density	Moderate - affects the number of possible candidates
Arterial type	Multi-lane divided; undivided or two-lane with shoulders	Multi-lane divided or undivided; one-way two-lane	Undivided one-way, two-way, two or more lanes	Moderate - affects the difference between the real path and the digitalised edge geometry
Parking	No	Some	Significant	Moderate - affects the existence of possible circular routes and revisited road links
Separate left-turn lanes	Yes	Usually	Some	Low - affects the difference between the real path and the digitalised edge geometry
Signals/ mile	1-5	4-10	6-12	Low - affects the continuity of the traffic flow
Speed limit	40-45 mph	30-40 mph	25-35 mph	High - affects the variability between nominal speed and real travel speed
Pedestrian activity	Little	Some	Usually	Low - affects the continuity of the traffic flow, thus, the variability between nominal speed and real travel speed
Roadside development	Low to medium density	Medium to moderate density	High density	Moderate - affects the positioning accuracy

Table 4.2: Features characterising road configuration (columns 1-4) [119], and expected impact on map-matching accuracy (column 5).

- Traffic flow is less continuous due to junctions, pedestrian crossings and traffic lights.
- Road density is higher, streets are closer to each other and even multilevel streets may be found.
- The correlation between vehicle speeds and nominative speed limits is smaller than the one in interurban roads.
- Cruising for parking does not follow shortest path assumption.

In addition, when the frequency of position measurements is low, the path inference becomes even more complex than at higher frequencies due to lesser input data for spatio-temporal correlations. Nevertheless, this is a generic drawback in both urban and non-urban areas.

We follow the next methodological steps to perform the proposed comparative evaluation of map-matching performance:

- Definition of grid-based road configuration metrics
- Generation of sample FCD observation dataset
- Evaluation of map-matching algorithms

4.2.1 Preparation of the Urban Movement Space

One of the first tasks is to prepare the Urban Movement Space. We will rely on the Eulerian approach and for this analysis, we only need the road network. A PostgreSQL database has been set up with PostGIS 2 extension (for spatial capabilities). Crowdsourced open digital maps from OpenStreetMap [120] is used as the road data source. The information extracted must be represented as a graph. To build the road network, only OSM ways that allow cars must be chosen. Each original OSM way is, in general, represented by several smaller network edges after the map adaptation process. How the map data is converted depends on the implementation and, thus, special attention must be paid to the identifiers of the edges when trying to refer them to original OSM way identifiers: the same OSM identifier can be represented by several network edge identifiers. The conversion is performed using the osm2po* tool, with the configuration shown in Table 4.3.

The final data structure obtained is the one already explained for the street network in subsection 3.2.1. In this study, we discard any time-variability in the structure of the network and we use a unique version. In addition, it must be noted that we evaluate and compare map-matching algorithms over urban and non-urban scenarios, and therefore, the area included is bigger than a city.

* https://osm2po.de/

OSM highway tag	Nominative speed, Km/h
motorway	120
motorway_link	30
trunk	90
trunk_link	30
primary	70
primary_link	30
secondary	60
secondary_link	30
tertiary	40
tertiary_link	20
residential	40
road	50
unclassified	50

Table 4.3: OSM ways selected and the nominative speed assigned to each type of road way classification

4.2.2 Definition of grid-based road configuration metrics

A grid-like structure, C, has been created for the spatial aggregation of the road network features. The map is divided into $(10^{-3})^\circ$ latitude x $(10^{-3})^\circ$ longitude grid-cells, in WGS84 ellipsoid, with an area of ~9000 m^2. This size has been chosen arbitrarily, but it is useful enough for comparison purposes among different environments.

It should be noted that only cells where at least one road edge exists are taken into consideration in set C. Each cell $C_j \in C$ is represented by a spatial polygon geometry $geom_{C_j}$. The intersection between an edge e_i and a grid-cell C_j can be defined as follows:

$$e_i \cap C_j = geom_{e_i} \cap geom_{C_j} \tag{4.2}$$

The result of the intersection can be generalised as a subset of e_i^p portions ($p = 1, 2, ...p_{i,j}$). The corresponding spatial linestring geometry of each portion is contained in the original geometry: $geom_{e_i}^p \subseteq geom_{e_i}$, and thus, the length is $l_{e_i}^p \leq l_{e_i}$.

Four metrics are calculated in each cell C_j:

Edge count (N): the number of different road edges that intersect (total or partially) with the cell polygon.

$$N_j = \sum_{i=1}^{n} x_i^j \tag{4.3}$$

where

$$x_i^j = \begin{cases} 0, & \text{if } e_i \cap C_j = \emptyset \\ 1, & \text{if } e_i \cap C_j \neq \emptyset \end{cases} \tag{4.4}$$

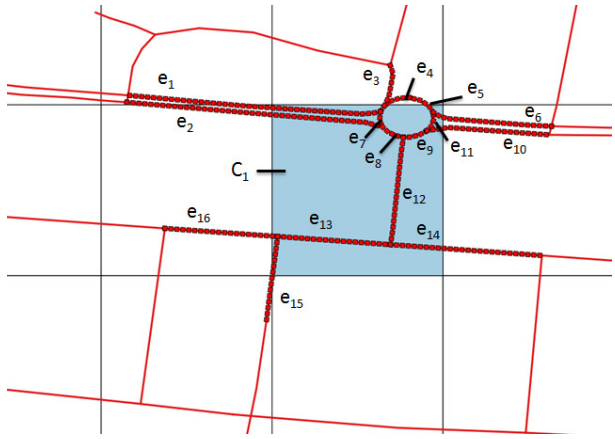


Figure 4.3: Representation of the road network in an urban area. Edges e_1 to e_{16} intersect (total or partially) with the C_1 cell polygon (darker).

all cells intersect, at least, with one edge, thus

$$\sum_{i=1}^{n} x_i^j \geq 1 \tag{4.5}$$

Average edge-length (L): average length of the edges that intersect (total or partially) with the cell polygon.

$$L_j = \frac{\sum_{i=1}^{n} x_i^j l_{e_i}}{\sum_{i=1}^{n} x_i^j} \tag{4.6}$$

Average nominative speed limit (V): average nominative speed of edges that intersect (total or partially) with the cell polygon, where v_{e_i} represents the nominative speed limit in edge e_i.

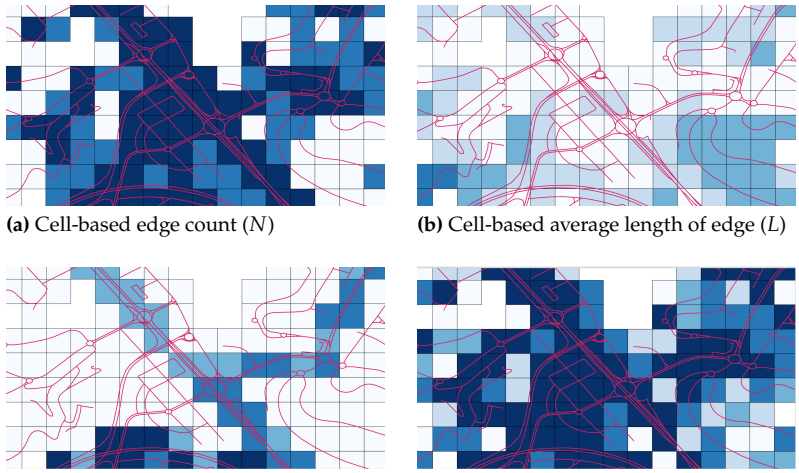
$$V_j = \frac{\sum_{i=1}^{n} x_i^j v_{e_i}}{\sum_{i=1}^{n} x_i^j} \tag{4.7}$$

Road density (R): expressed as length per unit area, this is, meters of linear road per cell polygon area in square meters.

$$R_j = \frac{\sum_{i=1}^{n} \sum_{p=1}^{p_{i,j}} l_{e_i}^p}{A} \tag{4.8}$$

Figure 4.3 represents a zoomed in sample grid cell and the road network edges that are involved when calculating these grid-based measurements. In Figure 4.4, cells are coloured according to the values for the four metrics.

To understand similarities among the road environments found in the tested scenarios, a Principal Component Analysis (PCA) [121] is proposed in this step. This allows to



(a) Cell-based edge count (N)

(b) Cell-based average length of edge (L)

(c) Cell-based average nominative speed limit (V)

(d) Cell-based road density (R)

Figure 4.4: Visual representation of the four parameters in each grid cell. Darker colours represent higher values in a five-category quantile classification. Red lines describe the geometry shape of the road edges.

the dimensionality of the data to be reduced from the original four variables to two and represent the scenarios in a 2D chart.

4.2.3 Generation of sample FCD observation datasets

SUMO traffic simulator [122] has been used to generate a synthetic dataset of significant size, which contains a wide set of track points generated in several settings. The area under analysis has been separated into five different known environment scenarios (s): three different city centres (s_1, s_2, s_3) and two mixed environments connecting two and three towns, respectively, including interurban roads (s_4, s_5). In each of the five scenarios, random car trips have been generated over one-hour simulations, following uniform coverages of the scenarios. Simulations have been configured to generate a FCD observation every second per trip (sample period $\tau = 1$).

A trip $t^m \in T_s$ (set of tracks in scenario s) is represented as a sequence of z FCD observations $\{o_1^m, o_2^m, ...o_z^m\}$. The ground-truth of each simulated trip t^m is represented as a sequence of ω distinct original identifiers of OSM ways, $GT_{t^m} = \{osm\_id_1^m, osm\_id_2^m, ...osm\_id_\omega^m\}$ where $\omega \leq z$. Please, note that the same osm_id code may be represented by many e road edges. The reason for using a sequence of edges instead of the original simulated sequence of track points is to allow the method for real tracks to be generalised since, in general, what is known is the original route travelled, not the real positions.

A Gaussian random error of standard deviation of $\sigma = 20$ metres has been added to every observation position coordinate, which is quite pessimistic in comparison with reality. However, since signal error is out of the scope of this evaluation, it adds the same uncertainty to every scenario regarding satellite signal quality, and allows focusing only on the influence of road configuration parameters.

4.2.4 Evaluation of map-matching accuracy

The sampling period used in the dataset generation is $\tau = 1$, but observations are sub-sampled in multiples of 5 to evaluate the performance of the matching algorithms on the same tracks under different low frequency conditions. Concretely, from $\tau = 5$ (skip 4 observations out of 5) to $\tau = 60$ (skip 59 observations out of 60). The objective of the map-matching is to associate observations to the most suitable set of road edges defined in a given digital vector map. Therefore, the result of the map-matching of a t^m trip at sampling period τ ($\{o_1^m, o_{1+\tau}^m, o_{1+2\tau}^m, ...\}$) is a set of calculated edges $MM(t^m) \rightarrow \{e_1^m, e_2^m, ...e_\phi^m\}$ where $\phi \leq \lfloor z/\tau \rfloor$. The same edge may appear more than once if τ is long enough to generate several sequential observations along the same road link.

A calculated edge is marked as incorrect if that edge does not exist in the original route. This must be understood as a False Positive detection. Mismatched points will not be marked as incorrect as long as they are matched to a road segment that is part of the route. This would not be strictly correct if the vehicle was actually traversing another segment, but this kind of mismatched errors cannot be detected, since the ground-truth reference is based on route edges.

We have defined three main quantitative metrics for matching performance: single observation-based edge accuracy and its aggregated values at cell and track levels.

Observation-based edge accuracy ($AO_{e_k^m}^m$) represents whether the calculated edge for a given observation o_k^m, where $k = 1, (1+\tau), ...(1+\lfloor z/\tau \rfloor)$, belongs to the original ground-truth (GT) route of trip t^m or not.

$$AO_{e_k^m} = \begin{cases} 1, & \text{if } osm\_id_{e_k}^m \in GT_{t^m} \\ 0, & \text{otherwise} \end{cases} \tag{4.9}$$

Track accuracy (AT_{t^m}) represents the average value of the observation-based edge accuracies of all the map-matched edges from a given trip t^m:

$$AT_{t^m} = \frac{\sum_{k=1}^{\phi} AO_{e_k^m}}{\phi} \tag{4.10}$$

Each map-matched FCD observation is associated to its nearest grid-cell. Thus, observation-based accuracy (AO_{e^m}) and further aggregates can be associated with the properties of each corresponding cell as well: N, L, V and R.

Cell accuracy (AC_{C_j}) represents, for multiple μ trips, the average value of the observation-based edge accuracy of map-matched observations that fall inside a given cell C_j:

$$AC_{C_j} = \frac{\sum_{m=1}^{\mu} \sum_{k=1}^{\phi_m} x_{m,k}^{j} AO_{e_k^m}}{\sum_{m=1}^{\mu} \sum_{k=1}^{\phi_m} x_{m,k}^{j}} \tag{4.11}$$

where

$$x_{m,k}^{j} = \begin{cases} 0, & \text{if } o_k^m \cap C_j = \emptyset \\ 1, & \text{if } o_k^m \cap C_j \neq \emptyset \end{cases} \tag{4.12}$$

These accuracy metrics have been evaluated for three different types of matching algorithm implementations, with the objective of analysing any dissimilarities in the accuracy performance related to the road configuration metrics.

Point-to-Curve (PC): The first algorithm is a geometric technique, the Point-to-Curve map-matching algorithm [99, 100]. This technique does not make use of any relation between sequential observation pairs, since each observation is matched to the nearest projection onto an edge curve. Therefore, the sampling period does not affect matching accuracy. It is fast and simple, but accuracy is relatively low.

During the implementation, the search for the nearest candidate has been limited to a maximum distance $dist$, therefore, some extreme observations are likely to be processed without any candidate edge, and thus, missed.

ST-Matching (ST): The second algorithm analysed in this study is based on the one proposed by Lou et al. [104] (ST-Matching) for low-sampling-rate GNSS trajectories. Even though the ST-Matching algorithm has some limitations, and some other authors have suggested improvements to it (i.e. [105]), it is still a reference algorithm and the underlying base of many others. The algorithm breaks into two types of analysis, which are then combined: Spatial Analysis and Temporal Analysis. The steps are: Candidate selection, Spatial Analysis, Temporal Analysis, Spatio-Temporal Analysis and Dynamic programming based optimisation.

A custom implementation has been developed upon PostGIS and pgRouting utilities for shortest path functions. The main parameters of the algorithm are the maximum distance ($dist$), the number of candidates (c) and the window size (w). The use of windows means that partial decisions are made after processing the observations contained in a given window. It must be noted that, for the sake of computation efficiency, our implementation of the algorithm

discards observations of the same window when matching is not possible, for example, if no candidates are found.

Hidden Markov Models (HMM): The third algorithm is the HMM algorithm presented by Newson and Krumm [110], according to the implementation of Mattheis et al. [118] using OSM data. HMM are quite simple and fast. They work in real-time and are robust. In HMM-based map-matching algorithms, candidate paths are sequentially generated and evaluated on the basis of their likelihood. Emission probability and transition probability are the basis of the evaluation of each candidate. When a new observation arrives, past hypotheses are extended. Among all candidates in the last stage, the surviving path with the highest joint probability is then selected as the final solution [111].
The evaluation has been carried out using the batch mode of the algorithm. This means considering full trips and, therefore, a decision is made once information from all observations is available.

4.2.5 Results

The outcome of the simulation of trips is a dataset with a total of over 330000 FCD observations, from 1208 trips (25.5% belong to s_1, 23.3% to s_2, 22.2% to s_3, 17.0% to s_4 and 12.0% to s_5). The average trip duration is of 4:38 minutes, with a minimum of 27 seconds and a maximum of 6:40 minutes.

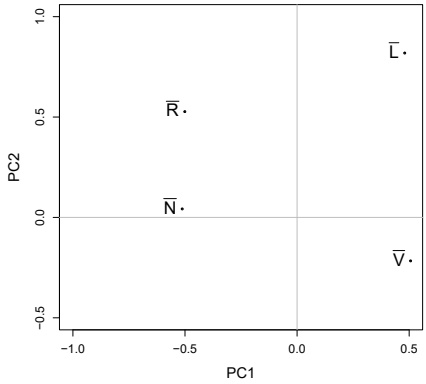
4.2.5.1 Grid-based exploratory analysis of the observations and scenarios

The variables in the five scenarios evaluated show subtle differences that differentiate urban scenarios from non-urban scenarios. Table 4.4 shows a summary of the average cell-based values of the four variables (edge count N, average edge-length L, average nominative speed limit V and road density R) in each of the five scenarios.

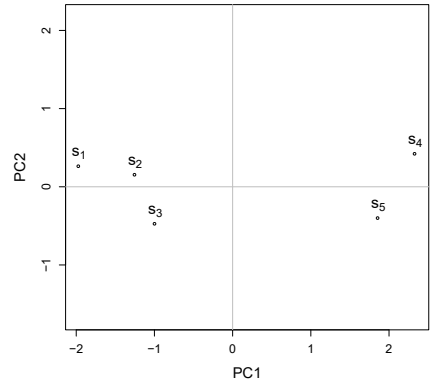
Scenario	Avg. V (km/h)	Avg. L (km)	Avg. R (m/m^2)	Avg. N
s_1	48.710	0.402	0.0182	3.606
s_2	49.759	0.450	0.0173	3.306
s_3	50.271	0.375	0.0157	3.245
s_4	53.273	0.864	0.0122	1.874
s_5	52.990	0.671	0.0116	1.963

Table 4.4: Average values of cells for all trips in each scenario.

PCA analysis has shown that the 99.8% of the variability of the scenarios can be explained by using two linearly independent variables, called principal components (and even the 95.7% of the variability can be explained using only one). As shown in Figure 4.5 (b), urban scenarios (s_1 to s_3) are clearly separated from interurban scenarios by the first component, which is constructed by very similar weights of



(a) Principal Components. The first two components are used as horizontal and vertical axes to represent the four original variables in a 2D space: average edge count ($\bar{N}$), average edge length ($\bar{L}$), average nominative speed limit ($\bar{V}$) and average road density ($\bar{R}$). The first component, which explains 95.7% of the variability, is supported equally by edge length and speed, while the contribution of road density and edge count is the opposite.



(b) Projection of the scenarios against the principal components. Urban scenarios (s_1 to s_3) can easily be discriminated from the interurban scenarios by using the first principal component. According to this, each of the four road configuration metrics on its own is descriptive enough to determine the level of "urbanness" of a set of scenarios, at least in a comparative mode.

Figure 4.5: Principal component analysis of the four road configuration variables against the five scenarios

the four variables, but positively correlated with higher road density and number of edges, as well as negatively correlated with higher length and nominative speeds.

4.2.5.2 Evaluation of algorithm performance

The three algorithms, processed over the same FCD dataset, produced the results that are now presented. The executions of the algorithms have been performed repeatedly for different τ values ranging from 5 to 60 seconds with the following fixed parameters: $dist = 20$, $c = 3$ and $w = 5$.

In Table 4.5, averages of the Track Accuracies (AT) obtained for each scenario are given for all τ values. Analysing each algorithm, accuracy is better in non-urban scenarios when using ST and PC algorithms, but this is not so evident in the case of the HMM algorithm.

A hypothesis test has been performed to evaluate whether the accuracy is negatively affected in the three map matching algorithms for urban tracks, at the multiple sampling τ periods:

$$H_0 : AT_{\{s_1,s_2,s_3\}} \geq AT_{\{s_4,s_5\}}) \tag{4.13}$$

$$H_1 : AT_{\{s_1,s_2,s_3\}} < AT_{\{s_4,s_5\}}) \tag{4.14}$$

AT	s_1	s_2	s_3	s_4	s_5
HMM	0.807	**0.835**	0.788	**0.815**	0.798
ST	0.723	0.740	0.673	**0.768**	**0.761**
PC	0.592	0.596	0.539	**0.684**	**0.682**
Global	0.707	0.723	0.666	**0.755**	**0.747**

Table 4.5: Average track accuracy, AT, in each scenario. Better global accuracy is obtained for s_4 and s_5 (two highest values in bold). However, analysing each algorithm, even though accuracy is better in non-urban scenarios when using ST and PC algorithms, this is not so evident in the case of the HMM algorithm.

We have found statistically significant evidence at $\alpha = 0.05$, to reject the Null Hypothesis H_0 and show that the mean $AT_{\{s_1,s_2,s_3\}}$ value is smaller than $AT_{\{s_4,s_5\}}$ for all $\tau < 40$ in the three matching algorithms.

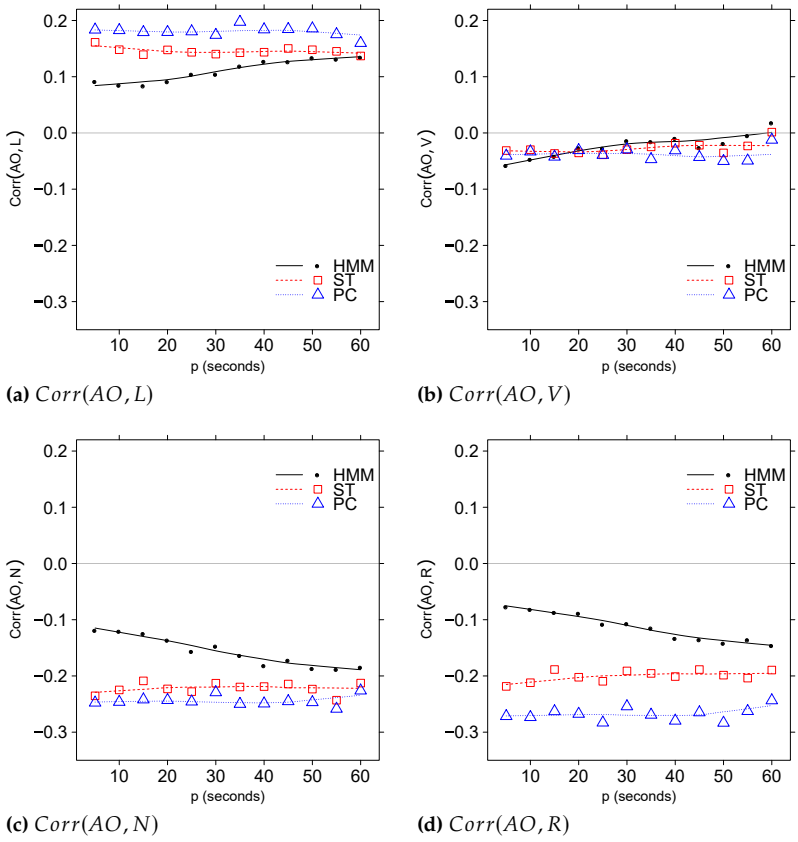
From the point of view of individual matched results, Table 4.6 summarises the correlation values among the AO and the road configuration variables. It is shown that the edge count (N) value is the metric that correlates most, negatively, with accuracy. The next is road density (R). The correlation between speed and accuracy is low.

$Corr(AO, ...)$	τ	L	V	N	R
HMM	-0.060	0.100	-0.036	**-0.139**	-0.097
ST	0.020	0.150	-0.030	**-0.225**	-0.206
PC	-0.005	0.182	-0.038	-0.245	**-0.270**
Global	-0.014	0.144	-0.033	**-0.207**	-0.197

Table 4.6: Summary of the correlation between road configuration metrics and the sampling period with the observation-based accuracy AO. The number of edges in each cell (N) is the metric that correlates most, negatively, with accuracy. In the case of the *Point-to-Curve* algorithm, road density affects the most (R), but very close to edge count. It can be noted that ST-Matching gets better values of accuracy when the sampling period τ becomes higher, since both variables are positively correlated.

Moreover, tests have allowed us to analyse whether the map-matching algorithms behave differently depending on the frequency of the observations. Figure 4.6 details the effect of sampling in these correlation values. The Point-to-Curve algorithm behaves similarly, no matter the sampling period used, as expected. What is shown is that the ST-Matching algorithm follows a relatively stable pattern as well. However, it is also shown that the HMM algorithm is more influenced by the sampling period, because the correlation associated with the HMM algorithm behaves differently according to different values of τ: the accuracy AO is more correlated to the values of L (positively), N (negatively) and R (negatively), as τ increases. AO has a slightly negative correlation with V in the three algorithms and, HMM and ST-Matching get close to zero as τ increases.

The actual AO values are represented against the full range of road configuration metrics in the following figures: Figure 4.7 for L, Figure 4.8 for V, Figure 4.9 for N



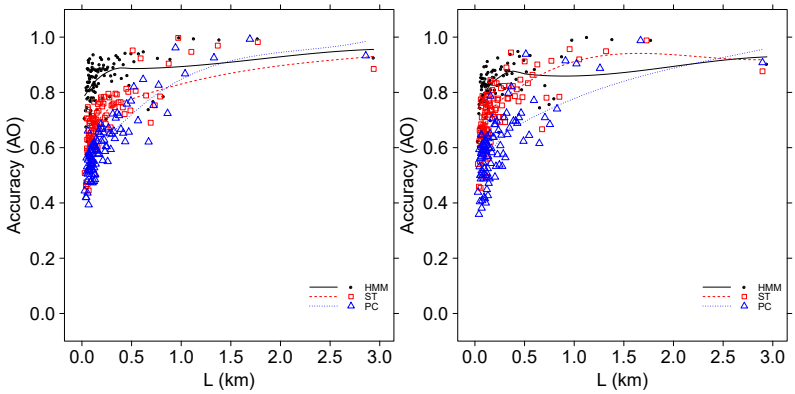
(a) $Corr(AO, L)$ **(b)** $Corr(AO, V)$

(c) $Corr(AO, N)$ **(d)** $Corr(AO, R)$

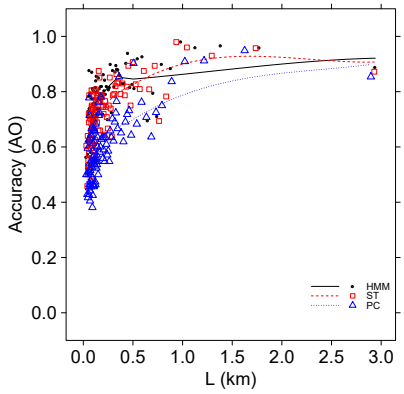
Figure 4.6: Correlation between road configuration metrics and observation-based edge accuracy (AO), in vertical axis, vs. FCD sampling period (τ), in horizontal axis. Each series represents the map-matching algorithm used.

and Figure 4.10 for R. The abscissa axes have been grouped into 100 bins, with groups containing at least two elements for the sake of representativeness. It is remarkable that the HMM algorithm outperforms the other two algorithms in all scenarios.

With all these results, we can conclude that it has been statistically demonstrated that the three algorithms behave worse under urban conditions than under interurban conditions at least at sample periods below 40 seconds. The features which are most correlated with accuracy are the number of edge counts in each cell and road density. It is shown, as well, that the average length of the edges also has some relevance while the nominal speed does not have a clear correlation with accuracy. The overall behaviour of the algorithms followed what was expected based on the literature. The use of Hidden Markov Models outperforms the others, and the worst in all cases was the Point-to-Curve geometric algorithm.



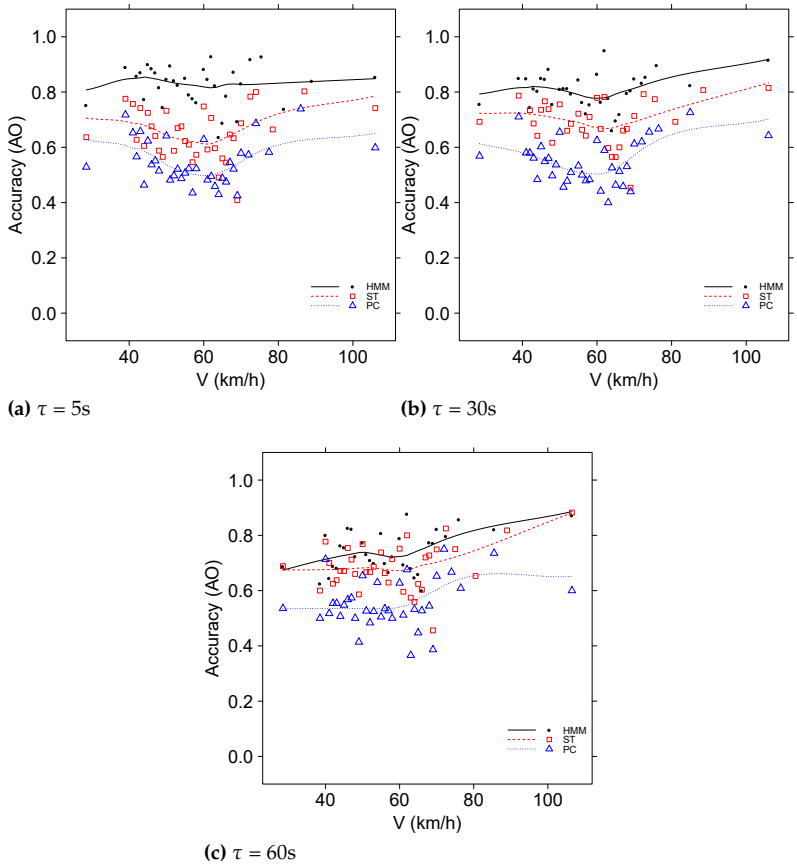
(a) $\tau = 5$s **(b)** $\tau = 30$s



(c) $\tau = 60$s

Figure 4.7: Average matching accuracy of individual observations (AO), according to edge length (L). For long edges fewer observations are available. It is shown that very low values of L obtain worse accuracy values than others slightly above.

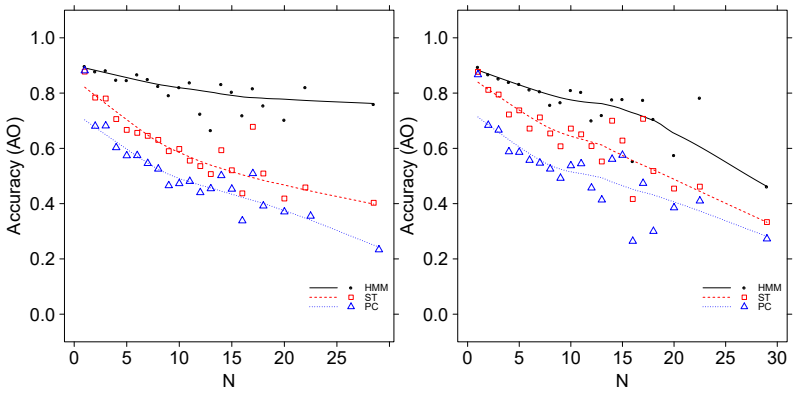
Therefore, the algorithm that we select to perform map-matching processes in urban scenarios is the one based on Hidden Markov Models. The conclusions obtained after our evaluation will lead to improvements in future algorithms. The overall good accuracy of the HMM algorithm could be enhanced if we increased efforts in the optimisation of matching observations inside more complex grid cells, when fast matching response is needed. We have identified the main factors that can be related to accuracy, and thus, further research from this point could cover the evaluation of strategies for these potential improvements of the HMM algorithm.



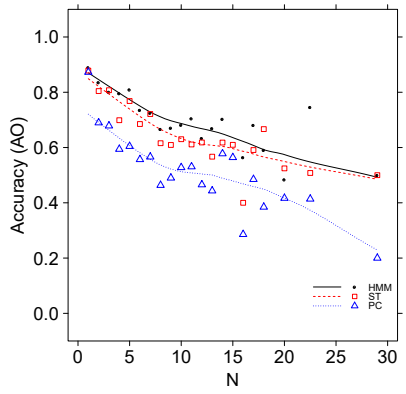
(a) $\tau = 5$s **(b)** $\tau = 30$s

(c) $\tau = 60$s

Figure 4.8: Average matching accuracy of individual observations (AO), according to average speed limit (V). It is shown that accuracy is better at very low or very high speeds, the three algorithms show a minimum at around 65km/h.

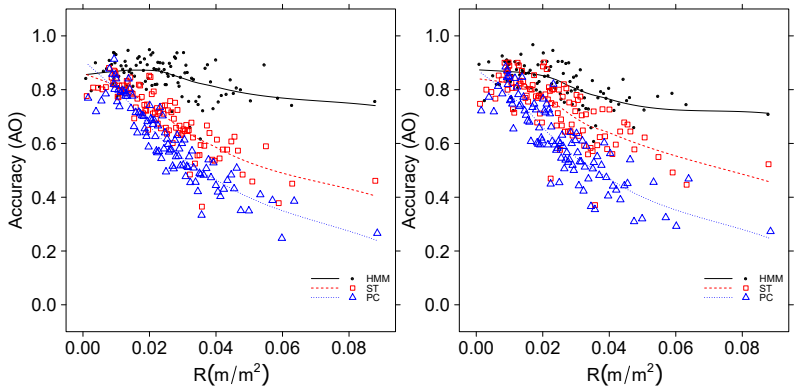


(a) $\tau = 5$s **(b)** $\tau = 30$s

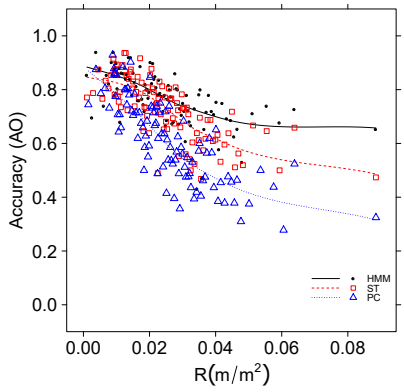


(c) $\tau = 60$s

Figure 4.9: Average matching accuracy of individual observations (AO), according to number of edge counts (N). Algorithm HMM is found to be more stable at lower sample periods.



(a) $\tau = 5$s **(b)** $\tau = 30$s



(c) $\tau = 60$s

Figure 4.10: Average matching accuracy of individual observations (AO), according to road density (R)

4.3 Summary

In this chapter we analysed the most frequent types of everyday mobility data available for urban managers. In order to allow combined exploitation of data coming from different sources, we have proposed an input data model to collect them, we have categorised and presented the main concerns when handling heterogeneities involved in these diverse types of data and the way they can be acquired, we have presented possible sources of uncertainty and, we have introduced the challenges when mapping them to the location of the city transportation resource where the movement has been observed. Moreover, we have studied how the map-matching algorithm's performance for mobile sensor data mapping is correlated with the characteristics of the road network in cities. To do so, we have introduced a quantitative evaluation of map-matching processes. It was performed regarding the digital representation of the road network configuration and its topology under four concrete grid-based features: number of edge counts, the average edge length, the average nominal speed limit and the road density. These features were defined and analysed thoroughly for 5 real world-based scenarios, including evaluations under different sampling periods ranging from 5s to 60s. We demonstrated that the three algorithms tested behave worse in urban scenarios than in non-urban scenarios for sampling periods under 40s. We selected the Hidden Markov Model-based map-matching algorithm as an appropriate method to solve some potential errors and spatial uncertainties of geolocated data, because accuracy is higher and more consistent across different scenarios. Finally, we envision that the outcomes of this analysis could be useful for future map-matching algorithm optimisations.

Estimating short-term mobility demand and applications

5

In Chapter 3, we proposed a digital representation model for urban transportation resources, the UMS. Then, in Chapter 4, we classified different mobility data sources and discussed how the data they provide can be associated or mapped to the representation model of the transportation resources. Now, in this chapter, we move on to exploit these data for advanced analytics and outline two cases for demand prediction. These two cases are aimed at short-term predictions of mobility data in two different scenarios.

The first case, described in Section 5.1, presents a method to estimate short-term road-traffic demand in a very localised area inside a city, more concretely, a complex intersection. The second case is described in Section 5.2 and presents a method to infer short-term parking occupancy estimations in underground parking lots, making use of combined measurements of traffic counts and parking occupancies of other remote parking lots across the city. Finally, the contributions of the chapter are summarised in Section 5.3.

5.1 Localised short-term mobility prediction

As previously mentioned, we have divided this chapter into two cases or scenarios where short-term mobility predictions can apply. The scenario that we discuss in this section is localised. That is, when the focus of these predictions or estimations is limited to a small geographical area inside the city. Both the data sensors used to produce the predictions and the outcome of the predictions are located in a relatively small zone, and fine spatial granularity is required. Please note that the aim here is not to predict the future situation of a single data sensor, which can be regarded as a time series problem, but the combination of many of these sensors located at short distances between them (< 5 metres). In this case, the initially proposed UMS representation and traditional map-matching techniques have limitations, since they rely on graphs comprised of edges and nodes that encode traffic links and their connections to a greater spatial extent. The use case we present is the need to estimate road traffic vehicle density in a very localised area inside a city, more specifically, at a complex intersection. Therefore, we will make use of the grid-based representation extension we proposed in subsection 4.1.4.3.

In the following subsection, we present our approach to model and compute short-term predictions of this vehicle model density.

5.1.1 Cell-based areal vehicle density prediction over complex junctions

In traffic engineering the concept of *density* is defined as the number of vehicles per unit of distance, which is valid for linear representations of traffic links. The vehicle density calculation that we propose for localised areas must be understood differently because it is not linear but areal. However, the objective is similar: to represent the concentration of vehicles. The original intuition of *vehicle density* in traffic engineering is the instantaneous view from an aerial camera, something that is usually not available for motorways. For traffic links, the calculation of this metric sometimes requires specific methods to be inferred from traffic loop counters [123]. Another related metric is *occupancy*, defined as the percent of time a point on the road is occupied by vehicles. In traffic engineering literature, both occupancy and density have been used to represent the concentration of vehicles, depending if the studies were more theoretical (favourable to density) or more practical (favourable to occupancy) [124]. Our proposed metric is a particular combination of both, we will use the concept of areal vehicle density, or vehicle density.

Estimating areal vehicle density in advance can be used to analyse the traffic flow situation and congestion risks, but it is also expected to have applications for future connected and automated vehicles and on the provision of communication networks that may be required to supply their needs. This estimation can be of great interest, for example, to estimate probabilities of communication signal shadowing effects. The Connected Vehicle paradigm has resulted in an explosion of research on methodologies that provide several degrees of automated control of all kinds. In fact, automated control of vehicle traffic can be done at multiple scales [125]. At the same time, from 5G onwards, networks are expected to strengthen the capabilities of automating communication network resources.

We describe a concrete methodology to tackle the problem of short-term traffic prediction with a fine granularity in space and time with complex urban intersections in mind, but valid for other types of locations. Our method introduces a grid-based short-term prediction solution to estimate forthcoming traffic demand, according to the number of vehicles observed and mapped onto a grid.

The objectives are:

- The development of a vehicle density model to represent the spatial distribution of vehicles inside complex roundabouts and intersections. The method infers the forthcoming spatial distribution of vehicles inside the area of interest.
- The adaptation and application of a Random Forest regression algorithm to our vehicle density model representation for short-term prediction and its evaluation under several design parameters.

The problem our work needs to address is the ability to handle very fast vehicle positioning inputs coming in simultaneously with small spatial granularity in Euclidean Space, instead of single point or lane-level measurements. Existing methods in the literature do not provide high spatial-temporal resolution in traffic prediction. Several studies have stressed the importance of making use of time-aggregated traffic measurements, even though many systems are able to collect data in short intervals (few seconds), to overcome the strong variability of traffic parameters. We use the concept of *temporal granularity* when referring to such minimal time aggregation units. Some authors use 5 minutes [71], others state that the 15-min interval is the best prediction interval [72] and some others recommended intervals not shorter than 10 minutes [73]. However, the key limitation of the existing methods for us is that they are all conceived for the domain of transportation planning and mobility where, in general, the basic spatial unit is the traffic link. To our knowledge, no other studies have studied fine-grain spatial traffic distribution inside an intersection. This problem goes beyond the urban mobility analysis as it is known today for city mobility managers. It will also likely be needed as more and more connected and automated vehicles interact with each other and with the infrastructure, and their communication requirements grow.

The set of location points and trajectories of vehicles as they pass through the area of interest, i.e. the intersection, needed to feed the density model, can be obtained from different sources. When vehicle density estimation is used for communication network applications, the location of vehicles inside a given small cell area can be obtained with custom techniques for positioning nodes. For instance, 5G communications include alternatives with availability of LoS (Line of Sight) and with NLoS (Non Line of Sight) [126]. Some positioning methods are able to use a single station [127], and some others may work cooperatively [128]. Other possible options are the inclusion of additional metadata encoded inside the communication packet headers, or the use of external sensors such as computer vision systems. However, the major challenge still lies in predicting the progression of traffic over time and identifying areas that are likely to have increased signal shadowing effects.

5.1.1.1 Areal vehicle density modelling

To model the spatial and temporal variations, we decided to rely on grid-based matching as presented in subsection 4.1.4.3 (Figure 5.1). An area covering a road intersection is divided into multiple cells of equal size.

The size of the cells is a design parameter, depending on the size and geometry of the intersection lanes. Moreover, a compromise between prediction accuracy and computation times must be ensured. We suggest sizes in which 1-2 average-sized vehicles could be accommodated. Specifically, this allows us to make a good statistical representation of vehicles that are stopped at intersection entries and moving along

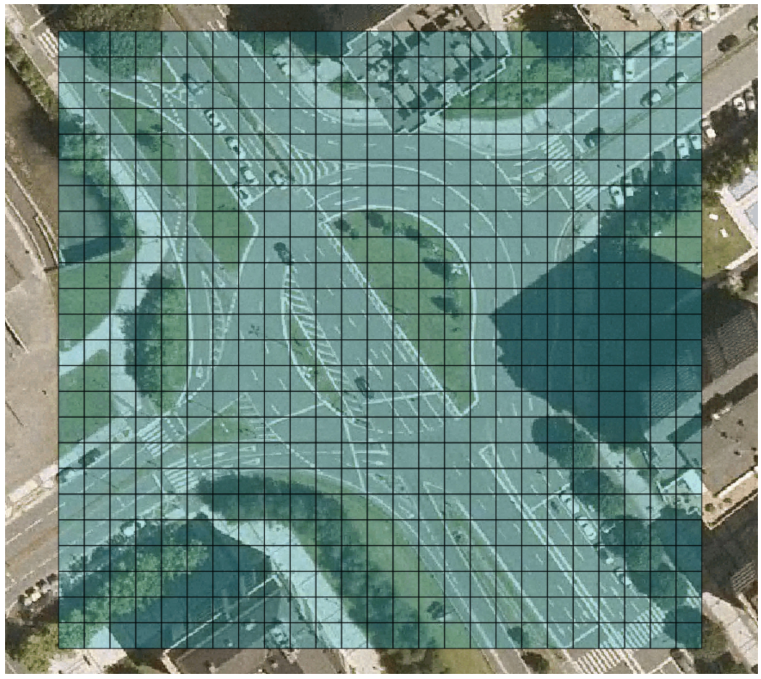


Figure 5.1: Example of grid over a roundabout in an urban scenario. Grid-cell size fits 1 or 2 cars maximum

internal lanes. We think that this size finds a balance because of its relatively small spatial grain.

The movement of vehicles through the intersection is continuous, but its data representation is discretised in space (grid cells) and time. Apart from the cell-based subdivision, positions of vehicles are observed at periodic samples. At each sample, we obtain a *snapshot*. Thus, this snapshot is understood as the current set of vehicle locations at a given time step, mapped onto the grid (shown in Figure 5.2). Note that in this case, we expect a simultaneous event that collects the location of all vehicles across the grid. In real applications, communication events from node vehicles are expected to be asynchronous, but the snapshot concept remains valid to represent the events taking place between two consequent time instants. Therefore, we define t_{samp} as time sampling resolution and T_{agg} as the integration period within which we will aggregate multiple snapshots to obtain an aggregated snapshot. The reason for aggregating snapshots is to obtain more stable measurements from highly dynamic movements of vehicles. If t_{samp} is short enough, the presence of the vehicle in all the subsequent cells can be observed along the trajectory, at least as it traverses the intersection. To ensure this, for $speed_{\max}$ as the maximum vehicle speed inside the intersection, we can set a conservative threshold $t_{\text{samp}} < min(L_X, L_Y) \cdot speed_{\max}$.

Thus, from each observation of a total duration of available observed data (in our

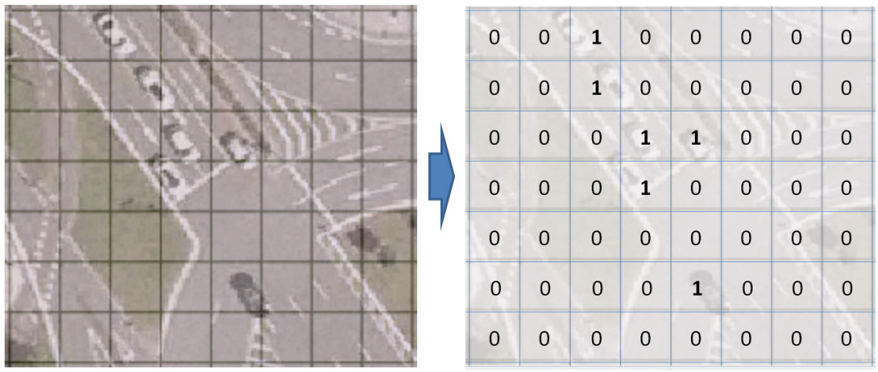


Figure 5.2: Vehicle positions observed in a snapshot, matched to grid cells. If a vehicle is located at a boundary among different cells, it is assigned to the cell that hosts the largest part of the vehicle area inside.

case a simulation run) of length T_{tot}, we obtain: a total number of $T_{\text{tot}}/t_{\text{samp}}$ snapshots, $\{sn_i\}_{i=1}^{T_{\text{tot}}/t_{\text{samp}}}$, where each snapshot sn_i is a $N_X \times N_Y$ matrix representing the number of vehicles in each of the cells at that instant; a total number of $T_{\text{tot}}/T_{\text{agg}}$ aggregated snapshots, $\{\Upsilon_k\}_{k=1}^{T_{\text{tot}}/T_{\text{agg}}}$, where each one is a $N_X \times N_Y$ matrix representing the sum aggregation of vehicles in each of the cells during the last aggregation period. We represent the sum aggregation of vehicles in one single x, y cell at aggregation period k as $v_{xy,k}$. Therefore, we represent vehicle density by Υ_k:

$$\Upsilon_k = \left[v_{xy,k}\right] = \begin{bmatrix} v_{11,k} & v_{11,k} & \dots & v_{1N_y,k} \\ v_{21,k} & v_{22,k} & \dots & v_{2N_y,k} \\ \dots & \dots & \dots & \dots \\ v_{N_x1,k} & v_{N_x2,k} & \dots & v_{N_xN_y,k} \end{bmatrix} = \sum_{i=1+(k-1)(T_{\text{agg}}/t_{\text{samp}})}^{k(T_{\text{agg}}/t_{\text{samp}})} sn_i \qquad (5.1)$$

where $1 <= k <= T_{\text{tot}}/T_{\text{agg}}$.

5.1.1.2 Short-term vehicle traffic prediction

The ordered sequence of snapshots and aggregated snapshots represent a time-series of traffic volume at each cell. Given the current discrete time k, having observed the previous pr aggregated snapshots, we want to predict a future aggregated snapshot with a prediction horizon of value h. Therefore, we use the measures from prior times $(\Upsilon_k, \Upsilon_{k-1}, \dots, \Upsilon_{k-pr})$ as input variables. This can be formulated as a regression problem, where the output variable is Υ_{k+h}.

Each Υ_k element is $N_X \times N_Y$-dimensional, and each cell will be processed in parallel as a single regression problem.

A recent approach found in the literature for short-term traffic prediction made use of an enhanced *k-Nearest Neighbour* (kNN) algorithm [82]. The authors successfully compared their proposal to other alternatives in the literature. To apply their algorithm to our problem, would have been a valid first option. However, it presents some inconveniences:

- kNN methods are *lazy learners* since they do not create any model, and at every inference, they need to look for the K nearest neighbours in the full training search space.
- kNN is hard to parallelise
- In our setting, multiple cells in the grid need to be analysed at the same inference process

A balance between accuracy and a fast response needs to be found when approaching a solution for vehicular communications. In this sense, a kNN algorithm-based inference may not be the most efficient one regarding computational requirements for the long term. Therefore, our contribution is to evaluate and compare its accuracy and computation times, with another 4-step method that we propose, based on a different machine learning technique. To do so, we developed a new prediction alternative, based on Random Forest Regression (RF) [129], which is easy to parallelise. Random Forests, an ensemble of decision trees, are able to handle thousands of input variables without variable selection, since different random subsets of features are chosen to split at each tree node. They are good at generalising with new input data if they fall inside a known data range. Two main parameters must be fine tuned: the number of trees, and the maximum depth.

The method we propose consists of the following four steps:

Grid simplification: The resolution of the grid compromises the speed of the calculation. But the topology of the road confines the greatest amount of vehicles only to some cells. We have defined a percentile threshold th_{limit}, which represents the minimum number of total vehicle observations in a cell for further consideration inside the algorithm; otherwise, we discard it. Consequently, we define a simplification ratio $0 \leq \gamma \leq 1$. The objective is to improve the computation costs with very little impact on accuracy. Another reason to consider grid simplification is that the function of each cell inside the grid might be different. While some mainly represent traffic that is approaching the intersection, others represent the main connection nodes inside the intersection, which are more affected by the effects of the signalling. For example, when a traffic light is red, or the congestion level is high, cells at these locations will observe high vehicle densities. At the same time, when a traffic light turns green, the cell may be released immediately descending the vehicle density in the corresponding Υ.
Random Forest Regression: Random Forest (RF) regression has been used for its simplicity and ease of configuration. RF has been proved as a strong machine

learning technique that provides good accuracy. Forests are ensembles of decision trees, making use of *bagging*, i.e., bootstrapped aggregation, which means that each tree is trained with random subsets of the input training records, and outputs are aggregated. The outputs of the different decision trees are averaged to obtain the final prediction value. The main difference between RF and other decision tree ensembles is that, in RF, the features considered when building each tree node are also selected at random.

Our regression model needs to fit the output value of $\Upsilon_{ij,k+h}$ based on a set of $n_{feat} = pr + 1$ features:

$$\Upsilon_{ij,k+h} = f(\Upsilon_{ij,k}, \Upsilon_{ij,k-1}, ..., \Upsilon_{ij,k-pr}) \tag{5.2}$$

Choose a Neighbour-order (Π): Here we make use of the concept of neighbour-order to specify the relationship between adjacent cells. The value of Π must be chosen with caution; in general, the higher the value, the higher is the increase in the dimensionality of the input space leading to higher variances in the output of the learning algorithm. However, one of the benefits of using RF is that its random feature selection makes it robust enough to handle high dimensional datasets efficiently. Computationally, using neighbours is a costly operation for model training, since it significantly increases the number of observed features used by the regression algorithm: $n_{feat} = (pr + 1)(2\Pi + 1)^2$.

Compute the Vehicle Presence Probability: The probability P_p represents the probability of having at least one vehicle in a cell at a certain instant. We propose a method to compute this value using the vehicle density prediction, which provides the number of vehicle observations expected in that cell during one integration period. So, c being the capacity of a cell, i.e., the number of average-sized vehicles that can occupy a cell at the same time due to its size, $m = \overline{\Upsilon_{ij,k}}$ the prediction obtained for the cell at aggregation instant t and $n_{\text{agg}} = T_{\text{agg}}/t_{\text{samp}}$ the number of samples aggregated when building the vehicle density, then:

$$P_p = \begin{cases} 1 - \frac{\binom{c\cdot(n_{\text{agg}}-1)}{m}}{\binom{c\cdot n_{\text{agg}}}{m}}, & m \leq c \cdot (n_{\text{agg}} - 1) \\ 1, & m > c \cdot (n_{\text{agg}} - 1) \end{cases} \tag{5.3}$$

where $\binom{A}{B}$ denotes a binomial coefficient.

For all these steps involving short-term vehicle traffic prediction, we can estimate the computational complexity of the proposed method. A RF regressor is based on a combination of multiple decision trees. With n_{train} number of training examples, n_{feat} number of features, n_{tree} trees and a fixed maximum tree depth max_{depth}, the computational complexity of training can be represented as $\mathcal{O}(n_{train} \cdot n_{feat} \cdot$

$max_{depth} \cdot n_{tree}$). For the full cell, since we are now able to discard some cells due to grid simplification, we get: $\mathcal{O}(\gamma \cdot N_x \cdot N_y \cdot n_{train} \cdot n_{feat} \cdot max_{depth} \cdot n_{tree})$.

When predicting, a binary decision needs to be made at each tree node, and thus, the computational complexity is reduced to $\mathcal{O}(\gamma \cdot N_x \cdot N_y \cdot max_{depth} \cdot n_{tree})$.

5.1.1.3 Experimental setup and results

We have adopted a vehicle traffic micro-simulation approach to generate the scenarios needed to test and validate the proposed methodology. In our study, we use the SUMO [61] simulator to generate datasets of vehicle traffic following different sets of flow demand. SUMO is purely microscopic and it performs a time-discrete stochastic car-following model simulation with a default step length of 1 second. The step length configuration of the simulation is equivalent to the sampling period we have defined. In our setup, the simulation has been set to $t_{\text{samp}} = 0.1$ seconds and the total length of each simulation run has been set to $T = 3600$ seconds, thus a total of 36000 snapshots were generated for each simulation run. A screenshot of the scenario, extracted from a simulation run, can be seen in Figure 5.3.

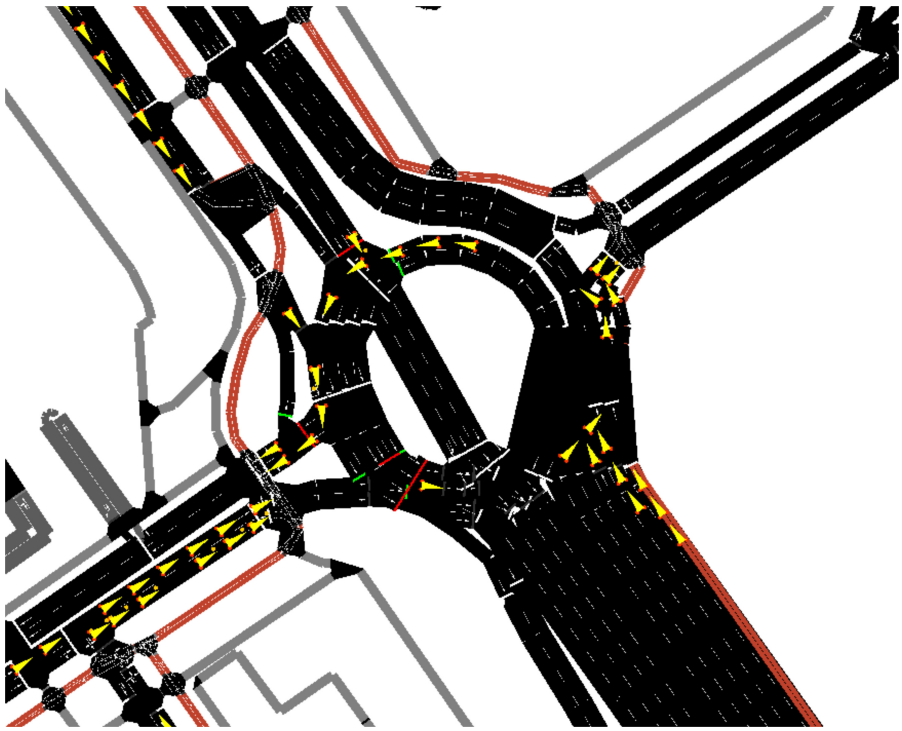


Figure 5.3: A screenshot extracted from SUMO [61] during simulation run. The small triangles represent vehicles.

The network has been imported from OSM XML files [120], and prepared making use of the NETCONVERT* tool, with special settings in order to import traffic light logics into the SUMO-routable network file. Trips and routes have been randomly generated using RANDOMTRIPS† SUMO tool, tuning the repetition-rate, to enable two different setups of demand patterns and repeating each pattern 5 times, obtaining 10 hours of traffic simulation. (see Figure 5.4).

In the proposed scenario the roundabout area has been divided in $N_x = 25$, $N_y = 24$, thus 600 cells of $L_x = 5$ and $L_y = 5$ meters.

(a) Pattern D1 **(b)** Pattern D2

Figure 5.4: Evolution of the total vehicle volume time series in each 60-minute length simulation run for demand patterns D1 (a) and D2 (b). Five simulations for each demand, cover a total duration of 10 simulation hours.

To assess the performance of our prediction algorithm, we choose the Root Mean Square Error (RMSE, Eq. 5.4) and the Mean Absolute Percentage Error (MAPE, Eq. 5.5). MAPE is unitless and insensitive to changes in the magnitude of the forecasts (unlike, e.g., the RMSE). The only negative aspect of using MAPE is that its value is not defined when the real observed value is 0. However, it allows us to make comparisons between aggregated snapshots of different T_{agg}:

$$\text{RMSE}_{ij} = \sqrt{\sum_{k=1}^{N} \frac{\left(\Upsilon_{ij,k} - \overline{\Upsilon_{ij,k}}\right)^2}{N}} \tag{5.4}$$

$$\text{MAPE}_{ij} = \left(\frac{1}{N} \sum_{k=1}^{N} \frac{\left|\Upsilon_{ij,k} - \overline{\Upsilon_{ij,k}}\right|}{\Upsilon_{ij,k}} \right) \cdot 100 \tag{5.5}$$

* https://sumo.dlr.de/docs/netconvert.html
† https://sumo.dlr.de/docs/Tools/Trip.html

where $\Upsilon_{ij,k}$ is the observed value in the i, j cell, $\overline{\Upsilon_{ij,k}}$ is the forecast value, and N is the number of records.

From the training data obtained according to the experimental setup, using hold-out, 75% of the dataset was used for training and 25% for testing, at random. By default, all tests are performed using $T_{\text{agg}} = 30s$, $pr = 15$ and $h = 2$. It is relevant to mention that results are given for equal-sized groups of traffic volume ranges. In fact, different performance is observed according to the volume group. With the objective of validating the results, we have compared these errors against those obtained by a *naive* prediction (RMSE-naive and MAPE-naive), which consists of choosing the last observed value. The *% improvement* rate represents the ratio of cells where the RMSE error of the RF algorithm is lower than RMSE-naive. The computation time is also computed and compared. To strengthen the validity of our approach, in comparison with other state of the art approaches, tests with the enhanced-kNN algorithm [82] are also presented, and from the different parameters compared, the best result for this algorithm is obtained with $K = 15$.

If we take the average *naive* predictions, and set the average RMSE values for each volume group as a reference value 1, we can make a simple comparison chart, depicted in Figure 5.5, representing the improvement ratio obtained by each algorithm: enhanced-kNN (kNN), Random Forest (RF), and Random Forest with $\Pi = 1$ (RF1). The best improvement is provided by RF1, and the worst (except for Grp.3) is provided by the kNN algorithm. All three perform best at volume groups Grp.2, Grp.3 and Grp.4.

One of the conclusions that we reach is that for extreme values (such as Grp.5 and Grp.6, which represent the highest volume groups), results are worse for the three algorithms according to the improvement rate. The main reason to explain this behaviour is that, since the number of elements in all groups is equal, for these extreme values, the volume range is wider. Therefore, prediction accuracy is worse. In addition, however, what we demonstrated with these results is that for these extreme cases, RF generalises better than kNN.

The average results of the RF short-term traffic predictions have also been evaluated for different prediction horizons, ranging from $h = 2$ to $h = 6$, which are summarised in Figure 5.6. As shown, the larger the prediction horizon, the larger the error.

The RF regression algorithm obtains better results than the naive approach in most cases for cells where vehicle density does not represent extreme values, that is, very low or very high volume groups. When looking at the RMSE error, all the configurable parameters tested achieve a better performance. Moreover, when comparing our RF and the enhanced-kNN approach, except for the volume group Grp.3, RF outperforms both RMSE and MAPE metrics, while needing shorter computation times.

Computation time: For the given training set, we measure the time needed to make inferences. The time values given below represent the average time needed to

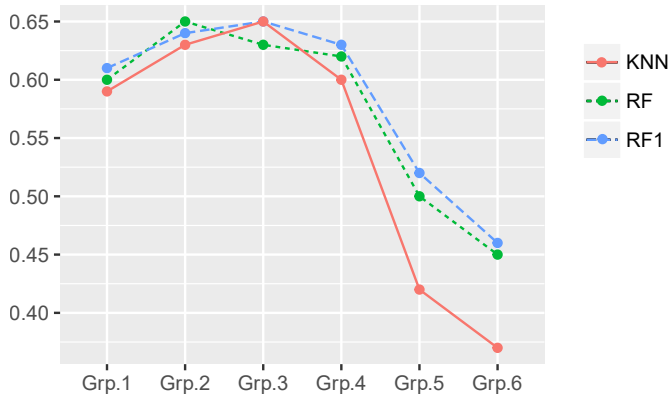


Figure 5.5: Improvement ratio of the RMSE value having the *naive* algorithm as reference value 1.

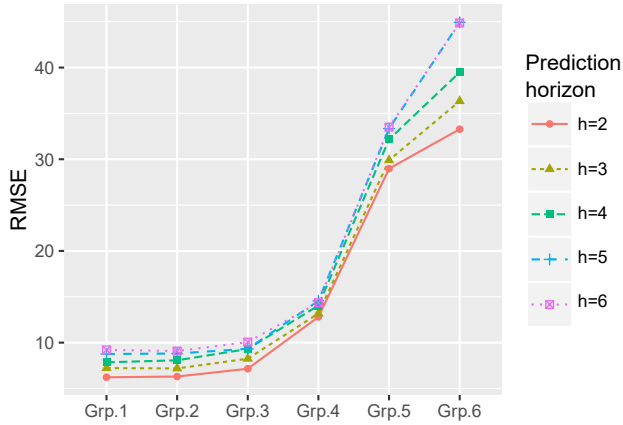


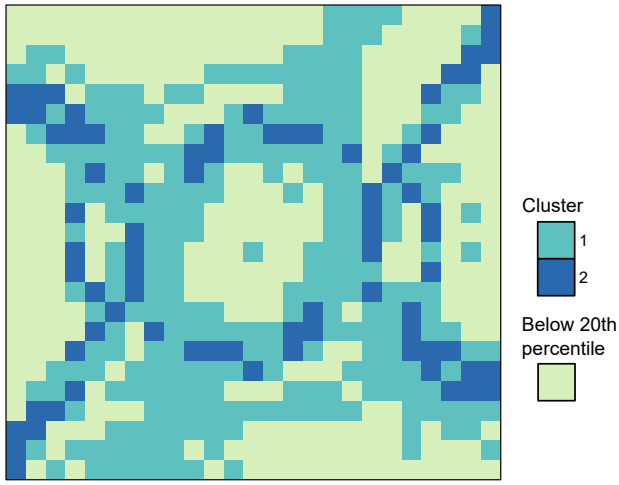
Figure 5.6: RMSE error for different volume groups and prediction horizons (h) for the RF algorithm. Higher the prediction horizon, the higher the error. Please note that the effect of the increase of the RMSE for different volume groups is because this error depends on magnitude.

compute the regression for a single cell. Therefore, the prediction for the full grid, unless simplified, implies computing $N_X \times N_Y$ regressions.

- For the RF tuning parameters tested (number of trees of 100, 250, 500 and 750 and maximum depth of 15 and 20), timing ranges between 0.26 and 6.20 ms. The increase in any two of these parameters increases computation time. If we extend the regression features including one-order neighbours, RMSE and MAPE errors are slightly smaller, but the computation time required gets worse (between 2.10 and 12.05 ms).
- The enhanced-kNN algorithm [82] takes 33.14 ms on average, by cell, to compute an estimation. The value of the chosen $K = \{5, 10, 15\}$ has little impact on the average value (±0.28 ms).

Due to the grid-like representation of intersections, the computation cost of the regression algorithm is directly linked to the number of vehicles, because more vehicles mean that more cells need to be taken into account. In order to find ways to simplify the execution, we have analysed the different behaviours of the cells inside the grid.

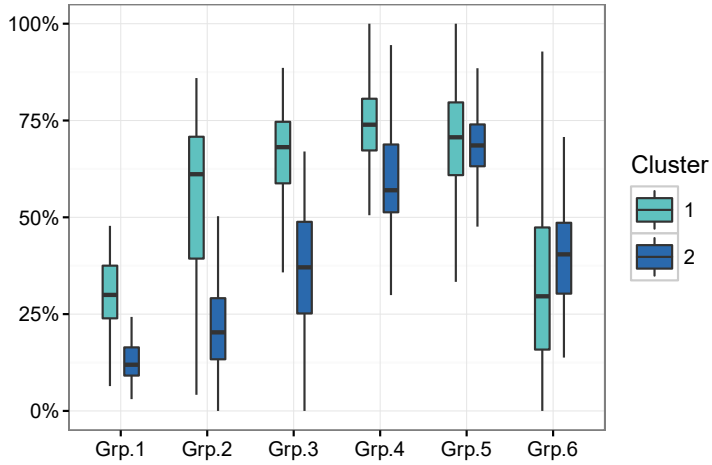
Cell analysis for grid simplification: We perform a hierarchical clustering [130] to find groups among the different cells, based on the measures we have proposed. The result is that the training data strongly support (with $p - value > 95\%$, using multi-scale bootstrap resampling) two main groups of cells, which are represented in Figure 5.7.



(a) Clusters

Figure 5.7: Assignment of cluster to each grid cell. During the grid simplification, values below 20th percentile (the th_{limit} threshold established over the total vehicle observations in a cell) are discarded; from the rest, the hierarchical clustering obtains two clear clusters of cells with similar behaviour.

We look to check if the predictions obtained by the RF for cells in each cluster are different. To show this, we represent the percentage of cells where the RF outperforms the *naive* algorithm in Figure 5.8. With these results, on one hand, we can safely assume that for cells belonging to Cluster 1, the RF algorithm is a better alternative most of the time for volume groups Grp.2, Grp.3, Grp.4 and Grp.5. On the other hand, for cells belonging to Cluster 2, the RF is a good choice only for volume groups Grp.4 and Grp.5. The prediction process can then be simplified.



(a) Cell improvement by cluster

Figure 5.8: Improvement ratios for each cluster. Cells of Cluster 1 show better results when applying the RF algorithm, than the *naive* method, improvement value over 50%.

5.1.2 Application: Wireless communication network optimisation for Urban ITS

The sample application that we present creates opportunities to control the communication network at the radio level in a 5G communication small-cell, enabling cooperative, connected and automated mobility management solutions. The study of communication demand requirements for specific ITS applications and the dynamic provision of resources will be key in the relatively near future of urban mobility.

To do so, we rely on the data-driven short-term traffic forecasting approach we have proposed, aimed at making use of the fine-grained spatio-temporal features of traffic micro-dynamics inside a localised small urban area. This area will be covered by a 5G small-cell. Our approach will help in making decisions in the next few minutes and anticipate changes in Radio Access Network (RAN) demand. Suitable traffic representation and prediction methods can help with the prediction of blockages in wireless networks for connected vehicles. We will present a method for the statistical estimation of received signal power at terahertz frequency bands, based on predicted vehicle density. This will open a wide range of possibilities to optimise dedicated links based on geo-location and mobility patterns for adequate beam-forming and beam-steering.

For this application, we make some assumptions:

1. Accurate geopositioning of vehicles is available. For instance, according to [128], sub-metre positioning accuracy with outage probabilities converging to zero could be achieved thanks to cooperative positioning in 5G networks.

2. It is considered that the communication bandwidth demand is proportional to vehicle volume.
3. The road intersection or roundabout is covered by one small-cell.
4. The number of reflectors and their location is predefined
5. Small-scale channel propagation details, such as delay distributions and queue models, are not considered.

The control action proposed to improve the RAN communications supply is to set up and manage configurable reflector orientations to overcome dynamic blockage and shadowing effects by means of virtual LoS, fed by the short-term vehicle density predictions provided. The most widely cited technology for providing high data rate satisfying 5G demands is terahertz band communication [131]. The molecular absorption, refraction, reflection and diffraction losses at the terahertz band have been quantified in many research works. However, the problem of determining the time-varying availability of LoS remains unsolved in the literature.

The approach consists of a scenario with a terahertz-band small-cell station with a set of reflectors in fixed locations, with direct LoS among them when there are no vehicles. The orientation of these reflectors can be changed in a timely manner. A system model description is depicted in Figure 5.9. Our hypothesis is that applying a data-driven short-term vehicle density prediction, we will be able to anticipate the communication needs and dynamically configure reflectors:

- to optimise the direction of beams to the areas with the highest demand,
- to reduce shadowed areas without LoS.

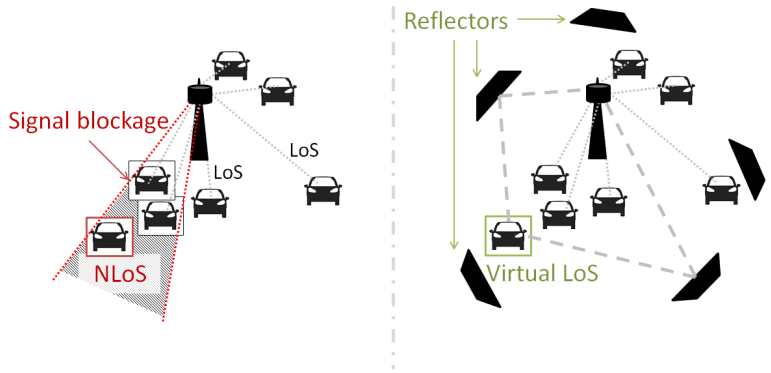


Figure 5.9: System model description. One of the vehicles is in a Non Line of Sight (NLoS) situation (left) due to signal blockage created by other vehicles. This situation is solved by the use of reflectors (right), which create a Virtual LoS.

We propose a workflow model to integrate all these functionalities into a continuous process where control actions can be taken to manage resources anticipating future situations (Figure 5.10). The aggregated snapshots are processed in a streamed way. A sliding interval of size S represents how often control actions are expected to

be taken. At time instant k, current Υ and pr previous Υ matrices are processed to obtain the $k + h$-th element. The sliding interval, represents the next k' value when the process will be repeated: $k' = k + S$. The prediction horizon h is key, since it sets the available time to make the computations for the traffic estimation and perform any control actions designed to improve the efficiency of the RAN based on that estimation, e.g., a reconfiguration of a set of reflectors using an optimisation algorithm. An important factor that needs to be considered when dimensioning the parameters is that if $S < h$, more than one process will need to be executed in parallel.

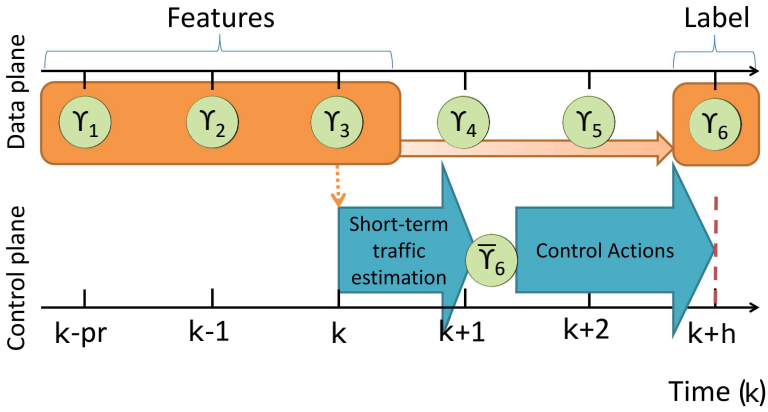
5.1.2.1 Radio coverage and LoS computation

Communications in connected vehicles can be bi-directional but, for simplicity, we use the small-cell station as the transmitter, and the vehicle node as the receiver.

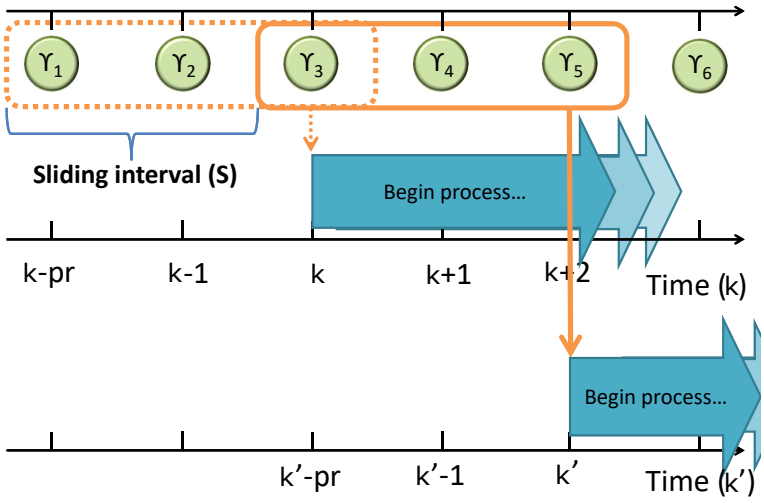
We need to compute the probability of finding an obstacle (a vehicle) in each cell, and to do so, we rely on vehicle presence probability previously introduced (P_p), derived from the vehicle density estimation. Next, following a link-budget analysis based on a 2.5D ray-tracing model [132], we analyse the power received at a given cell. We say that we use a 2.5D model because we consider some heights, without building a full 3D model. For the link-budget calculation, we must include Non-Line of Sight losses, which depend on the probabilities we have computed. Each step is described as follows:

Direct Coverage computation: The power received directly from the transmitter in a given cell is a function of the frequency of the signal, the distance from the transmitter to the centre of the corresponding cell and the path-loss [133]. The path-loss computation takes into account spreading, molecular absorption, reflections and scattering. In our case, we specifically want to compute the effect of obstacles that interfere with the LoS between the transmitter and the receiver (Figure 5.11). Thus, NLoS losses substitute scattering and reflection losses.

Non-Line of Sight losses: This loss is due to other cars located between the transmitter and the receiver. It is computed with the cascaded knife edge method [134], where losses due to up to the nearest three obstacles are summed, and are a function of the heights of obstacles, the heights of the transmitter and receiver, and the distances among them [134]. The loss due to each obstacle is a function of Vehicle Presence Probability P_p, which is derived from the estimated vehicle density, as we described previously.



(a) Single process



(b) Sliding window process

Figure 5.10: Υ matrices are processed in stream. Prediction horizon h limits the available time for the short-term prediction and any control actions that need to be taken (a). Computation can be performed upon arrival of each aggregated snapshot matrix, but this is likely to be very inefficient. In this example (b), a sliding interval of $S = 2$ is shown.

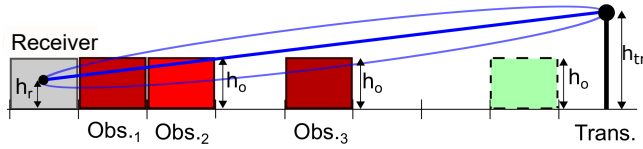


Figure 5.11: Direct LoS is obstructed by several obstacles (vehicles) found between transmitter and the receiver. These obstacles are generalised to the size of a cell and a height h_o. We calculate the shadowing weight of each obstacle at each time, based on the estimated vehicle density ($\overline{\Upsilon_{ij,k}} > 0$), but only the three nearest obstacles from the receiver are considered (shaded in red). The cell closest to the transmitter (in dashed lines), also expects a positive density, but its effect is neglected in this method.

5.1.2.2 Orienting reflectors to optimise antenna coverage

As mentioned, the use case application of the localised traffic prediction presented is the optimisation of terahertz small-cell antenna coverage, by dynamically adapting the orientation of a set of reflectors. The proposed approach includes a set of reflectors, whose locations are fixed. The actual selection of the number of reflectors, as well as the selection of locations for the small-cell antenna and the reflectors deserve a thorough study, out of the scope of this analysis, but we find that building a model for a reasonable geometric deployment based on the intersection's physical characteristics could be of interest. However, they need to be installed at a suitable height over the traffic. A cost-effective alternative is to reuse existing traffic lights, street-light poles and similar infrastructure.

The orientation of the reflectors is variable. The angle of each reflector is independent of others. Due to the fluctuation in the number of vehicles tracing different trajectories, the wireless signal demand and signal shadows may vary with time from one location to another inside the same cell. We seek to optimise the set of angle values to ensure the best coverage according to the estimated vehicle flow: maximise the power received by each vehicle, and ensure a minimum power for most vehicles. Consequently, we use the outcome of the RF predictions to feed a reflector angle orientation optimisation process based on a Genetic Algorithm (GA). GAs are well-known heuristic search and optimisation tools [135].

Before detailing the algorithm, we need to define the effects of the use of reflectors, and the objective function that the GA will try to optimise:

Reflector Coverage computation: As long as the cell is in the scope of the reflectors, each with a given orientation, to compute the total power received in a cell we have to sum the power received directly from the transmitter and the power that comes from each of the installed reflectors. To do so, the NLoS loss must be computed using the corresponding sum of distances.

Objective Function: Once we know how to compute the power in each cell, for a given set of reflector orientations, the aim is to optimise coverage of the whole scene.

For this, we propose an objective function that consists of the mean power received by each vehicle. This is a function of the estimated vehicle density and the total signal power expected in each cell. Note that $\Upsilon_{ij,k}$ is not available in the optimisation process and therefore its prediction $\overline{\Upsilon_{ij,k}}$ is used. There is a risk of orientating all the signal strength to the areas with high vehicle density, leaving areas with low vehicle density without any signal coverage. Therefore, we adapt the power received in that cell to be truncated by a threshold and penalise the cells where the minimum signal power is not received.

Genetic Algorithm: Each individual in the GA encodes a configuration of reflectors and their chromosomes are the orientation angles of the reflectors themselves. The GA must be initialised: the first population can be generated by assigning chromosomes to all individuals randomly or by inserting a specific individual. For example, as a starting point, a predefined configuration that obtains good coverage in different scenarios can be used. If we consider that the traffic will not change considerably in the following minutes, the current configuration of reflectors can also be used to initialise one of the individuals in the population. Two individuals are recombined by taking half of the angles by random from one of the parents and taking the remaining half of the angles from the other parent. Mutation increases or decreases the value of the 25% of the angles, at random.

The benefit of the GA is that it always retains a feasible result and the available time for the control actions can be dedicated to the optimisation of generations. We propose the use of average vehicle densities to create the base populations and to use them as initialisations. Then, at each new t', previously calculated populations are used.

Reflector system scenario parameters: To validate the approach, we design a reflector system scenario with a set of concrete experimental values which we proceed to describe. The transmitter is set at the centre of the roundabout, at a fixed height from the ground of $h_{\text{tr}} = 2$ metres. A total of eight reflectors with fixed locations are evenly distributed. Their locations were arbitrarily chosen, following an intuitive deployment to cover the area efficiently. Figure 5.12 shows the actual locations of the transmitter and all the reflectors in the roundabout scenario. The grid represents the spatial subdivision of the small cell, while the colour in each cell represents the variance of the aggregated snapshot. We choose to work at $f = 0.3$ THz frequency and we make use of the same transmitter power as other authors [133]. To perform the measurements we have set the following parameters: the height of vehicles, and thus the height of the obstacles, is set to $h_o = 1.5$ meters [136] while the receiver is located at $h_r = 1$ meter high. Additionally, and since the size of the cells is 5×5, we set the capacity to $c = 2$ vehicles. Regarding the GA parameters, the population has 20 individuals and 2000 generations are computed before stopping and taking the best reflector configuration so far. In each generation, two new individuals are created. A total of four angles are taken from a parent and the four remaining angles

are taken from another parent to create the first individual. The second one is created the other way round. Mutation changes the orientation of two out of eight reflectors randomly.

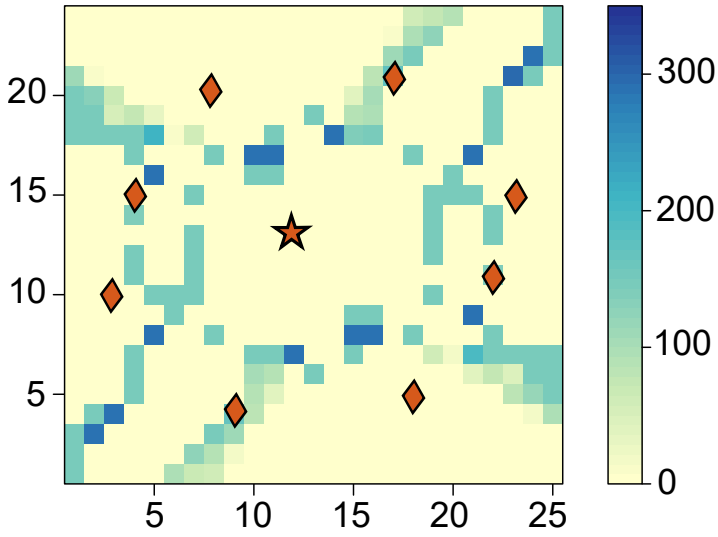


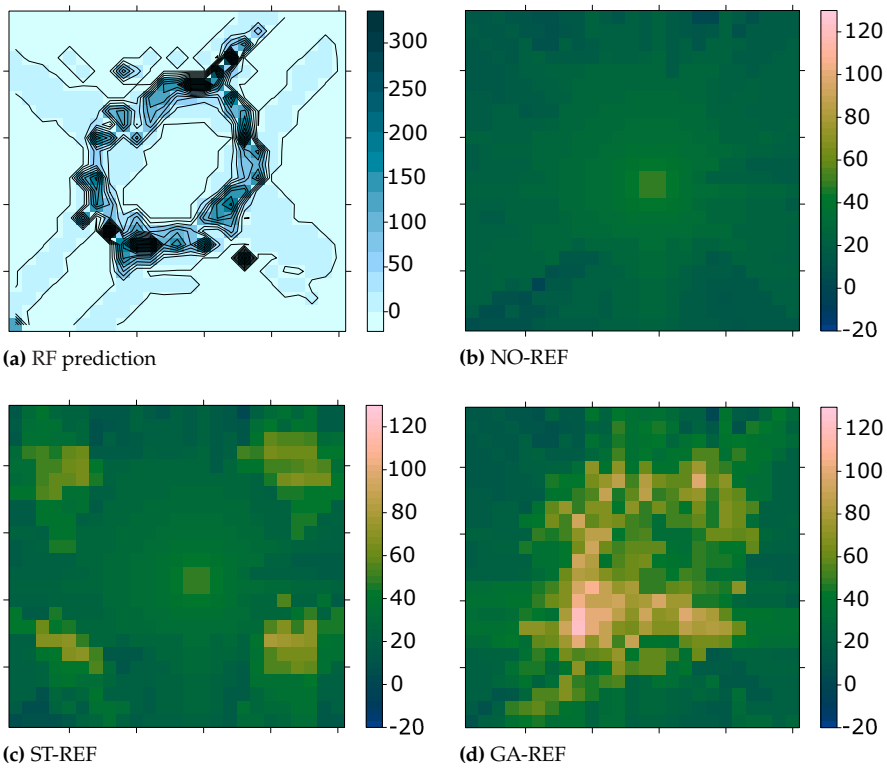
Figure 5.12: Vehicle density (Υ) in each cell during a 30 second aggregation of snapshots. Darker cells represent a higher vehicle density; generally, they are located in and around traffic lights and lane junctions. These cells also suffer from higher variance. This is due to the number of vehicles that wait stopped at the same cell for several seconds, and then they are released to continue the flow. In this figure, we include the location of the transmitter (star symbol) and the reflectors (rhomboids) used in our experimental setup.

We set the threshold of 35 dBm as the minimum amount of received power sought. Moreover, we have set an arbitrary penalisation value of 30 for testing purposes. The selection of the most suitable set of parameters was out of the scope of this study.

Three different reflector configurations have been tested: no reflectors (NO-REF), a static *naive* approach with static reflectors orientated towards the most occupied area in couples (ST-REF) and the solution provided by the GA (GA-REF). In ST-REF, we orient reflectors to the inbound-outbound roads, since the vehicles in these roads are prone to suffer NLoS coverage due to the vehicles inside the roundabout.

5.1.2.3 Radio coverage with and without a reflector system configuration

Figure 5.13 shows an illustrative comparison among radio coverage results in each cell for a given traffic prediction, under different conditions. First, at time t, we collect the sequence of the previous vehicle densities and we estimate a forthcoming one, h observations ahead. Based on that estimation ($\overline{\Upsilon_{k+h}}$), we compute the GA. Then, we compare the results for the three configurations, computing the signal power received by the real set of connected vehicles measured by Υ_{k+h}, in each case.



(a) RF prediction **(b)** NO-REF **(c)** ST-REF **(d)** GA-REF

Figure 5.13: For a given Υ prediction (a), we show the received power (dBm) for three configurations. The first one (b), represents the scenario without any reflectors. The second one (c), uses the *naive* static configuration of reflectors, and the third (d), uses the GA-based optimised configuration. In (b), we observe shadows in the upper and lower areas of the grid, due to the high volume of vehicles at the roundabout entrances. In (c), the signal received in some regions which are distant from the transmitter increases. However, coverage improves more in (d), where shadowed zones are reduced and the areas with the highest expected demand are being supplied with higher values.

The use of reflectors, even with the static approach, results in a clear improvement of the received power. Moreover, this value increases drastically with the application of the GA. Results indicate that the number of vehicles that are covered by a power lower than 35 dBm is at least 13% lower with the *naive* approach and it can reach a 25% decrease. With the GA, the decrease compared to the no reflector scenario is at least 53% and it can reach 99%.

If we relax the threshold to 25 dBm, results are similar. The decrease is between 30% and 85% when compared to the static approach, and between 39% and 100% against the GA.

Our GA-based optimisation significantly outperformed the static configuration on all the tested cases. This is a substantial advancement compared to existing work on reflectors and their orientation.

5.1.2.4 Discussion

Overall, our approach has shown promising results to estimate future NLoS situations and improve communication network coverage.

Our grid-like RF regression algorithm is an efficient prediction method in cells where vehicle density does not represent very high or very low values. Moreover, we observed that two types of cells are supported by the data we used, and that the application of our method is favourable in one of the types. This leads to the possibility of simplifying the prediction process, applying the regression algorithm only to a part of the cells and ultimately reducing computational costs, leading to a fast estimation of NLoS. Thus, extra time can be used by the genetic algorithm in optimising the orientations of reflectors. As shown, the simple use of static reflectors improves the average power received by vehicles. However, results have been greatly enhanced when using the dynamic configuration method that we have presented.

However, we would like to raise some discussion topics, in particular regarding vehicle density:

- Traffic dynamics might be expected to vary in mixed traffic, when connected and non-connected vehicles are involved (e.g., regular and automated vehicles with different levels of inter-vehicle interactions). The peculiarities of this kind of traffic were not addressed in our study, but it is a relevant research topic that deserves further analysis. Moreover, the lack of available real traffic datasets with both temporal and spatial details limits our analysis. However, the effect of using generated traffic data in our prediction approach is expected to have low impact because 1) small sampling periods are used to generate each snapshot, being able to observe the smallest changes between snapshots; and 2) many samples are combined in each aggregated snapshot, statistically smoothing the vehicle density.
- We see room for improvements in density prediction accuracy in the future; for example, combining different prediction methods. Nevertheless, additional computational needs should also be evaluated. Focusing on prediction accuracy under abnormal conditions is of great interest [137].

5.2 City-wide short-term mobility predictions

In the previous section we focused on a small and localised area inside a city, such as a complex intersection. In contrast, in this section, we will discuss the exploitation of mobility data in an area of greater extent, since we are going to address a set of mobility data sensors or information systems that provide data recorded at different locations across a city. Due to the interconnected nature of mobility, a joint analysis of data recorded simultaneously at different locations in the city is likely to provide

richer information than data from single locations and offer more extensive insights that can lead to a better understanding of the situation and trends of how transport is behaving. Moreover, the joint analysis of different types of sensors is likely to strengthen conclusions.

In the following subsections, we present our approach to model and compute the short-term predictions of parking occupancy in underground parking lots, based on data from city-wide parking occupancy (periodic event-based) and traffic counter (integration period-based) sensors.

5.2.1 Parking occupancy prediction and spatio-temporal interactions

We propose and evaluate data-driven methods based on Random Forest to forecast short-term underground parking lot occupancy, analysing the benefits of a combined use of measurements obtained at other parking lots and traffic counters deployed throughout the city network.

When off-street parking lots are reaching their maximum capacity, streets nearby are more likely to become congested. Thus, predicting the behaviour of demand in these locations is key in order to safely plan traffic redirection or dynamic pricing policies. We tackle the problem of the real-time estimation of the forth-coming occupancy of off-street parking lots, according to previous measurements of occupancy values and measurements of traffic counts at the main entries and exits of the city. To do so, we make use of the Urban Movement Space introduced in Chapter 3 that supports analysing comprehensive spatial interactions among road segments and parking entrances. Therefore, a joint analysis with traffic counts even at main entry roads is expected to provide more detailed insight of the road situation in a city centre. Based on this representation we built different short-term parking lot occupancy prediction variants and compared their performance globally as well as in special situations where predictability for non intelligent systems is more difficult.

Before focusing on parking occupancy estimation, we will introduce some of the main concepts regarding spatial weight matrices and how spatial interaction is usually modelled in the literature.

5.2.1.1 Spatial weight matrices

A spatial weight matrix W is a $N \times N$ positive matrix which specifies neighbourhood existence for each observation, with N representing the number of data locations. In each row i, a non-zero element w_{ij} defines j as being a neighbour of i [138]. Traditionally, all elements in a spatial weight matrix are defined as positive and row

standardisation is generalised (that is, all elements in each row of the W matrix sum one).

The intuition of the neighbourhood is usually one of the following three [139]: inspiration based on theoretical spatial dependence, geometrical indicators or data-based evidence.

Two main options are found in the literature when specifying spatial weights:

- Exogenous: Some researchers set the W matrix a priori, exogenously, based on some intuition about geographical proximity. Binary values that take 0 or 1 depending on the existence of a common border or depending on a threshold distance, are quite common, as well as distance decay functions. Other economic measures based on interregional trade flows, income differences and so forth are also found [140].
- Endogenous: Other authors chose to determine the spatial weight matrices endogenously from the data [141]. There are critical opinions about the selection of exogenous approaches, since the spatial lag operator can differ from the spatial structure underlying in the data [140]. The first study that tackled the computation of a weight matrix endogenously was [142], which suggested building the weights by maximising the value of Moran's I (one of the most well known metrics for spatial autocorrelation measurement).

Some authors enumerated three general problems with spatial weights: 1) the imprecision in the storage of polygon and vertices, when common boundary lengths are the basis of the neighbourhood metrics, e.g.; 2) the possible high degree of heterogeneity in the spatial distribution when the weights are based on distance 3) to ensure that the resulting weights are meaningful, finite and non-negative, special care must be taken when the weights are based on any other "economic" distance or another general metric [138].

Thus, the benefits or problems that arise due to the use of weight matrices are issues that literature has not yet clearly solved.

Most applications of spatial econometrics that make use of spatial weight matrices are based on point or areal spatial units, such as countries or regions. Therefore, when considering the application of these concepts to street networks some design decisions must be made.

5.2.1.2 Modelling spatial interaction in networks

Regarding generic spatial interaction, several studies have addressed the problem from a network point of view. In this case, a matrix of weights defines the dependency structure or the network. It represents the presence or absence of some linkage, or

the magnitude of some measure of possible dependence between each of the nodes [143].

Specifically, when it comes to transportation applications, some authors have presented limitations on the use of traditional spatial weight matrices. For instance, to account for the contributive versus competitive nature of traffic links, [144],[145], presented an alternative network weight matrix with the following features:

- Matrix elements may be negative or positive to capture the competitive and complementary nature of links
- The diagonal elements may not be fixed to zero
- The matrix elements may acknowledge the demand configuration, not only spatial dependencies based on the network topology and link costs

Some studies have confirmed that the spatial correlation between traffic links follows a more sophisticated pattern than a simple distance rule [146]. While other studies supported that network weight matrices are theoretically defensible, if they closely reflect the dependence structure of the links and embed additional network dynamics such as cost of links and demand configuration [144].

At any rate, most spatial weight matrices are fixed for a network structure. Only a few examples of dynamic spatial weight matrices are found in the literature, changing at fixed intervals or according to traffic conditions [147],[148],[149].

5.2.1.3 Parking occupancy estimation

To our knowledge, the only work covering a joint prediction between traffic and off-street parking occupancy data is the one published by Ziat et al. [150]. The authors propose an Heterogeneous Time-Series Representation algorithm (HTSR) that makes use of a representation learning method. Experiments were based on data from 50 roads and 30 car parks in the city of Lyon (France), collected during 15 days and aggregated in 20-minute windows. HTSR including prior spatial information significantly outperformed all the other models compared. In addition, joint predictions were found better than independent estimations for traffic and parking occupancies. Our work, which uses a longer time period dataset (6 month vs. 15 days), provides two main contributions that differ from theirs: 1) an analysis of algorithm behaviour during special scenarios and times of the day and 2) a study of the influence of measurement point locations and their network distances on the trained models.

The proposed methodology tackles the problem of estimating the parking occupancy values as a regression problem. Parking occupancy values are sampled at regular intervals. Our aim is, at any discrete time k, to provide a numerical estimated value $\overline{\Phi_{i,k}}$ of the occupancy of the parking lot i, where $0 \geq \Phi_{i,k} \geq 100$.

We proceed with the following sequential development of four partial objectives:

1. To train a model for short-term forecasting of parking occupancy
2. To study the contribution of using recent traffic measurements together with recent parking occupancies, and to identify which measurements have a greater influence on the prediction made
3. To study spatial interactions and check if the model can benefit from the use of spatial weight matrices
4. To study system performances globally and in special situations

5.2.2 Short-term parking occupancy forecasting model

The data-driven method chosen is the well-known, and already used in the thesis, Random Forest (RF), but we need to select the predictors. The main reason to choose a decision-tree method like Random Forest is its ability to explain the contribution of each predictor on the best trained model. This explanation cannot be done, for instance, with deep neural networks. Given that at any measurement point i, at a discrete time k, the observed parking occupancy value is $\Phi_{i,k}$, we propose the following differential metric $\Delta_{i,k}$, as defined in Eq. 5.6, which represents the evolution:

$$\Delta_{i,k} = (\Phi_{i,k} - \Phi_{i,k-1}) \tag{5.6}$$

Subsequently, we define the following l-order metric for parking lots (Eq. 5.7), where l represents the time lag between two observation samples:

$$\Delta^{l}_{i,k} = (\Phi_{i,k} - \Phi_{i,k-l}) \tag{5.7}$$

With these elements, we can now train our **RF** algorithm and obtain:

$$\begin{aligned}\overline{\Phi_{i,k+h}} = f(&\Phi_{1,k}, \Delta^{1}_{1,k}, \Delta^{2}_{1,k}, ...\Delta^{l}_{1,k}, ... \\ &\Phi_{i,k}, \Delta^{1}_{i,k}, \Delta^{2}_{i,k}, ...\Delta^{l}_{i,k}, ... \\ &\Phi_{\rho,k}, \Delta^{1}_{\rho,k}, \Delta^{2}_{\rho,k}, ...\Delta^{l}_{\rho,k})\end{aligned} \tag{5.8}$$

where $\overline{\Phi_{i,k+h}}$ is the value predicted at time k for the future $k + h$ time instant. Parameter h represents the prediction horizon. The prediction horizon can be given as a discrete time index or absolute time; in this analysis we will use the absolute time since it is more intuitive (in minutes). Note that we consider not only measurements

for the parking lot of interest but from all the ρ parking lots in the city whose data we may have.

5.2.3 Inclusion of traffic counter data

Instead of limiting the training data to parking occupancy values obtained throughout the city, we aim to evaluate the contribution of including additional data towards the improvement of estimations, such as traffic counter measurements. To distinguish them, we use **RF-unimodal** to refer to the case where RF is applied to the previously explained data (limited to parking occupancy data) vs. **RF-multimodal** to the system built by combining data from different sources (combining parking occupancy data with traffic counter data).

The formal difference between data representing counts of vehicles ($\Psi_{j,k}$, absolute integers, only bounded by the capacity of the street link) and the percentage of occupied space at parking lots ($\Phi_{i,k}$, bounded between 0-100), is significant. However, since tree-based models like RF do not require feature scaling, the preparation and inclusion in the feature set is done the same way. We define the following l-order metric for traffic counters (Eq. 5.9):

$$\Delta^l_{\rho+j,k} = (\Psi_{j,k} - \Psi_{j,k-l}) \tag{5.9}$$

The RF-multimodal algorithm relies now on additional features of τ traffic counters. The dimensionality increases up to $(l+1)*(\rho+\tau)$, as shown in Eq. 5.10:

$$\begin{aligned}\overline{\Phi_{i,k+h}} = f(&\Phi_{1,k}, \Delta^1_{1,k}, \Delta^2_{1,k}, ...\Delta^l_{1,k}, ... \\ &\Phi_{i,k}, \Delta^1_{i,k}, \Delta^2_{i,k}, ...\Delta^l_{i,k}, ... \\ &\Phi_{\rho,k}, \Delta^1_{\rho,k}, \Delta^2_{\rho,k}, ...\Delta^l_{\rho,k}, \\ &\Psi_{1,k}, \Delta^1_{\rho+1,k}, \Delta^2_{\rho+1,k}, ...\Delta^l_{\rho+1,k}, ... \\ &\Psi_{\tau,k}, \Delta^1_{\rho+\tau,k}, \Delta^2_{\rho+\tau,k}, ...\Delta^l_{\rho+\tau,k})\end{aligned} \tag{5.10}$$

5.2.4 Spatial interaction study with spatial weight matrices

To evaluate the effects of some degree of spatial interaction among the measurements at each data source location, we propose two endogenous spatial weight matrix options (Eq. 5.11) for a fully connected graph, based on the Pearson correlation:

$$W_{i,j} = Corr(\Delta_i, \Delta_j) \tag{5.11}$$

where Δ_i and Δ_j represent the full set of training records (see Equations 5.6 and 5.10 for all training k values) for two measurement points i and j, regardless of being parking lots or traffic counters.

The first matrix option is unique for the whole training data set. We name the variant **FWR** (Fixed Weight Matrix Regression) using this matrix to ponderate the regression predictors. The second option, is a set of 24 weight matrices; one for each of the 24 hours in a day. The RF variant using these set of matrices is named **HWR** (Hourly Weight Matrix Regression). Both variants are multimodal with parking and traffic measures.

Therefore, we have now defined the four RF variants that we aim to analyse: RF-unimodal, RF-multimodal, FWR and HWR.

5.2.5 Experimental setup and results

An historical data set spanning 6 months from the city of Donostia (Spain) was used to evaluate the presented methodology. The network graph was built from OSM data, parking lot occupancy and traffic counter measurements, as well as inventory locations, which were provided by the council mobility department. The data consists of 14 parking lots and 35 traffic counters reporting data at each 15-minute interval, some with uneven availability of data and missing or inconsistent values (those records were removed). Each data record includes all parking and traffic sensor data. However, we focus specifically on the study of the occupancy of a single parking lot, located at a very central location of the city. In the following, we refer to it as P_1. The selection of this parking lot is due to its variability in terms of reaching high occupancy values. Figure 5.14 represents the spatial distribution of the different measurement points.

By way of example, Figure 5.15 describes the metrics for this central parking lot during a typical two-week period. A consistent pattern is repeated on weekdays with two main occupancy peaks (around approximately 12:00 and 18:00) every day. The Friday evening peak is higher than the evening peak on other working days. In addition, occupancies on Saturdays oscillate around the maximum capacity sustained during the whole day. On Sunday, after a significant peak around 13:30 occupancy drops drastically after 18:00.

To assess the accuracy of the parking prediction, we choose the MAPE error, already introduced in a previous analysis in Section 5.1 (Eq. 5.12):

$$\mathrm{MAPE}_i = \left(\frac{1}{N} \sum_{k=1}^{N} \frac{\left| \Phi_{i,k} - \overline{\Phi_{i,k}} \right|}{\Phi_{i,k}} \right) \cdot 100 \tag{5.12}$$

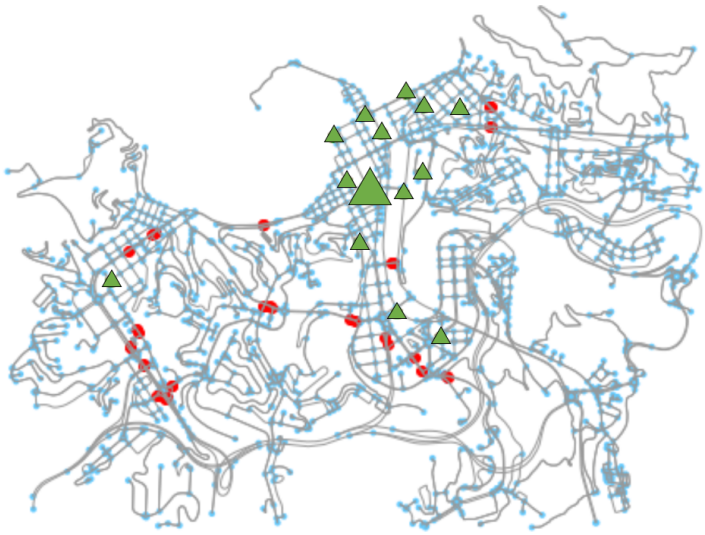


Figure 5.14: Locations of the traffic counters (red dots) and parking lots (green triangles) inside the city street network. Location of P_1 is marked with a bigger triangle.

where N is the number of observations. A negative aspect of using MAPE is that its value is not defined when the real observed value is 0 (which was the reason to rely on RMSE in Section 5.1), but this is not a drawback for this study.

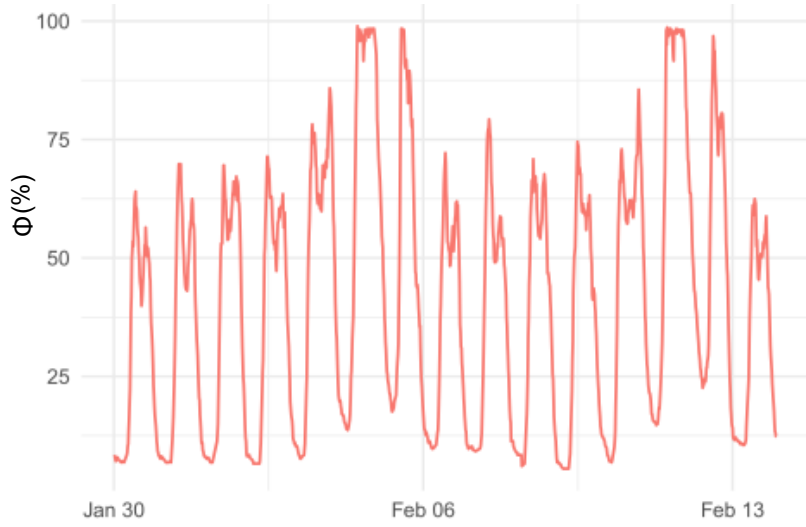
Another metric that we propose to compare the results to, based on MAPE, is the ratio of unique test records, out of all test records, for which the estimation given by an algorithm improves the estimation given by another algorithm. We name this metric the *improvement rate*, already introduced previously.

As baselines for comparison, we present two *naive* methods. The first one, called **naive** (Eq. 5.13), sets the last observed occupancy increase (or decrease) as the estimated one. The second, called **naive-stats** (Eq. 5.14), sets the statistical average value of occupancy increase (or decrease) for time of day H, and day of week D.

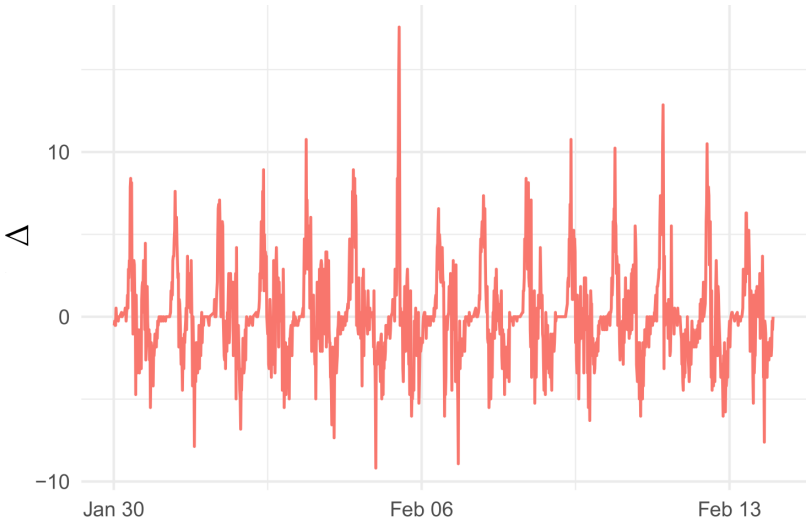
$$\overline{\Delta_{i,k}} = \Delta_{i,k-1} \tag{5.13}$$

$$\overline{\Delta_{i,k}} = \overline{\Delta_{i,H,D}} \tag{5.14}$$

They are both very easy to implement, with very low computational cost, and they can be suitable for specific situations. The first can be a good estimator if changes happen slowly and the second is an easy and strong method to estimate typical daily patterns.



(a) Absolute occupancy $\Phi_{1,k}$



(b) Differentiated occupancy $\Delta^1_{1,k}$

Figure 5.15: Parking occupancy for P_1, over 15 days, beginning on 30th January 2017, Monday.

In addition, we also include comparisons to Linear Regression (LR) with parking or parking and traffic input data, as well as Support Vector Machine algorithm (SVM) with parking data.

5.2.5.1 Data set sampling and model validation in Random Forest

The six-month training data set consists of 18800 records which are trained and tested by a 10-fold cross-validation for the Random Forest algorithm with the different data variants. In order to create the folds, all records were shuffled at random and then divided by 10. Therefore, all folds contain an equivalent distribution of heterogeneous samples of data.

In each fold, 90% of records were used for training, from which 10% were kept for model validation and RF parameter tuning (number of trees and number of variables randomly chosen at each split). Variable dimensionality is high, especially for RF-multimodal, FWR and HWR algorithms, and thus, the parameter tuning process is computationally expensive (we evaluated up to 1000 trees and 100 variables at each split). Once the best model was chosen, final tests were carried out with the remaining 10% of the records in each fold.

5.2.6 Global results

Algorithms have been built using $l = 4$, and prediction horizons of $h = 1$ (15 min), 2 (30 min), 3 (45 min), 4 (60 min), 5 (75 min), 6 (90 min). The MAPE error to assess the accuracy of each algorithm is depicted in Table 5.1 (best result for each prediction horizon in bold):

h	naive	naive-stat	LR-unim.	LR-multi.	SVM	RF-unim.	RF-multim.	FWR	HWR
15	3.66	3.00	3.24	2.99	9.39	**2.73**	2.75	2.78	3.67
30	6.20	4.79	5.70	4.95	14.90	4.22	**4.16**	4.45	6.20
45	9.72	6.32	8.43	6.83	15.93	5.48	**5.36**	5.99	8.63
60	13.43	7.81	11.48	8.74	25.44	6.62	**6.46**	7.50	11.04
75	17.21	9.22	14.78	10.71	32.82	7.72	**7.57**	9.04	13.69
90	21.04	10.04	18.62	12.74	29.06	**7.98**	8.05	9.85	16.38

Table 5.1: Average RF error of each algorithm at different prediction horizons h (in minutes)

For all prediction horizons, the best results are obtained with RF-unimodal or RF-multimodal regression algorithms with very subtle differences between them. The SVM algorithm has the worst results in all cases. It is shown that including the

proposed weight matrices does not improve the estimation (FWR behaves better than HWR). It is clearly shown that, as might intuitively be expected, all algorithms achieve worse results as prediction horizon increases. However, their behaviour is different. As depicted in Figure 5.16, for prediction horizons of 90 minutes, RF-unimodal and RF-multimodal algorithms approximate towards a 200% increase of the error at 15 minutes (192.71% and 192.13%, respectively), while the errors obtained by the naive algorithm and the unimodal linear regression increase at a faster pace, linearly, up to a 474.97% and a 475.21% increase respectively.

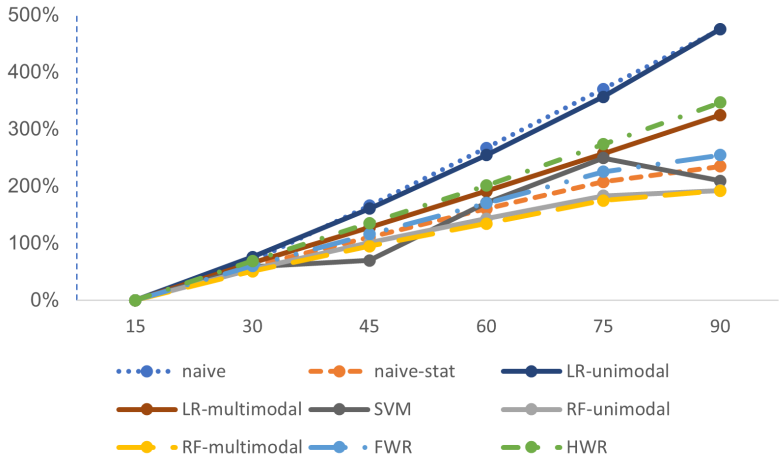


Figure 5.16: Error growth rates for increase in prediction horizon with the $h = 15$ minutes MAPE as reference value

Incorporating traffic counter data, in general terms, provides small improvements. RF-multimodal slightly outperforms the RF-unimodal variant for different horizons (improvement rates between 52.31% and 54.19%) except for $h = 15$ (48.98%). The improvement rates of the RF-multimodal algorithm range from 60.33% to 75.40% when compared to the naive algorithm, and from 53.72% to 58.25% when compared to the naive-stat algorithm.

In order to analyse the behaviour during different hours of the day, Figure 5.17 represents a disaggregated hourly representation of the average error obtained by the four RF-based variants and other comparing algorithms. For the RF variants, even though the absolute MAPE value differs, the qualitative behaviour of each variant is similar at all times. The highest errors are obtained at morning peak hours, between 8:00 and 10:00. The RF variants with the selected endogenous weight matrix, and particularly HWR, perform worse.

All the global results shown here were for parking P_1, and $l = 4$, as previously mentioned. For comparison purposes, we have also tested the average MAPE error obtained by using $l = 2$: for a prediction horizon of $h = 60$ minutes, RF-multimodal

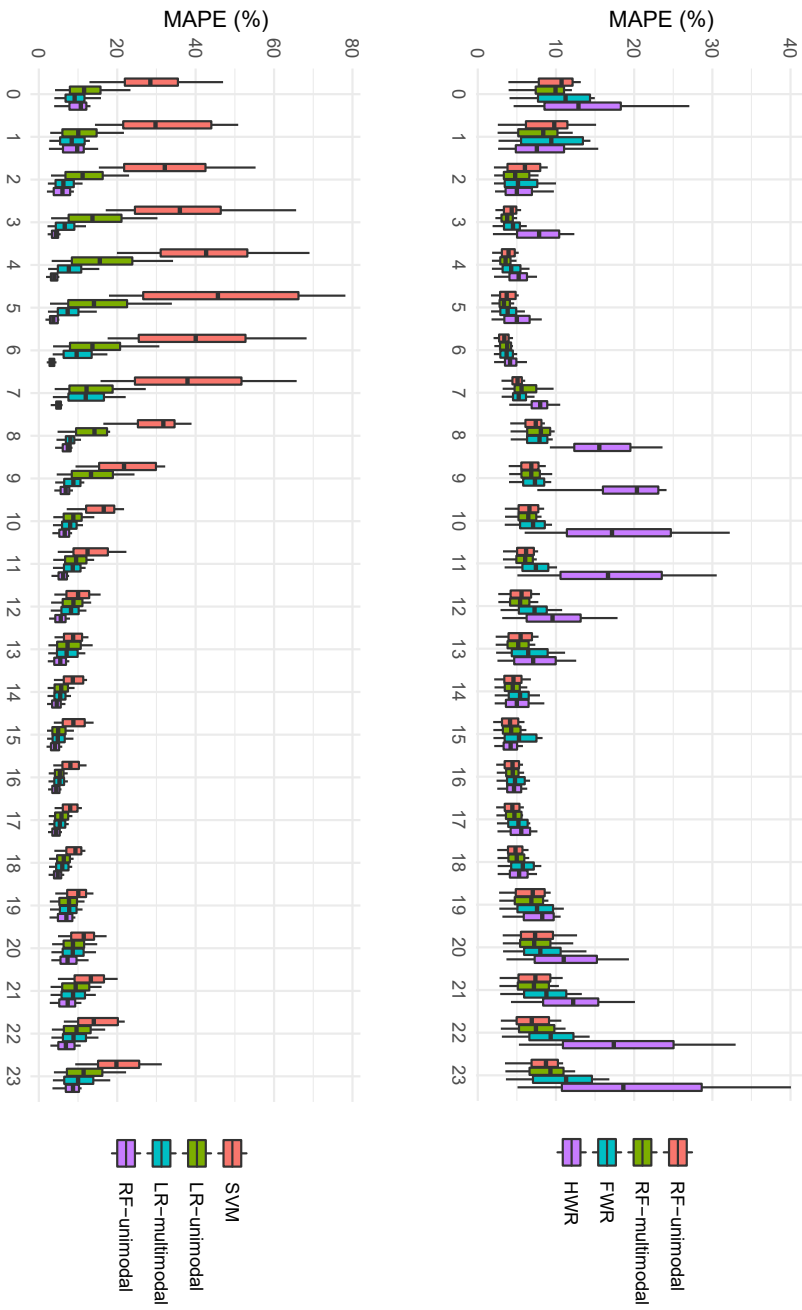
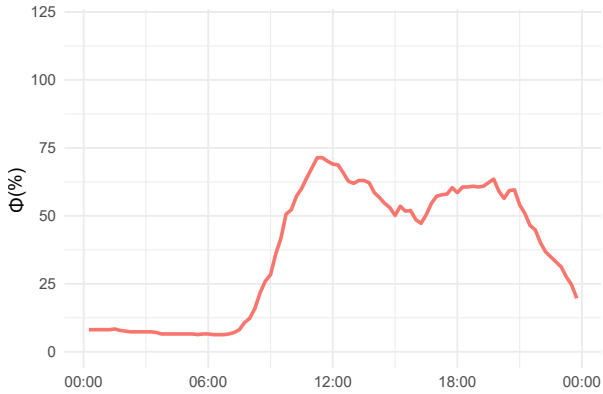


Figure 5.17: Hourly MAPE error comparison of different algorithms variants

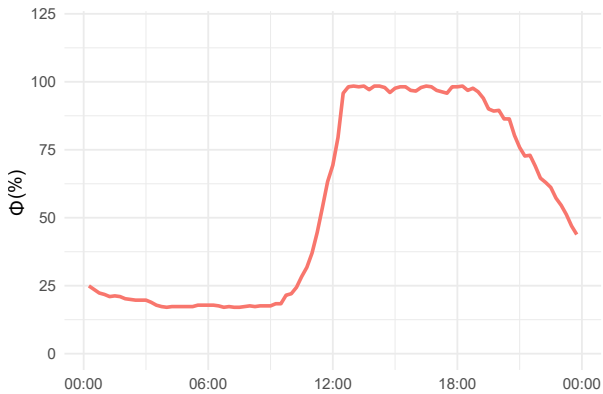
error increases from 6.46% up to 9.64% when reducing the number of previous observations used in the model. For comparison purposes, we check the performance in four other parking lots for this same setting, obtaining errors of 8.06%, 3.64%, 5.29% and 3.69%.

5.2.6.1 Results for specific scenarios

The global results show the overall performance of the compared methods within the full set of data records. However, according to specific calendar scenarios, parking occupancy may follow regular or unexpected patterns, e.g., on holidays or during special events that impact mobility throughout the city. In fact, the final aim of a system as such, is to determine how scenarios that are out of the every day known patterns will evolve.



(a) Working Thursday (D1)

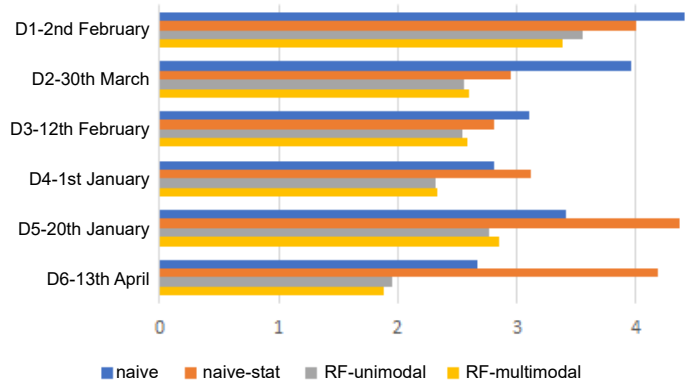


(b) Thursday during Easter (SD6)

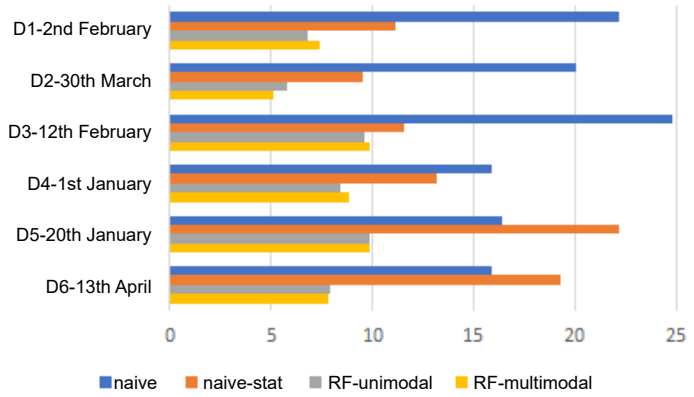
Figure 5.18: Parking occupancy evolution at P_1 on two different Thursdays.

Therefore, we have selected six different dates. On the one hand, dates D1-D3

represent typical working and weekend days. On the other hand, dates SD4-SD6 represent special dates (SD4-New Year, SD5-local holiday, SD6-Easter Thursday). The parking occupancy behaviour on special days varies. On SD4, parking occupancy is low due to a limited activity in the city. On SD5, occupancy during the day is low but reaches its maximum capacity at midnight. On SD6, occupancy rises fast at noon to reach its maximum capacity. Figure 5.18 depicts the occupancy at the parking lot under study for D1 and SD6.



(a) Selected dates with $h = 15$ minutes



(b) Selected dates with $h = 90$ minutes

Figure 5.19: RF comparison on different dates.

For these days, averaged RF errors (at $h = 60$ minutes) are depicted in Table 5.2. In Figure 5.19, the differences for shorter and larger prediction horizons are depicted. What is shown is that RF-multimodal obtains the best result except for SD5, where RF-unimodal obtains the best result. The naive-stat algorithm behaves better on regular dates than on special dates, as expected, but the opposite happens for the naive algorithm. In general, when using RF-multimodal instead of the naive-stat

method, the error is reduced by up to 37%, 47% and 63% on special dates (SD4, SD5 and SD6, respectively).

	naive	naive-stat	LR-unim.	LR-multim.	SVM	RF-unim.	RF-multim.	FWR	HWR
D1	14.60	8.58	16.42	8.39	41.88	5.92	**5.62**	6.82	11.33
D2	13.72	6.93	13.08	9.34	28.83	5.55	**4.79**	5.48	10.36
D3	15.27	10.51	7.17	7.30	11.00	6.82	**6.63**	8.46	15.29
SD4	10.20	11.09	7.40	6.58	13.73	7.15	**6.96**	8.42	9.79
SD5	10.88	15.32	8.07	8.23	14.91	**7.79**	8.09	8.13	10.21
SD6	10.43	14.06	9.28	8.83	19.95	5.67	**5.24**	6.65	9.88

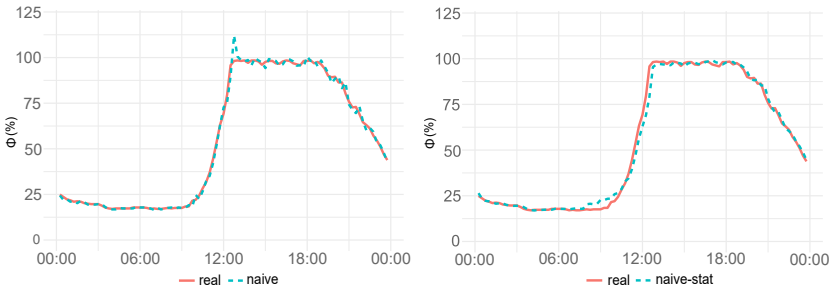
Table 5.2: Average RF error for each algorithm at different dates, at $h = 60$ minutes

In order to describe the rationale behind the behaviour of the predictions on special dates, Figures 5.20 and 5.21 represent the short-term predictions for a concrete day (13th April during Easter, SD6), where the fast rise of occupancy was not expected for an average Thursday. It is shown that for a very short prediction horizon (*h=15* minutes), all four algorithms perform well. When this prediction horizon increases, the RF-based algorithms outperform and are able to follow real behaviour very closely. It should be noted that no information about the type of day is included in the model and this makes it more flexible, as it adapts to the data received without the need to build and train specialised models. However, including additional variables such as the type of day could be of interest for future work, to see if such specialised models offer improvements worth taking into account.

5.2.7 Influence of measurement points and their locations on the RF model

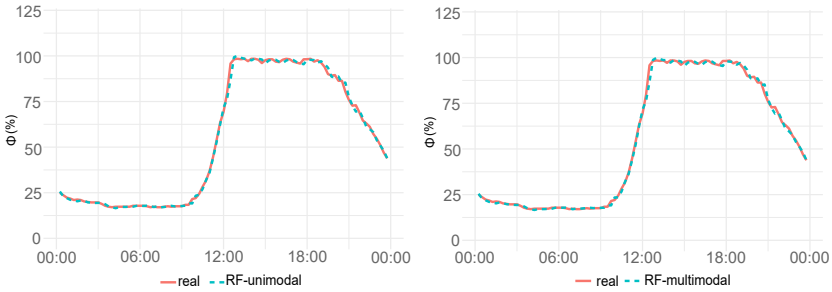
In addition, we studied how each of the measurement points distributed in the city, contributed to the random forest trees. As previously detailed, for each measurement point, we use $l + 1$ features, but for this analysis we gather them all together in the same group in order to evaluate the influence of the spatial closeness as a whole. The variable importance is found to be consistent across different train-test folds.

The analysis compares the influence of each feature used by the RF-multimodal algorithm, at different prediction horizons ($h = 15$ and $h = 60$ minutes). Two important outcomes are obtained from this analysis, which can be observed in the summary chart in Figure 5.22. Features related to the parking lot under analysis (distance = 0) are important, especially at $h = 15$, but the next most important features do not follow any distance rule. The first outcome, thus, is that the nearest



(a) Naive

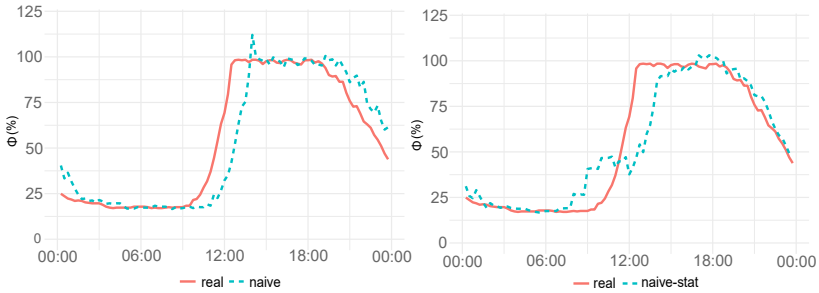
(b) Naive-stat



(c) RF-unimodal

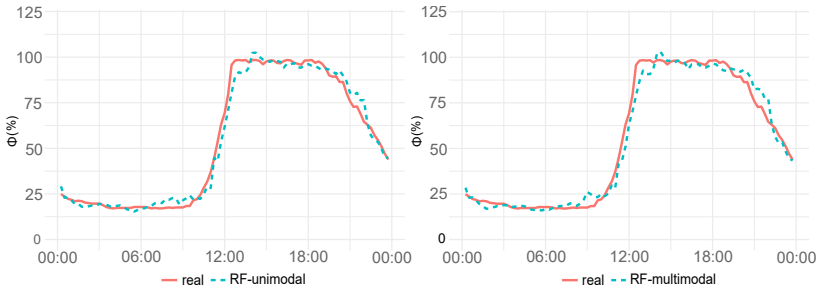
(d) RF-multimodal

Figure 5.20: Predictions during Easter Thursday with *h=15* minutes



(a) Naive

(b) Naive-stat

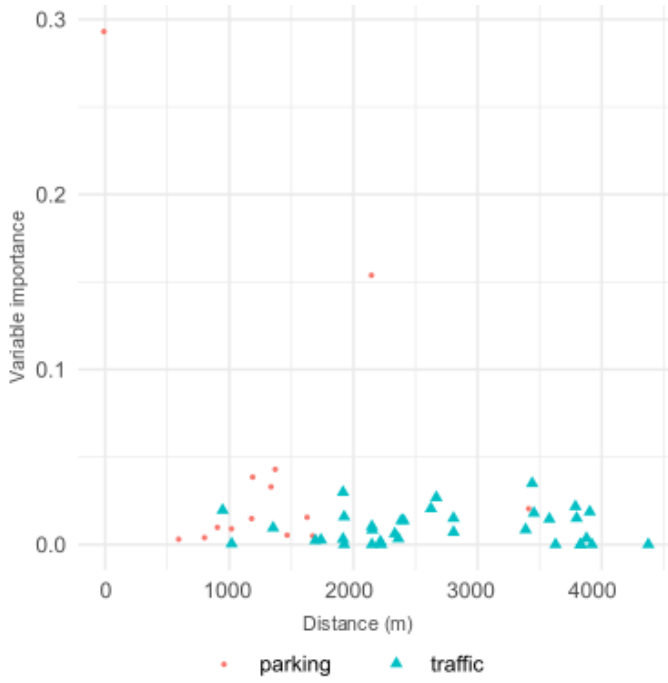


(c) RF-unimodal

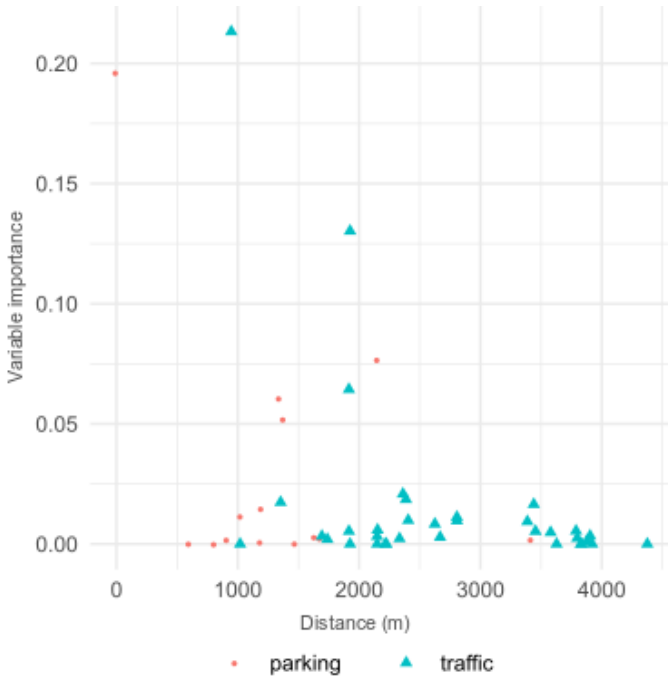
(d) RF-multimodal

Figure 5.21: Predictions over Easter Thursday with *h=90* minutes

measurement points (both parking lots or traffic counters) are not the variables that contribute the most to the RF model. The second is that, at higher prediction horizons, traffic-related variables increase their influence in the model. The traffic counters that were found to be more relevant are located at city exit directions (outbound traffic) and intermediate locations. Surprisingly, the effect of incoming traffic counters at entry directions was lower. At first, this outcome looks somewhat surprising, since intuitively, inbound traffic was expected to be more relevant to the status of future parking occupancies (inbound traffic increase is expected to increase parking occupancy). However, the same reasoning can be given for outbound traffic (outbound traffic increase is expected to decrease parking occupancy). Regardless, the truth is that this finding has opened an interesting, yet unresolved, discussion.



(a) Prediction horizon $h = 15$ minutes



(b) Prediction horizon $h = 60$ minutes

Figure 5.22: Comparison of variable importance (averaged on the 10 folds) against road network distances to the parking lot under analysis.

5.3 Summary

In this Chapter, we have contributed to methods for the exploitation of mobility data for advanced analytics, especially for short-term demand prediction. The use cases that we have studied were set in two different scenarios inside a city, which are relevant for current and future urban mobility.

On the one hand, in the scenario presented in Section 5.1, we aimed to present and solve a novel traffic prediction problem, where understanding the fine-grain spatial distribution of traffic is the main objective. Existing traffic prediction methods, which are focused on traffic links, are not the best option when handling this fine-grain spatial distribution. What we have done is:

- Propose a descriptive grid model to represent the density of these vehicles in space and time, as an alternative to the usual point and link-based representations of vehicle volumes and traffic flows in traditional traffic prediction.
- Propose and evaluate a custom method using a machine learning algorithm based on Random Forests (RF) regression to estimate short-term vehicle density, in comparison to an existing kNN-based algorithm. We have demonstrated that our RF approach is faster (in all volume groups) and more accurate (in all volume groups except one). We have also discovered the existence of two groups of cells inside the grid, which enables the simplification of the prediction process, without compromising accuracy. This outcome allows vehicle density predictions to be estimated in real time.
- Apply this study to an application use case for high quality 5G communication links, which are of great interest in the near future for connected urban mobility. More specifically, we have presented a method to relate the estimated vehicle density with the received coverage at different locations inside the intersection. Our study has shown the estimated benefits provided to for wireless communication signal power received at each cell position inside a grid. To date and to our knowledge, no previous works have addressed dynamic changes of the radio network based on traffic prediction. As we have shown, the benefits of our approach are promising and relevant in the near future for the urban mobility ecosystem, which is expected to be disrupted by different degrees of automation and multi-agent collaboration. The outcomes are important to validate the idea of highly adaptable networks and their benefits to high-frequency communications.

On the other hand, in the scenario presented in Section 5.2, we solve a different problem. We tackle the analysis of short-term prediction but from a very different point of view. The analysis is based on making use of a set of multiple sensors located at different points across the city, combining parking occupancy measures and traffic counts. Factors such as travel frequency and mode split of private car can also have a direct impact on parking occupancy but these kinds of mid/long-term variables have

not been considered. The objective has been to predict the situation of underground parking lots in the very near future and to understand the interactions among the different measurements. The outcomes are the following:

- The basic machine learning algorithm used is also Random Forest, but we have developed multiple variants on top of that algorithm to study the impact of including specific circumstances and additional information such as weight matrices. The results have shown that some of the variants proposed are very robust at different time horizons and for different scenarios on special dates.
- Moreover, the use of endogenous weight matrices has been proposed and evaluated as a variant, but this has been found not to be beneficial for prediction accuracy.
- In this application we have also studied the influence of measurement points and their locations on the RF model. From this analysis, one of the main conclusions that we have arrived at is that data from closer measurement points do not represent stronger contributions to the trained model. This is somewhat in line with other research found in the literature and reinforces the same idea when street networks are involved.

Machine learning techniques for data-driven predictions extensive and they are evolving fast. The base technique used in this study is the Random Forest Regression, which is not among the most novel techniques, but is a useful algorithm that allows the influence of the selected features to be understood. The most recent techniques, relying on neural networks, provide good results, in particular Graph Convolutional Networks look promising. These techniques are worth looking into in future work.

Conclusions 6

Throughout the different chapters of this dissertation, we have presented multiple aspects to interrelate the digital representation of the transportation resources and the data that can be obtained from mobile and fixed sensor-based observations of citizen movements that take place inside a city.

6.1 Summary

After an introduction to the study in Chapter 1, we compiled a vast number of concepts in Chapter 2, because the concepts that needed to be introduced before locating and developing our contributions are diverse. We have tried to establish some background to the geographical information systems applied to transportation, describe multiple approaches found in the literature and present some of the open issues that we have addressed, to some extent, with our contributions. These contributions were disclosed in Chapters 3, 4 and 5, and many of them have already been published in international conferences and scientific journals.

In Chapter 3, we described the type of information that is available for city mobility decision makers both regarding digital transportation infrastructure representation and movement data itself. We have explained some challenges and difficulties faced in any attempt towards integrated urban mobility analysis, in large part due to the heterogeneity of infrastructure types, and the often limited availability of quality digital maps (in particular for some non-motorised transport modes). As a consequence, we have proposed the concept named Urban Movement Space, as a combination of multiple types of information layers that can account for main heterogeneities found when representing resources offered (infrastructure assets) for mobility purposes [2]. Moreover, we have developed the concept of the Urban Movement Space from two perspectives (Eulerian and Lagrangian): a city-wide global perspective, and a local perspective with the moving entity (e.g. a vehicle) as the central element of the reference system.

In Chapter 4 we collated and classified multiple types of sensors capable of observing and quantifying mobility-related data. Then, we described the main aspects that need to be taken into account when matching the data obtained from these sensors to the location encoded on a digital map, in our case the UMS. In particular, we detailed uncertainty sources and the differences between fixed and mobile sensors when accomplishing the association of data with the digital map. In addition, one of the main contributions in this chapter was the proposal of a quantitative evaluation

of map-matching algorithms over urban and non-urban scenarios, with the use of a set of four metrics we defined to characterise the type of environment from the digitalised map. As a result, we selected the most appropriate map-matching algorithm to overcome one of the most frequent errors and uncertainties that is usually found in mobility data: spatial uncertainty [3].

Finally, in Chapter 5, we focused on short-term mobility demand predictions, as an application of the mobility data that we are able to collect and associate to an adequate digital transportation infrastructure representation. We divided the chapter into two main sections. The basic prediction method was Random Forest regression algorithm in both sections, but the applications and approaches were very different. In the first application we focused on a small local area within a city to predict the spatial distribution of traffic density in complex crossings such as roundabouts. We proposed a traffic density representation model for localised areas, promoting a different problem statement and solution alternative that traditional traffic prediction methods, which are mainly based on traffic links, have not tackled to our knowledge [4]. We obtained rapid, high accuracy results with the prediction algorithm we developed and we presented and demonstrated a possible application use-case for the improvement of 5G radio resources, which is promising for forthcoming communication needs in cooperative and connected urban mobility solutions. A patent application was submitted with the method designed for this. In contrast, we proposed a city-wide parking occupancy prediction method, combining traffic counters and parking lot occupancies located at remote distances across a city. In the evaluation of our method, we used real parking and occupancy data in the city of Donostia over a 6 month period (this work has been submitted for review in a journal but it has not been published yet).

6.2 Discussion and future work

In the previous chapters, we disclosed and evaluated multiple concrete examples where the digitalised transportation infrastructure in a city and the movement data representation are handled together. We already raised some discussion topics in each chapter and we still see that there is a vast area of potential research in multiple aspects.

With regard to the Urban Movement Space, the proposed framework is limited to the main types of transportation modes, but these are rapidly changing. For instance, dockless shared modes have not been considered. In addition, we must note that the UMS is a conceptual representation and it does not imply a concrete implementation. Regardless, building the UMS from data available in each city requires significant effort in preparing and adapting data. Building reusable data import tools to make these tasks easier may be of some assistance but pursuing universal solutions is not realistic. Moreover, we believe that there is an interesting amount of research to do

regarding the partitioning of the network representation, in order to enhance the large-scale computation and processing capabilities, especially in operations such as map-matching.

Apart from the multimodal framework presented for the infrastructure representation, the practical algorithm and quantitative evaluations we have developed and presented in this work rely mostly on car vehicle traffic (and parking) data where the combination of data derived from other transport modes has been limited. This is an aspect that we would like to extend in future studies.

If we focus on the numerical evaluations we performed, in some cases, the results of multiple algorithms have made us conclude that selecting a suitable one from a set of algorithms can be useful, if we categorise the functional behaviour of each point/cell in space (for example for map-matching and for short-term vehicle density prediction). This is worth evaluating more in depth together with any extended computational needs that this may bring.

Apart from the improvements on the work developed during this dissertation, new perspectives can be envisioned. For instance, we already mentioned the new neural network algorithms such as Graph Convolutional Networks for short-term prediction, where the information of the network structure can be encoded in the model. This is a relevant area of study that is worth exploring and validating. Moreover, in our study, we did not try to characterise the user at any point, but this is an important factor in urban mobility, and some data sources provide options to extract this kind of information. For instance, a very recent study provides a framework to characterise profiles of users that are an active part of the mobility in a city, according to "how long they are seen and how regularly this happens" [151]. They divided these profiles into structural vs. random and day visitors vs. staying visitors.

6.3 Concluding remarks

We would like to stress some concluding remarks regarding our claimed contributions. One of the main outcomes is that it is not realistic to think that a single data structure can be used in all cities to represent assets and movement data, but having a common framework can help integrate a collection of operational solutions that operate in silos for a more integrated planning, monitoring and analysis of mobility in cities.

City mobility decision makers are becoming more and more involved with city IT departments and requesting data access and APIs for any software used by transportation operators in new contracts, but this seems an objective for the long term. The Urban Movement Space concept can serve as a catalogue of the very diverse transportation mode resources representation when requesting inventory data.

Mobility-related sensor data are heterogeneous but can be classified into clear groups. This categorisation is helpful for aggregation in space and time. In particular, the spatial and network-based nature of transportation resources makes this sensor data analysis different from general time series data.

We have described differences between fixed and moving sensors. In particular, matching any information from moving sensors accurately and timely onto the digital map can be very challenging. Plenty of works have been published in the literature on algorithms that tackle this problem but our perspective on evaluating them over quantitative metrics of the network is novel.

Last but not least, our final contributions focused on short-term predictions of urban mobility data. Our approach covering traffic density estimation in a localised area such as complex intersections represents a new interdisciplinary vision as it also engages 5G. It covered the positioning data collection, the extension of density estimation towards a signal blockage probability estimation and a final application scenario using reflector orientation. The approach over a city-wide parking occupancy prediction with real data has opened the path to continue evaluating better weight matrices and discovering hidden interactions among multiple distant data collection locations.

In conclusion, we believe that this work can be a relevant point for subsequent incremental studies and developments in the very challenging area of urban mobility in a variety of aspects.

6.4 Summary of publications

During the research process, we presented a series of observable contributions to the scientific community in a series of publications in doctoral consortiums, international conferences and journals, also including a patent application. Some contents of this thesis have been partially reproduced from these works:

6.4.1 Doctoral Consortiums

- ***Harbil Arregui*** *Mobility Data Mining for Time-Dependent Urban Network Modeling. UPV/EHUko II. Doktorego Jardunaldiak. Gure ikerketa – II Jornadas Doctorales de la UPV/EHU. Nuestra investigación. July 2019*

6.4.2 International conference proceedings

- ***Harbil Arregui****, Oihana Otaegui, Olatz Arbelaitz.* Data-Driven Representation Model of Urban Movement Space. *ICGDA 2020: Proceedings of the 2020 3rd*

International Conference on Geoinformatics and Data Analysis. April 2020, pp 24–28. https://doi.org/10.1145/3397056.3397061

- *Itziar Urbieta, Irati Mendikute,* ***Harbil Arregui****, Oihana Otaegui.* Concerns on Design and Performance of a Local and Global Dynamic Map. *LBS 2019: Adjunct Proceedings of the 15th International Conference on Location-Based Services. November 2019, pp 31–36.* https://doi.org/10.34726/lbs2019

6.4.3 Journal articles

- *Michael Taynnan Barros, Gorka Velez,* ***Harbil Arregui****, Estíbaliz Loyo, Kanika Sharma, Andoni Mujika, Brendan Jennings.* CogITS: cognition-enabled network management for 5G V2X communication. *IET Intelligent Transport Systems, Volume 12, Issue 1. March 2020, pp 182–189.* https://doi.org/10.1049/iet-its.2019.0111. (Impact Factor 2019: 2.480, Q2-16/36 in Transportation Science & Technology)
- ***Harbil Arregui****, Andoni Mujika, Estibaliz Loyo, Gorka Velez, Michael T. Barros, Oihana Otaegui.* Short-Term Vehicle Traffic Prediction for Terahertz Line-of-Sight Estimation and Optimization in Small Cells. *IEEE Access, Volume 7. April 2019, pp 144408-144424.* https://doi.org/10.1109/ACCESS.2019.2910225 (Impact Factor 2019: 3.745, Q1-35/156 in Computer Science, Information Systems)
- ***Harbil Arregui****, Estibaliz Loyo, Oihana Otaegui, Olatz Arbelaitz.* Impact of the road network configuration on map-matching algorithms for FCD in urban environments. *IET Intelligent Transport Systems, Volume 12, Issue 1. February 2018, pp 12–21.* https://doi.org/10.1049/iet-its.2017.0061 (Impact Factor 2018: 2.050, Q3-19/37 in Transportation Science & Technology)

6.4.4 Under review

- ***Harbil Arregui****, Oihana Otaegui, Olatz Arbelaitz* Short-term parking occupancy forecasting with Random Forests over a city network graph. ***Pending:*** *Submitted to Journal of Intelligent Transportation Systems*

6.4.5 Patent applications

- *Seán Gaines, Oihana Otaegui Madurga, Gorka Vélez Isasmendi,* ***Harbil Arregui Martiarena****, Andoni Mujika Amunarriz, Estíbaliz Loyo Mendivil.* Method for orienting reflectors of a terahertz communications system. *24th April 2018* ***Pending:*** *EP18382277.4* ***Granted in US:*** *P191894US*

Bibliography

Here are the references in citation order.

[1] Lewis Dijkstra and Hugo Poelman. 'Cities in Europe: the new OECD-EC definition'. In: *Regional focus* 1.2012 (2012), pp. 1–13 (cited on page 1).

[2] Harbil Arregui, Oihana Otaegui, and Olatz Arbelaitz. 'Data-Driven Representation Model of Urban Movement Space'. In: *Proceedings of the 2020 3rd International Conference on Geoinformatics and Data Analysis*. ICGDA 2020. Marseille, France: Association for Computing Machinery, 2020, pp. 24–28. DOI: 10.1145/3397056.3397061 (cited on pages 3, 123).

[3] H. Arregui et al. 'Impact of the road network configuration on map-matching algorithms for FCD in urban environments'. In: *IET Intelligent Transport Systems* 12.1 (2018), pp. 12–21. DOI: 10.1049/iet-its.2017.0061 (cited on pages 3, 124).

[4] H. Arregui et al. 'Short-Term Vehicle Traffic Prediction for Terahertz Line-of-Sight Estimation and Optimization in Small Cells'. In: *IEEE Access* 7 (2019), pp. 144408–144424 (cited on pages 3, 124).

[5] TeleAtlas. *Tele Atlas ® MultiNet ® 3.5.1 User Guide* (cited on page 6).

[6] Ling Zheng et al. 'Lane-level road network generation techniques for lane-level maps of autonomous vehicles: A survey'. In: *Sustainability (Switzerland)* 11.16 (2019), pp. 1–19. DOI: 10.3390/su11164511 (cited on page 7).

[7] NDS. *Navigation Data Standard Open Lane Model Documentation. Open Lane Model version 1.0*. Tech. rep. NDS e.V., 2016, pp. 1–224 (cited on page 7).

[8] ASAM e.V. *ASAM OpenDRIVE - Open Dynamic Road Information for Vehicle Environment*. Tech. rep. Association for Standardization of Automation and Measuring Systems (ASAM), Mar. 2020 (cited on page 7).

[9] Pascal Neis and Dennis Zielstra. 'Recent Developments and Future Trends in Volunteered Geographic Information Research: The Case of OpenStreetMap'. In: *Future Internet* 6.1 (2014), pp. 76–106. DOI: 10.3390/fi6010076 (cited on pages 7, 36).

[10] Christian S. Jensen et al. 'Data Modeling for Mobile Services in the Real World'. In: *Advances in Spatial and Temporal Databases: 8th International Symposium, SSTD 2003, Santorini Island, Greece, July 2003. Proceedings*. Ed. by Thanasis Hadzilacos et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 1–9. DOI: 10.1007/978-3-540-45072-6_1 (cited on page 7).

[11] Paul Scarponcini. 'Generalized Model for Linear Referencing in Transportation'. In: *Geoinformatica* 6.1 (Mar. 2002), pp. 35–55. DOI: 10.1023/A:1013716130838 (cited on page 8).

[12] Ralf Hartmut Güting, Teixeira de Almeida, and Zhiming Ding. 'Modeling and Querying Moving Objects in Networks'. In: *The VLDB Journal* 15.2 (June 2006), pp. 165–190. DOI: 10.1007/s00778-005-0152-x (cited on pages 8, 13).

[13] A. Ajmar, E. Arco, and P. Boccardo. 'A spatial database model for mobility management'. In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* 42.2/W13 (2019), pp. 1495–1502. DOI: 10.5194/isprs-archives-XLII-2-W13-1495-2019 (cited on page 8).

[14] T. M. Adams, N. A. Koncz, and A. P. Vonderohe. *NCHRP Report 460 - Guidelines for the Implementation of Multimodal Transportation Location Referencing Systems*. Tech. rep. NCHRP, 2005, p. 59 (cited on page 8).

[15] *Digital Radio Mondiale (DRM); DRM-TMC (Traffic Message Channel)*. ETSI TS 102 668. V1.1.1. European Telecommunications Standards Institute. Apr. 2009 (cited on page 9).

[16] Kees Wevers and Teun Hendriks. 'AGORA-C on-the-fly location referencing'. In: *12th World Congress on Intelligent Transport SystemsITS AmericaITS JapanERTICO*. 2005 (cited on page 9).

[17] TomTom International B.V. *OpenLR ™ White Paper - An open standard for encoding, transmitting and decoding location references in digital maps*. Tech. rep. TomTom International B.V., 2012 (cited on page 9).

[18] Rüdiger Ebendt and Louis Calvin Touko Tcheumadjeu. 'An approach to geometry-based dynamic location referencing'. In: *European Transport Research Review* 9.3 (2017). DOI: 10.1007/s12544-017-0254-8 (cited on page 9).

[19] European Commission. *D2.8.I.7 Data Specification on Transport Networks – Technical Guidelines*. Tech. rep. March. 2014, p. 246 (cited on pages 9, 10).

[20] Knut Jetlund, Erling Onstein, and Lizhen Huang. 'Information exchange between GIS and geospatial its databases based on a generic model'. In: *ISPRS International Journal of Geo-Information* 8.3 (2019). DOI: 10.3390/ijgi8030141 (cited on page 12).

[21] Mathew A. Toups. 'A study of three paradigms for storing geospatial data: distributed-cloud model, relational database, and indexed flat file'. MA thesis. University of New Orleans Theses and Dissertations, 2016 (cited on page 12).

[22] A. Fox et al. 'Spatio-temporal indexing in non-relational distributed databases'. In: *2013 IEEE International Conference on Big Data*. Oct. 2013, pp. 291–299. DOI: 10.1109/BigData.2013.6691586 (cited on page 13).

[23] Fay Chang et al. 'Bigtable: A Distributed Storage System for Structured Data'. In: *7th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. 2006, pp. 205–218 (cited on page 13).

[24] Jia Yu, Jinxuan Wu, and Mohamed Sarwat. 'Geospark: A cluster computing framework for processing large-scale spatial data'. In: *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM. 2015, p. 70 (cited on page 13).

[25] Michael A. Whitby, Rich Fecher, and Chris Bennight. 'GeoWave: Utilizing Distributed Key-Value Stores for Multidimensional Data'. In: *Advances in Spatial and Temporal Databases*. Ed. by Michael Gertz et al. Cham: Springer International Publishing, 2017, pp. 105–122 (cited on page 13).

[26] Michael R. Evans et al. 'Fast and Exact Network Trajectory Similarity Computation: A Case-study on Bicycle Corridor Planning'. In: *Proceedings of the 2Nd ACM SIGKDD International Workshop on Urban Computing*. UrbComp '13. Chicago, Illinois: ACM, 2013. DOI: 10.1145/2505821.2505835 (cited on pages 13, 21).

[27] Ouri Wolfson et al. 'Moving Objects Databases: Issues and Solutions'. In: *Proceedings of the 10th International Conference on Scientific and Statistical Database Management*. SSDBM '98. Washington, DC, USA: IEEE Computer Society, 1998, pp. 111–122. DOI: 10.1109/SSDM.1998.688116 (cited on page 13).

[28] Elena Camossi, Michela Bertolotto, and Elisa Bertino. 'Multigranular Spatiotemporal Models: Implementation Challenges'. In: *Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '08. Irvine, California: ACM, 2008, 63:1–63:4. DOI: 10.1145/1463434.1463508 (cited on pages 13, 15).

[29] Kulsawasd Jitkajornwanich et al. 'A survey on spatial, temporal, and spatiotemporal database research and an original example of relevant applications using SQL ecosystem and deep learning'. In: *Journal of Information and Telecommunication* 0.0 (2020), pp. 1–36. DOI: 10.1080/24751839.2020.1774153 (cited on page 14).

[30] Tao Cheng et al. 'Spatiotemporal Data Mining'. In: *Handbook of Regional Science*. Ed. by Manfred M. Fischer and Peter Nijkamp. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1173–1193. DOI: 10.1007/978-3-642-23430-9_68 (cited on pages 15, 17, 19).

[31] Atsuyuki Okabe and Kokichi Sugihara. 'Basic Computational Methods for Network Spatial Analysis'. In: *Spatial Analysis along Networks*. John Wiley & Sons, Ltd, 2012, pp. 45–80. DOI: 10.1002/9781119967101.ch3 (cited on page 15).

[32] Jeremy Mennis and Diansheng Guo. 'Spatial data mining and geographic knowledge discovery—An introduction'. In: *Computers, Environment and Urban Systems* 33.6 (2009). Spatial Data Mining–Methods and Applications, pp. 403–408. DOI: https://doi.org/10.1016/j.compenvurbsys.2009.11.001 (cited on page 15).

[33] S.J. Rey. 'Mathematical Models in Geography'. In: *International Encyclopedia of the Social & Behavioral Sciences*. Ed. by Neil J. Smelser and Paul B. Baltes. Oxford: Pergamon, 2001, pp. 9393–9399. DOI: https://doi.org/10.1016/B0-08-043076-7/02516-X (cited on page 15).

[34] Kang Liu, Song Gao, and Feng Lu. 'Identifying spatial interaction patterns of vehicle movements on urban road networks by topic modelling'. In: *Computers, Environment and Urban Systems* 74 (2019), pp. 50–61. DOI: https://doi.org/10.1016/j.compenvurbsys.2018.12.001 (cited on page 16).

[35] Shashi Shekhar et al. 'Spatiotemporal Data Mining: A Computational Perspective'. In: *ISPRS International Journal of Geo-Information* 4.4 (2015), pp. 2306–2338. DOI: 10.3390/ijgi4042306 (cited on pages 17, 19).

[36] Bart Kuijpers, Harvey J. Miller, and Walied Othman. 'Kinetic prisms: incorporating acceleration limits into space–time prisms'. In: *International Journal of Geographical Information Science* 31.11 (2017), pp. 2164–2194. DOI: 10.1080/13658816.2017.1356462 (cited on page 18).

[37] Yiannis Kamarianakis and Poulicos Prastacos. 'Space-time Modeling of Traffic Flow'. In: *Comput. Geosci.* 31.2 (Mar. 2005), pp. 119–133. DOI: 10.1016/j.cageo.2004.05.012 (cited on page 18).

[38] Jianbin Chen et al. 'Localized Space-Time Autoregressive Parameters Estimation for Traffic Flow Prediction in Urban Road Networks'. In: *Applied Sciences* 8.2 (2018). DOI: 10.3390/app8020277 (cited on page 19).

[39] Shu-Lan Lin et al. 'The application of space-time ARIMA model on traffic flow forecasting'. In: *2009 International Conference on Machine Learning and Cybernetics*. Vol. 6. July 2009, pp. 3408–3412. DOI: 10.1109/ICMLC.2009.5212785 (cited on page 19).

[40] Lin Qi, Huiyuan Zhang, and Markus Schneider. 'SNAL: Spatial Network Algebra for Modeling Spatial Networks in Database Systems'. In: *Proceedings of the 2nd International Conference on Geographical Information Systems Theory, Applications and Management - Volume 1: GISTAM,* INSTICC. SciTePress, 2016, pp. 145–152. DOI: 10.5220/0005880001450152 (cited on page 19).

[41] Dapeng Zhang and Xiaokun (Cara) Wang. 'Transit ridership estimation with network Kriging: a case study of Second Avenue Subway, NYC'. In: *Journal of Transport Geography* 41 (2014), pp. 107–115. DOI: http://dx.doi.org/10.1016/j.jtrangeo.2014.08.021 (cited on pages 19, 21, 43).

[42] Venkata M. V. Gunturi and Shashi Shekhar. 'Lagrangian Xgraphs: A Logical Data-Model for Spatio-Temporal Network Data: A Summary'. In: *Advances in Conceptual Modeling: ER 2014 Workshops, ENMO, MoBiD, MReBA, QMMQ, SeCoGIS, WISM, and ER Demos, Atlanta, GA, USA, October 27-29, 2014. Proceedings*. Ed. by Marta Indulska and Sandeep Purao. Cham: Springer International Publishing, 2014, pp. 201–211. DOI: 10.1007/978-3-319-12256-4_21 (cited on page 19).

[43] Chen Zhong et al. 'Detecting the dynamics of urban structure through spatial network analysis'. In: *International Journal of Geographical Information Science* 28.11 (2014), pp. 2178–2199. DOI: 10.1080/13658816.2014.914521 (cited on page 20).

[44] Michael Batty. *Big Data, High Frequency and Low Frequency Cities*. University Lecture. July 2018. URL: http://spatialcomplexity.blogweb.casa.ucl.ac.uk/files/2018/07/BATTY-Shanghai-June-2018.pdf (cited on page 20).

[45] Andrea Hess et al. 'Data-driven Human Mobility Modeling: A Survey and Engineering Guidance for Mobile Networking'. In: *ACM Comput. Surv.* 48.3 (Dec. 2015), 38:1–38:39. DOI: 10.1145/2840722 (cited on page 20).

[46] Cynthia Chen et al. 'The promises of big data and small data for travel behavior (aka human mobility) analysis'. In: *Transportation Research Part C: Emerging Technologies* 68 (2016), pp. 285–299. DOI: http://dx.doi.org/10.1016/j.trc.2016.04.005 (cited on page 20).

[47] Akram Nour, Bruce Hellinga, and Jeffrey Casello. 'Classification of automobile and transit trips from Smartphone data: Enhancing accuracy using spatial statistics and GIS'. In: *Journal of Transport Geography* 51 (2016), pp. 36–44. DOI: http://dx.doi.org/10.1016/j.jtrangeo.2015.11.005 (cited on page 21).

[48] Wei Tu et al. 'Coupling mobile phone and social media data: a new approach to understanding urban functions and diurnal patterns'. In: *International Journal of Geographical Information Science* 31.12 (2017), pp. 2331–2358. DOI: 10.1080/13658816.2017.1356464 (cited on page 21).

[49] Kumiko Maeda et al. 'Urban pedestrian mobility for mobile wireless network simulation'. In: *Ad Hoc Networks* 7.1 (2009), pp. 153–170. DOI: http://dx.doi.org/10.1016/j.adhoc.2008.01.002 (cited on page 21).

[50] Stefano Maria Iacus et al. 'Human mobility and COVID-19 initial dynamics'. In: *Nonlinear Dynamics* 101.3 (2020), pp. 1901–1919. DOI: 10.1007/s11071-020-05854-6 (cited on page 21).

[51] Nick Warren Ruktanonchai et al. 'Using Google Location History data to quantify fine-scale human mobility'. In: *International Journal of Health Geographics* 17.1 (July 2018), p. 28. DOI: 10.1186/s12942-018-0150-z (cited on page 21).

[52] Chen Zhong et al. 'Variability in Regularity: Mining Temporal Mobility Patterns in London, Singapore and Beijing Using Smart-Card Data'. In: *PLOS ONE* 11.2 (Feb. 2016), pp. 1–17. DOI: 10.1371/journal.pone.0149222 (cited on page 21).

[53] Andrew A. Campbell et al. 'Factors influencing the choice of shared bicycles and shared electric bikes in Beijing'. In: *Transportation Research Part C: Emerging Technologies* 67 (2016), pp. 399–414. DOI: http://dx.doi.org/10.1016/j.trc.2016.03.004 (cited on page 21).

[54] Erik Jenelius and Haris N. Koutsopoulos. 'Travel time estimation for urban road networks using low frequency probe vehicle data'. In: *Transportation Research Part B: Methodological* 53 (2013), pp. 64–81. DOI: http://dx.doi.org/10.1016/j.trb.2013.03.008 (cited on page 21).

[55] S. M. Khan, K. C. Dey, and M. Chowdhury. 'Real-Time Traffic State Estimation With Connected Vehicles'. In: *IEEE Transactions on Intelligent Transportation Systems* PP.99 (2017), pp. 1–13. DOI: 10.1109/TITS.2017.2658664 (cited on pages 21, 27).

[56] Karst T. Geurs, Lissy La Paix, and Sander Van Weperen. 'A multi-modal network approach to model public transport accessibility impacts of bicycle-train integration policies'. In: *European Transport Research Review* 8.4 (2016), p. 25. DOI: 10.1007/s12544-016-0212-x (cited on page 21).

[57] Mark Brackstone and Mike McDonald. 'Car-following: a historical review'. In: *Transportation Research Part F: Traffic Psychology and Behaviour* 2.4 (1999), pp. 181–196. DOI: http://dx.doi.org/10.1016/S1369-8478(00)00005-X (cited on page 22).

[58] J. Esser and M. Schreckenberg. 'Microscopic Simulation of Urban Traffic Based on Cellular Automata'. In: *International Journal of Modern Physics C* 08.05 (1997), pp. 1025–1036. DOI: http://dx.doi.org/10.1142/S0129183197000904 (cited on page 22).

[59] Peter Hidas. 'Modelling lane changing and merging in microscopic traffic simulation'. In: *Transportation Research Part C: Emerging Technologies* 10.5 - 6 (2002), pp. 351–371. DOI: https://doi.org/10.1016/S0968-090X(02)00026-8 (cited on page 22).

[60] Hussein Dia. 'An agent-based approach to modelling driver route choice behaviour under the influence of real-time information'. In: *Transportation Research Part C: Emerging Technologies* 10.5 - 6 (2002), pp. 331–349. DOI: https://doi.org/10.1016/S0968-090X(02)00025-6 (cited on page 22).

[61] Daniel Krajzewicz et al. 'Recent Development and Applications of SUMO - Simulation of Urban MObility'. In: *International Journal On Advances in Systems and Measurements* 5.3&4 (Dec. 2012), pp. 128–138 (cited on pages 22, 90).

[62] Michael Behrisch et al. 'SUMO - Simulation of Urban MObility: An overview'. In: *SIMUL 2011, The Third International Conference on Advances in System Simulation*. 2011, pp. 63–68 (cited on page 22).

[63] Stefan Krauß. 'Microscopic Modeling of Traffic Flow: Investigation of Collision Free Vehicle Dynamics'. PhD thesis. 1998 (cited on page 22).

[64] P.G. Gipps. 'A behavioural car-following model for computer simulation'. In: *Transportation Research Part B: Methodological* 15.2 (1981), pp. 105–111. DOI: http://dx.doi.org/10.1016/0191-2615(81)90037-0 (cited on page 22).

[65] Kai Nagel and Michael Schreckenberg. 'A cellular automaton model for freeway traffic'. In: *J. Phys. I France* 2.12 (1992), pp. 2221–2229. DOI: 10.1051/jp1:1992277 (cited on page 22).

[66] Tomas Potuzak. 'Distributed-Parallel Road Traffic Simulator for Clusters of Multi-core Computers'. In: *Proceedings of the 2012 IEEE/ACM 16th International Symposium on Distributed Simulation and Real Time Applications*. DS-RT '12. Dublin, Ireland: IEEE Computer Society, 2012, pp. 195–201. DOI: 10.1109/DS-RT.2012.36 (cited on page 22).

[67] Y. Xu et al. 'Efficient graph-based dynamic load-balancing for parallel large-scale agent-based traffic simulation'. In: *Proceedings of the Winter Simulation Conference 2014*. Dec. 2014, pp. 3483–3494. DOI: 10.1109/WSC.2014.7020180 (cited on page 22).

[68] Marcus Rickert and Kai Nagel. 'Dynamic traffic assignment on parallel computers in TRANSIMS'. In: *Future Generation Computer Systems* 17.5 (2001). I: Best of Websim99. II: Traffic Simulation, pp. 637–648. DOI: https://doi.org/10.1016/S0167-739X(00)00032-7 (cited on page 22).

[69] Dirk Helbing and Péter Molnár. 'Social force model for pedestrian dynamics'. In: *Phys. Rev. E* 51 (5 May 1995), pp. 4282–4286. DOI: 10.1103/PhysRevE.51.4282 (cited on page 23).

[70] Henry Lieu. 'Revised Monograph on Traffic Flow Theory'. In: *US Department of Transportation Federal Highway Administration* (2003) (cited on page 25).

[71] Jiuh-Biing Sheu. 'A stochastic modeling approach to dynamic prediction of section-wide inter-lane and intra-lane traffic variables using point detector data'. In: *Transportation Research Part A: Policy and Practice* 33.2 (1999), pp. 79–100. DOI: https://doi.org/10.1016/S0965-8564(98)00023-8 (cited on pages 25, 85).

[72] Peter Babington. *HCM 2010 : Highway Capacity Manual*. Washington, D.C: Transportation Research Board, 2010 (cited on pages 25, 85).

[73] P Vythoulkas. 'Alternative approaches to short term traffic forecasting for use in driver information systems'. In: *Transportation and traffic theory* 12 (1993), pp. 485–506 (cited on pages 25, 85).

[74] Moshe Levin and Yen-Der Tsao. 'On forecasting freeway occupancies and volumes (abridgment)'. In: *Transportation Research Record* 773 (1980) (cited on page 26).

[75] Wei-Hua Lin, Qingying Lu, and Joy Dahlgren. 'Dynamic procedure for short-term prediction of traffic conditions'. In: *Transportation Research Record: Journal of the Transportation Research Board* 1783 (2002), pp. 149–157 (cited on page 26).

[76] Eleni I. Vlahogianni, John C. Golias, and Matthew G. Karlaftis. 'Short-term traffic forecasting: Overview of objectives and methods'. In: *Transport Reviews* 24.5 (2004), pp. 533–557. DOI: 10.1080/0144164042000195072 (cited on pages 26, 27).

[77] Eleni I. Vlahogianni, Matthew G. Karlaftis, and John C. Golias. 'Short-term traffic forecasting: Where we are and where we are going'. In: *Transportation Research Part C: Emerging Technologies* 43, Part 1 (2014). Special Issue on Short-term Traffic Flow Forecasting, pp. 3–19. DOI: https://doi.org/10.1016/j.trc.2014.01.005 (cited on page 26).

[78] M.G. Karlaftis and E.I. Vlahogianni. 'Statistical methods versus neural networks in transportation research: Differences, similarities and some insights'. In: *Transportation Research Part C: Emerging Technologies* 19.3 (2011), pp. 387–399. DOI: https://doi.org/10.1016/j.trc.2010.10.004 (cited on page 26).

[79] Gaetano Fusco, Chiara Colombaroni, and Natalia Isaenko. 'Short-term speed predictions exploiting big data on large urban road networks'. In: *Transportation Research Part C: Emerging Technologies* 73 (2016), pp. 183–201. DOI: http://dx.doi.org/10.1016/j.trc.2016.10.019 (cited on page 26).

[80] Jianhua Guo, Wei Huang, and Billy M. Williams. 'Real time traffic flow outlier detection using short-term traffic conditional variance prediction'. In: *Transportation Research Part C: Emerging Technologies* 50 (2015). Special Issue on Road Safety and Simulation, pp. 160–172. DOI: https://doi.org/10.1016/j.trc.2014.07.005 (cited on page 26).

[81] Yanru Zhang, Yunlong Zhang, and Ali Haghani. 'A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model'. In: *Transportation Research Part C: Emerging Technologies* 43.Part 1 (2014). Special Issue on Short-term Traffic Flow Forecasting, pp. 65–78. DOI: https://doi.org/10.1016/j.trc.2013.11.011 (cited on page 27).

[82] Filmon G. Habtemichael and Mecit Cetin. 'Short-term traffic flow rate forecasting based on identifying similar traffic patterns'. In: *Transportation Research Part C: Emerging Technologies* 66.Supplement C (2016). Advanced Network Traffic Management: From dynamic state estimation to traffic control, pp. 61–78. DOI: https://doi.org/10.1016/j.trc.2015.08.017 (cited on pages 27, 88, 92, 93).

[83] Jianhua Guo, Wei Huang, and Billy M. Williams. 'Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification'. In: *Transportation Research Part C: Emerging Technologies* 43.Part 1 (2014). Special Issue on Short-term Traffic Flow Forecasting, pp. 50–64. DOI: https://doi.org/10.1016/j.trc.2014.02.006 (cited on page 27).

[84] Stephen Clark. 'Traffic prediction using multivariate nonparametric regression'. In: *Journal of transportation engineering* 129.2 (2003), pp. 161–168 (cited on page 27).

[85] Head K Larry. 'Event-based short-term traffic flow prediction model'. In: *Transportation Research Record* 1510 (1995), pp. 125–143 (cited on page 27).

[86] Tim Tiedemann et al. 'Concept of a Data Thread Based Parking Space Occupancy Prediction in a Berlin Pilot Region'. In: *Artificial Intelligence for Transportation: Advice, Interactivity and Actor Modeling: Papers from the 2015 AAAI Workshop*. Association for the Advancement of Artificial Intelligence, 2015 (cited on page 27).

[87] Thomas Martinetz and Klaus Schulten. *A "Neural-Gas" Network Learns Topologies*. 1991. URL: http://web.cs.swarthmore.edu/%7B~%7Dmeeden/DevelopmentalRobotics/fritzke95.pdf (cited on page 27).

[88] Eleni I. Vlahogianni et al. 'A Real-Time Parking Prediction System for Smart Cities'. In: *Journal of Intelligent Transportation Systems* 20.2 (2016), pp. 192–204. DOI: 10.1080/15472450.2015.1037955 (cited on page 27).

[89] Haitao Zhang and Jiming Li. 'Deep learning based parking prediction on cloud platform'. In: *Proceedings - 2018 4th International Conference on Big Data Computing and Communications, BIGCOM 2018* (2018), pp. 132–137. DOI: 10.1109/BIGCOM.2018.00028 (cited on page 28).

[90] Alejandro Vaisman and Esteban Zimányi. 'Mobility data warehouses'. In: *ISPRS International Journal of Geo-Information* 8.4 (2019), pp. 1–22. DOI: 10.3390/ijgi8040170 (cited on page 28).

[91] Geoff Boeing. 'OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks'. In: *Computers, Environment and Urban Systems* 65 (2017), pp. 126–139. DOI: 10.1016/j.compenvurbsys.2017.05.004 (cited on page 28).

[92] Luca Pappalardo et al. 'Scikit-mobility: a Python library for the analysis, generation and risk assessment of mobility data'. In: *arXiv* (2019) (cited on page 29).

[93] Rupprecht Consult. *Guidelines for Developing and Implementing a Sustainable Urban Mobility Plan, Second Edition*. Tech. rep. Forschung & Beratung GmbH (editor), 2019 (cited on page 32).

[94] Hongyu Zhang and Jacek Malczewski. 'Volunteered Geographic Information and the Future of Geospatial Data'. In: ed. by C. E. Calazans Campelo, M. Bertolotto, and P. Corcoran. Hershey PA, USA: IGI Global, 2017. Chap. Quality Evaluation of Volunteered Geographic Information:The Case of OpenStreetMap (cited on page 36).

[95] Y. Horita and R. S. Schwartz. 'Extended electronic horizon for automated driving'. In: *2015 14th International Conference on ITS Telecommunications (ITST)*. 2015, pp. 32–36 (cited on page 46).

[96] Kenya Sato et al. 'Stream LDM: local dynamic map (LDM) with stream processing technology'. In: *The Science Engineering Review of Doshisha University* 53.3 (2012) (cited on page 48).

[97] Julian Eggert et al. 'Driving situation analysis with relational local dynamic maps (R-LDM)'. In: *Proc. Symp. Future Active Safety Technology*. 2017 (cited on page 48).

[98] F. Damerow et al. 'Intersection Warning System for Occlusion Risks Using Relational Local Dynamic Maps'. In: *IEEE Intelligent Transportation Systems Magazine* 10.4 (2018), pp. 47–59 (cited on page 48).

[99] D. Bernstein and A. Kornhauser. 'An introduction to map matching for personal navigation assistants'. In: *Proceedings of the International Conference of Transportation Research Board*. 1996 (cited on pages 62, 65, 72).

[100] C. White, D. Bernstein, and A. Kornhauser. 'Some map matching algorithms for personal navigation assistants'. In: *Transportation Research Part C: Emerging Technologies* 8.1-6 (2000), pp. 91–108. DOI: http://dx.doi.org/10.1016/S0968-090X(00)00026-7 (cited on pages 62, 72).

[101] J. S. Greenfeld. 'Matching GPS Observations to Locations on a Digital Map'. In: Washington DC: Transportation Research Board, Jan. 2002 (cited on page 62).

[102] S. Brakatsoulas et al. 'On Map-Matching Vehicle Tracking Data'. In: *Proceedings of the 31st International Conference on Very Large Data Bases, Trondheim, Norway, August 30 - September 2, 2005*. 2005, pp. 853–864 (cited on page 62).

[103] N. Velaga, M. Quddus, and A. Bristow. 'Developing an enhanced weight-based topological map-matching algorithm for intelligent transport systems'. In: *Transportation Research Part C: Emerging Technologies* 17.6 (2009), pp. 672–683. DOI: http://dx.doi.org/10.1016/j.trc.2009.05.008 (cited on page 62).

[104] Y. Lou et al. 'Map-matching for Low-sampling-rate GPS Trajectories'. In: *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. GIS '09. Seattle, Washington: ACM, 2009, pp. 352–361. DOI: 10.1145/1653771.1653820 (cited on pages 62, 65, 72).

[105] J. Yuan et al. 'An Interactive-Voting Based Map Matching Algorithm'. In: *Mobile Data Management (MDM), 2010 Eleventh International Conference on*. May 2010, pp. 43–52. DOI: 10.1109/MDM.2010.14 (cited on pages 62, 72).

[106] H. Yang et al. 'An Enhanced Weight-based Topological Map Matching Algorithm for Intricate Urban Road Network'. In: *Procedia - Social and Behavioral Sciences* 96.0 (2013). Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th COTA International Conference of Transportation Professionals (CICTP2013), pp. 1670–1678. DOI: http://dx.doi.org/10.1016/j.sbspro.2013.08.189 (cited on page 62).

[107] Ying J. et al. 'Spatial-temporal mining for urban map-matching'. In: *UrbComp'14*. New York, USA, Aug. 2014 (cited on page 62).

[108] M. Quddus and S. Washington. 'Shortest path and vehicle trajectory aided map-matching for low frequency GPS data'. In: *Transportation Research Part C: Emerging Technologies* 55.0 (2015). Engineering and Applied Sciences Optimization (OPT-i) - Professor Matthew G. Karlaftis Memorial Issue, pp. 328–339. DOI: http://dx.doi.org/10.1016/j.trc.2015.02.017 (cited on page 62).

[109] W. Y. Ochieng, M. A. Quddus, and R. B. Noland. 'Map-matching in complex urban road networks'. In: *Brazilian Journal of Cartography* 55.2 (2004), pp. 1–18 (cited on page 62).

[110] 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*. GIS '09. Seattle, Washington: ACM, 2009, pp. 336–343. DOI: 10.1145/1653771.1653818 (cited on pages 62, 73).

[111] C.Y. Goh et al. 'Online map-matching based on Hidden Markov model for real-time traffic sensing applications'. In: *Intelligent Transportation Systems (ITSC), 2012 15th International IEEE Conference on*. Sept. 2012, pp. 776–781. DOI: 10.1109/ITSC.2012.6338627 (cited on pages 62, 73).

[112] Y. Ahres et al. 'Real-Time Dense Map Matching with Naive Hidden Markov Models: Delay versus Accuracy'. 2014 (cited on page 62).

[113] M. Quddus, W. Ochieng, and R. Noland. 'Current map-matching algorithms for transport applications: State-of-the art and future research directions'. In: *Transportation Research Part C: Emerging Technologies* 15.5 (2007), pp. 312–328. DOI: http://dx.doi.org/10.1016/j.trc.2007.05.002 (cited on page 62).

[114] O. Mazhelis. 'Using recursive Bayesian estimation for matching GPS measurements to imperfect road network data'. In: *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. Sept. 2010, pp. 1492–1497. DOI: 10.1109/ITSC.2010.5625138 (cited on page 62).

[115] T. Tao Feng and H. Timmermans. 'Map Matching of GPS Data with Bayesian Belief Networks'. In: *Proceedings of the Eastern Asia Society for Transportation Studies*. 9. 2013 (cited on page 62).

[116] Mahdi Hashemi and Hassan A. Karimi. 'A critical review of real-time map-matching algorithms: Current issues and future directions'. In: *Computers, Environment and Urban Systems* 48 (2014), pp. 153–165. DOI: 10.1016/j.compenvurbsys.2014.07.009 (cited on page 62).

[117] Z. c. He et al. 'On-line map-matching framework for floating car data with low sampling rate in urban road networks'. In: *IET Intelligent Transport Systems* 7.4 (Dec. 2013), pp. 404–414. DOI: 10.1049/iet-its.2011.0226 (cited on page 65).

[118] S. Mattheis et al. 'Putting the car on the map: A scalable map matching system for the Open Source Community'. In: *INFORMATIK 2014: Workshop Automotive Software Engineering*. 2014 (cited on pages 65, 73).

[119] Richard Dowling. *Multimodal Level of Service Analysis for Urban Streets: Users Guide*. Tech. rep. 2009. DOI: 10.17226/23086 (cited on pages 65, 66).

[120] *OpenStreetMap contributors*. Retrieved from http://planet.openstreetmap.org. Planet dump [Data file from: 2016-01-26] (cited on pages 67, 91).

[121] George H Dunteman. *Principal components analysis*. 69. Sage, 1989 (cited on page 69).

[122] Michael Behrisch et al. 'SUMO - Simulation of Urban MObility: An overview'. In: *SIMUL 2011, The Third International Conference on Advances in System Simulation*. 2011, pp. 63–68 (cited on page 70).

[123] Tony Z. Qiu et al. 'Estimation of freeway traffic density with loop detector and probe vehicle data'. In: *Transportation Research Record* 2178 (2010), pp. 21–29. DOI: 10.3141/2178-03 (cited on page 84).

[124] Fred L. Hall. 'Traffic stream characteristics'. In: *Revised Monograph on Traffic Flow Theory* 165 (1992), pp. 2.1–2.36 (cited on page 84).

[125] P. Kachroo et al. 'Multiscale Modeling and Control Architecture for V2X Enabled Traffic Streams'. In: *IEEE Transactions on Vehicular Technology* 66.6 (June 2017), pp. 4616–4626. DOI: 10.1109/TVT.2017.2693235 (cited on page 84).

[126] Arash Shahmansoori, Gonzalo Seco-Granados, and Henk Wymeersch. 'Survey on 5G Positioning'. In: *Multi-Technology Positioning*. Ed. by Jari Nurmi et al. Cham: Springer International Publishing, 2017, pp. 165–196. DOI: 10.1007/978-3-319-50427-8_9 (cited on page 85).

[127] J. Talvitie et al. 'Novel Algorithms for High-Accuracy Joint Position and Orientation Estimation in 5G mmWave Systems'. In: *2017 IEEE Globecom Workshops (GC Wkshps)*. Dec. 2017, pp. 1–7. DOI: 10.1109/GLOCOMW.2017.8269069 (cited on page 85).

[128] A. Dammann, R. Raulefs, and S. Zhang. 'On prospects of positioning in 5G'. In: *2015 IEEE International Conference on Communication Workshop (ICCW)*. June 2015, pp. 1207–1213. DOI: 10.1109/ICCW.2015.7247342 (cited on pages 85, 95).

[129] Leo Breiman. 'Random Forests'. In: *Machine Learning* 45.1 (2001), pp. 5–32. DOI: 10.1023/A:1010933404324 (cited on page 88).

[130] Peter HA Sneath, Robert R Sokal, et al. *Numerical taxonomy. The principles and practice of numerical classification*. 1973 (cited on page 94).

[131] I. F. Akyildiz, J. M. Jornet, and C. Han. 'TeraNets: ultra-broadband communication networks in the terahertz band'. In: *IEEE Wireless Communications* 21.4 (2014), pp. 130–135 (cited on page 96).

[132] Sajjad Hussain. 'Efficient Ray-Tracing Algorithms for RadioWave Propagation in Urban Environments'. PhD thesis. Dublin, Ireland: School of Electronic Engineering, Dublin City University, Sept. 2017 (cited on page 97).

[133] M. T. Barros, R. Mullins, and S. Balasubramaniam. 'Integrated Terahertz Communication with Reflectors for 5G Small Cell Networks'. In: *IEEE Transactions on Vehicular Technology* PP.99 (2016), pp. 1–1. DOI: 10.1109/TVT.2016.2639326 (cited on pages 97, 100).

[134] ITU-R. *Propagation by Diffraction*. Draft new Report. Geneva: International Telecommunication Union, 2009 (cited on page 97).

[135] Kalyanmoy Deb. 'An introduction to genetic algorithms'. In: *Sadhana* 24.4 (Aug. 1999), pp. 293–315. DOI: 10.1007/BF02823145 (cited on page 99).

[136] Mate Boban et al. 'Impact of Vehicles as Obstacles in Vehicular Ad Hoc Networks'. In: *IEEE Journal on Selected Areas in Communications* 29.1 (2011) (cited on page 100).

[137] Fangce Guo, John W. Polak, and Rajesh Krishnan. 'Predictor fusion for short-term traffic forecasting'. In: *Transportation Research Part C: Emerging Technologies* 92 (2018), pp. 90–100. DOI: https://doi.org/10.1016/j.trc.2018.04.025 (cited on page 103).

[138] Luc Anselin. 'Under the hood: Issues in the specification and interpretation of spatial regression models'. In: *Agricultural Economics* 27.3 (2002), pp. 247–267. DOI: 10.1111/j.1574-0862.2002.tb00120.x (cited on pages 104, 105).

[139] Alireza Ermagun and David Levinson. 'An Introduction to the Network Weight Matrix'. In: *Geographical Analysis* 50.1 (2018), pp. 76–96. DOI: 10.1111/gean.12134 (cited on page 105).

[140] Esteban Fernandez-Vazquez. 'Estimating spatial weighting matrices in cross-regressive models by entropy techniques'. In: *50th Congress of the European Regional Science Association: "Sustainable RegionalGrowth and Development in the Creative Knowledge Economy"*. Jönköping, Sweden, 2010 (cited on page 105).

[141] Elżbieta Antczak. 'Building W Matrices Using Selected Geostatistical Tools: Empirical Examination and Application'. In: *Stats* 1.1 (2018), pp. 112–133. DOI: 10.3390/stats1010009 (cited on page 105).

[142] S. A. L. M. Kooijman. 'Some Remarks on the Statistical Analysis of Grids Especially with Respect to Ecology'. In: *Annals of Systems Research: Publikatie van de Systeemgroep Nederland Publication of the Netherlands Society for Systems Research*. Ed. by B. Van Rootselaar. Boston, MA: Springer US, 1976, pp. 113–132. DOI: 10.1007/978-1-4613-4243-4_6 (cited on page 105).

[143] William R. Black. 'Network Autocorrelation in Transport Network and Flow Systems'. In: *Geographical Analysis* 24.3 (1992), pp. 207–222. DOI: 10.1111/j.1538-4632.1992.tb00262.x (cited on page 106).

[144] Alireza Ermagun and David Levinson. 'An Introduction to the Network Weight Matrix'. In: *Geographical Analysis* 50.1 (2018), pp. 76–96. DOI: 10.1111/gean.12134 (cited on page 106).

[145] Alireza Ermagun and David M Levinson. 'Development and application of the network weight matrix to predict traffic flow for congested and uncongested conditions'. In: *Environment and Planning B: Urban Analytics and City Science* 46.9 (2019), pp. 1684–1705. DOI: 10.1177/2399808318763368 (cited on page 106).

[146] Alireza Ermagun and David Levinson. 'Spatiotemporal traffic forecasting: review and proposed directions'. In: *Transport Reviews* 38.6 (2018), pp. 786–814. DOI: 10.1080/01441647.2018.1442887 (cited on page 106).

[147] X. Min et al. 'Short-term traffic flow forecasting of urban network based on dynamic STARIMA model'. In: *2009 12th International IEEE Conference on Intelligent Transportation Systems*. 2009, pp. 1–6 (cited on page 106).

[148] Tao Cheng et al. 'A Dynamic Spatial Weight Matrix and Localized Space–Time Autoregressive Integrated Moving Average for Network Modeling'. In: *Geographical Analysis* 46.1 (2014), pp. 75–97. DOI: 10.1111/gean.12026 (cited on page 106).

[149] Peibo Duan et al. 'STARIMA-based traffic prediction with time-varying lags'. In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* 1 (2016), pp. 1610–1615. DOI: 10.1109/ITSC.2016.7795773 (cited on page 106).

[150] Ali Ziat et al. 'Joint prediction of road-traffic and parking occupancy over a city with representation learning'. In: *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC* (2016), pp. 725–730. DOI: 10.1109/ITSC.2016.7795634 (cited on page 106).

[151] N. Walravens et al. 'Monitoring Movement in the Smart City: Opportunities and Challenges of Measuring Urban Bustle'. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* VI-4/W2-20.October (2020), pp. 181–188. DOI: 10.5194/isprs-annals-vi-4-w2-2020-181-2020 (cited on page 125).

Notation

The next list describes some of the most significant symbols that have been used within the body of the document. Please note that some common latin alphabet letters may have been used in different chapters with different meanings.

Δ differential metric between two observations

γ simplification ratio

Φ parking occupancy at a given time instant

Π neighbour-order

Ψ traffic volume during an integration period

σ standard deviation

τ sampling period

Υ vehicle density obtained as an aggregation of multiple presence snapshots

cb_e travel time cost in backward direction

cf_e travel time cost in forward direction

D day of week

E set of directed edges

G non-planar geometric graph

$geom_e$ geometry of edge, linestring

$geom_v$ geometry of node, point

H time of day (hour from 00 to 23)

h prediction horizon

h_{tr} height of transmitter

h_o height of obstacle

l time lag between two observation samples

l_e	length of edge
L_X	width (in metres) of each cell in a regular grid
L_Y	height (in metres) of each cell in a regular grid
max_{depth}	maximum depth of Random Forest algorithm trees
n_{feat}	number of features
n_{train}	number of records for training
n_{tree}	number of trees in Random Forest algorithm
N_X	number of cells in X axis of a grid
N_Y	number of cells in X axis of a grid
pr	number of previous records
S	sliding interval
sn	snapshot matrix representation of an instantaneous number of vehicles inside the grid
T_{agg}	duration of integration period
t_{samp}	sampling resolution or sampling period
T_{tot}	total duration of available observed data
th_{limit}	threshold
V	set of nodes
v_e	nominal speed of edge

List of Acronyms

A
ADAS Advanced Driver Assistance Systems. 46
API Application Programming Interface. 12, 14, 24, 57, 58
ARIMA Auto Regressive Integrated Moving Average. 18, 20, 26, 27, 30

C
CDR Call Detail Record. 21
CSV Coma-Separated Values. 14

E
EC European Comission. 1

F
FCD Floating Car Data. xvi, xxi, 60, 62, 65, 67, 70, 72–74, 76

G
GA Genetic Algorithm. 99–102
GBFS General Bikeshare Feed Specification. 24
GIS Geographic Information System. 6, 12, 13, 40
GIS-T Geographic Information Systems in Transportation. 8
GNSS Global Navigation Satellite System. 29, 43, 54, 59, 63, 65, 72
GTFS General Transit Feed Specification. 12, 23, 24, 41

H
HDFS Hadoop File System. 13

I
IoT Internet of Things. 1, 57
ITS Intelligent Transport System. 1, 10, 45, 52

J
JSON JavaScript Object Notation. 14

K
kNN k-Nearest Neighbour. 88, 92, 93, 121
KPI Key Performance Indicator. xvi, 32, 53, 64

L
LBS Location Based Systems. 20, 54
LDM Local Dynamic Map. xvi, 45, 46, 48, 50, 52
LoS Line of Sight. xxii, 85, 96, 97, 99

LPR Label Plate Recognition. 54, 55, 60, 61
LRS Linear Referencing System. 8
LSTM Long Short Term Memory. 27

M
MaaS Mobility as a Service. 3, 31
MAPE Mean Absolute Percentage Error. 17, 91–93, 109, 110, 112
MDS Mobility Data Specification. 24
MOD Moving Objects Database. 13
MSE Mean Squared Error. 17

N
NLoS Non Line of Sight. 85, 96, 97, 99, 101, 103
NN Neural Network. 26, 27

O
OECD Organisation for Economic Cooperation and Development. 1
OGC Open Geospatial Consortium. 12, 13
OSM OpenStreetMap. 7, 28, 35, 36, 40, 48, 67, 70, 73, 91, 109

P
PCA Principal Components Analysis. 69, 73
PDF Portable Document Format. 35, 58
POI Point of Interest. 6

R
RAN Radio Access Network. 95, 96
RDBMS Relational Database Management System. 12, 13, 47
RF Random Forest. xvi, xix, xxii, 88, 89, 92–95, 99, 102, 103, 107–109, 112, 113, 115, 117, 119, 121, 122
RMSE Root Mean Square Error. xxii, 91–93, 110
ROC Receiver Operating Characteristic. 17

S
STDM Spatio-Temporal Data Mining. 15
STN Spatio-temporal statistic in Network space. 19
SUMO Simulation of Urban MObility. xxi, 29, 90, 91
SUMP Sustainable Urban Mobility Plan. xxi, 32, 33

T
TAZ Traffic Assignment Zone. 35, 41

U
UMS Urban Movement Space. xvi, 36–38, 40, 41, 43, 45–47, 52, 53, 63, 83, 123, 124
UTC Universal Time Coordinated. 49

V
V2X Vehicle-to-Everything. 7

W
WLAN Wireless Local Area Network. 20

X
XML eXtensible Markup Language. 14