NOVA SCHOOL OF
SCIENCE & TECHNOLOGY

DEPARTAMENT OF
ELECTRICAL AND COMPUTER ENGINEERING

A DATA-DRIVEN METHODOLOGY TOWARDS MOBILITY- AND TRAFFIC--RELATED BIG SPATIOTEMPORAL DATA FRAMEWORKS

PAULO ALVES FIGUEIRAS

Master in Electrical and Computer Engineering

DOCTORATE IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon

December, 2021

DEPARTAMENT OF
ELECTRICAL AND COMPUTER ENGINEERING

A DATA-DRIVEN METHODOLOGY TOWARDS MOBILITY- AND TRAFFIC-RELATED BIG SPATIOTEMPORAL DATA FRAMEWORKS

PAULO ALVES FIGUEIRAS

Master in Electrical and Computer Engineering

Adviser: Ricardo Jardim-Gonçalves
Full Professor, FCT-NOVA

Co-adviser: João Moura Pires
Associate Professor, FCT-NOVA

Examination Committee:

Chair: José Júlio Alves Alferes,
Full Professor, FCT-NOVA

Rapporteurs: Hervé Panetto,
Full Professor, Université de Lorraine
Maribel Yasmina Campos Alves Santos,
Full Professor, Universidade do Minho

Adviser: Ricardo Jardim-Gonçalves,
Full Professor, FCT-NOVA

Members: Celson Pantoja Lima,
Associate Professor, Universidade Federal do Oeste do Pará
Nenad Stojanović,
CEO, NISSATECH Innovation Center
Teresa Cristina de Freitas Gonçalves,
Associate Professor, Universidade de Évora
José Júlio Alves Alferes,
Full Professor, FCT-NOVA
Ruben Duarte Dias da Costa,
Invited Auxiliar Professor, FCT-NOVA

DOCTORATE IN ELECTRICAL AND COMPUTER ENGINEERING

NOVA University Lisbon
December, 2021

A Data-driven Methodology Towards Mobility- and Traffic-related Big Spatiotemporal Data Frameworks

“Time and space are modes by which we think and not conditions in which we live”

Albert Einstein

Acknowledgements

To properly acknowledge everyone who accompanied me in this and other challenges in my research career and my life, it is quite possible that the pages poured into this document would not suffice. Nevertheless, and due to the impossibility to rightly reference all of them, this acknowledgement goes to the ones that were directly involved in this great challenge that is to write a PhD thesis.

First and foremost, I would like to thank my supervisor, Professor Ricardo Jardim Gonçalves, PhD, for the huge support and confidence that was deposited in me, not only during the undertaking of this PhD, but also since I started my research career. On equal grounds, I would like to thank my co-supervisor, Professor João Moura Pires, PhD, for the relentless availability and precious knowledge sharing and support throughout this path.

A special acknowledgement to the great institution that harboured me since 2000 and which has become my home, the NOVA School of Science and Technology, NOVA University of Lisbon, and particularly to the Department of Electrical and Computer Engineering and to all its teaching, non-teaching and secretariat staff, with a special emphasis to Helena Inácio, who always found the best way to support me and solve any bureaucratic "challenge" related to this work.

To the research centre that welcomed me and gave me the chance to pursue my research career, the Centre for Technologies and Systems of the Institute for the Development of New Technologies (CTS-UNINOVA), and to my research colleagues, my appreciation for all the support and deposited confidence. I must leave a special acknowledgement to Rúben Costa, PhD, my eternal colleague, not only for the moments of fellowship, discussion and, sometimes, frustration we shared, but also for the friendship and trust that he always transmitted, and for all the support he has given me.

An infinite acknowledgement to my family and specially to my parents, for all they gave and taught, since without them I would not be who or where I am today, and to all my friends for the limitless friendship, patience and positive energy throughout this work and all my life. And, because sometimes the best comes last, a lifetime-worth acknowledgement to my Kacu, my wife, girlfriend and best friend, for never giving up, for putting up with me and for simply being who she is.

A final acknowledgement to the European Commission and to the Portuguese Government for their unsurmountable financial contributions toward the execution of the research projects that are at the heart of this thesis, from the Seventh and Horizon 2020 Framework Programmes to Portugal 2020, without which my research work would never been possible.

Abstract

Human population is increasing at unprecedented rates, particularly in urban areas. This increase, along with the rise of a more economically empowered middle class, brings new and complex challenges to the mobility of people within urban areas. To tackle such challenges, transportation and mobility authorities and operators are trying to adopt innovative Big Data-driven Mobility- and Traffic-related solutions. Such solutions will help decision-making processes that aim to ease the load on an already overloaded transport infrastructure. The information collected from day-to-day mobility and traffic can help to mitigate some of such mobility challenges in urban areas.

Road infrastructure and traffic management operators (RITMOs) face several limitations to effectively extract value from the exponentially growing volumes of mobility- and traffic-related Big Spatiotemporal Data (MobiTrafficBD) that are being acquired and gathered. Research about the topics of Big Data, Spatiotemporal Data and specially MobiTrafficBD is scattered, and existing literature does not offer a concrete, common methodological approach to setup, configure, deploy and use a complete Big Data-based framework to manage the lifecycle of mobility-related spatiotemporal data, mainly focused on geo-referenced time series (GRTS) and spatiotemporal events (ST Events), extract value from it and support decision-making processes of RITMOs.

This doctoral thesis proposes a data-driven, prescriptive methodological approach towards the design, development and deployment of MobiTrafficBD Frameworks focused on GRTS and ST Events. Besides a thorough literature review on Spatiotemporal Data, Big Data and the merging of these two fields through MobiTraffiBD, the methodological approach comprises a set of general characteristics, technical requirements, logical components, data flows and technological infrastructure models, as well as guidelines and best practices that aim to guide researchers, practitioners and stakeholders, such as RITMOs, throughout the design, development and deployment phases of any MobiTrafficBD Framework.

This work is intended to be a supporting methodological guide, based on widely used Reference Architectures and guidelines for Big Data, but enriched with inherent characteristics and concerns brought about by Big Spatiotemporal Data, such as in the case of GRTS and ST Events. The proposed methodology was evaluated and demonstrated in various real-world use cases that deployed MobiTrafficBD-based Data Management, Processing, Analytics and Visualisation methods, tools and technologies, under the umbrella of several research projects funded by the European Commission and the Portuguese Government.

Keywords: Big Spatiotemporal Data, Mobility- and Traffic-related data framework, data-driven methodology, geo-referenced time series, spatiotemporal events

Resumo

A população humana cresce a um ritmo sem precedentes, particularmente nas áreas urbanas. Este aumento, aliado ao robustecimento de uma classe média com maior poder económico, introduzem novos e complexos desafios na mobilidade de pessoas em áreas urbanas. Para abordar estes desafios, autoridades e operadores de transportes e mobilidade estão a adotar soluções inovadoras no domínio dos sistemas de Dados em Larga Escala nos domínios da Mobilidade e Tráfego. Estas soluções irão apoiar os processos de decisão com o intuito de libertar uma infraestrutura de estradas e transportes já sobrecarregada. A informação colecionada da mobilidade diária e da utilização da infraestrutura de estradas pode ajudar na mitigação de alguns dos desafios da mobilidade urbana.

Os operadores de gestão de trânsito e de infraestruturas de estradas (em inglês, *road infrastructure and traffic management operators — RITMOs*) estão limitados no que toca a extrair valor de um sempre crescente volume de Dados Espaciotemporais em Larga Escala no domínio da Mobilidade e Tráfego (em inglês, *Mobility- and Traffic-related Big Spatiotemporal Data — MobiTrafficBD*) que estão a ser colecionados e recolhidos. Os trabalhos de investigação sobre os tópicos de Big Data, Dados Espaciotemporais e, especialmente, de *MobiTrafficBD*, estão dispersos, e a literatura existente não oferece uma metodologia comum e concreta para preparar, configurar, implementar e usar uma plataforma (*framework*) baseada em tecnologias *Big Data* para gerir o ciclo de vida de dados espaciotemporais em larga escala, com ênfase nas série temporais georreferenciadas (em inglês, *geo-referenced time series — GRTS*) e eventos espaciotemporais (em inglês, *spatiotemporal events — ST Events*), extrair valor destes dados e apoiar os *RITMOs* nos seus processos de decisão.

Esta dissertação doutoral propõe uma metodologia prescritiva orientada a dados, para o *design*, desenvolvimento e implementação de plataformas de *MobiTrafficBD*, focadas em *GRTS* e *ST Events*. Além de uma revisão de literatura completa nas áreas de Dados Espaciotemporais, *Big Data* e na junção destas áreas através do conceito de *MobiTrafficBD*, a metodologia proposta contem um conjunto de características gerais, requisitos técnicos, componentes lógicos, fluxos de dados e modelos de infraestrutura tecnológica, bem como diretrizes e boas práticas para investigadores, profissionais e outras partes interessadas, como *RITMOs*, com o objetivo de guiá-los pelas fases de *design*, desenvolvimento e implementação de qualquer plataforma *MobiTrafficBD*.

Este trabalho deve ser visto como um guia metodológico de suporte, baseado em Arquiteturas de Referência e diretrizes amplamente utilizadas, mas enriquecido com as características e assuntos implícitos relacionados com Dados Espaciotemporais em Larga Escala, como no caso de *GRTS* e *ST Events*. A metodologia proposta foi avaliada e demonstrada em vários cenários reais no âmbito de projetos de investigação financiados pela Comissão Europeia e pelo Governo português, nos quais foram implementados métodos, ferramentas e tecnologias nas áreas de Gestão de Dados, Processamento de Dados e Ciência e Visualização de Dados em plataformas *MobiTrafficBD*,

Palavas chave: Dados Espaciotemporais em Larga Escala, Plataforma de dados relacionados com Mobilidade e Tráfego, Metodologia orientada a dados, Series Temporais Georreferenciadas, Eventos Espaciotemporais

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acronyms

AI	Artificial Intelligence
ANPR	Automatic Number Plate Recognition
API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
BDVA	Big Data Value Association
BI	Business Intelligence
BRIN	Block Range Index
BSON	Binary JSON
CCTV	Closed Circuit Television
CEP	Complex Event Processing
CPU	Central Processing Unit
CRISP-DM	Cross Industry Standard Process for Data Mining
CSV	Comma Separated Values
CUDA	Compute Unified Device Architecture
DATEX	Data Exchange
DM	Data Mining
DSRM	Design Science Research Methodology
DST	Data Science Trajectory
EC	European Commission

ELT	Extract-Load-Transform
ELTL	Extract-Load-Transform-Load
ETL	Extract-Transform-Load
ETSI	European Technical Specification
EU	European Union
FCT	*Faculdade de Ciências e Tecnologia*
FP7	Framework Programme 7
FTP	File Transfer Protocol
GIS	Geographic Information Systems
GPS	Global Positioning System
GPU	Graphics Processing Unit
GRTS	Geo-referenced Time Series
GTFS	General Transit Feed Specification
GUI	Graphical User Interface
HCE	Host Card Emulation
HDFS	Hadoop Distributed File System
HTTP	Hypertext Transfer Protocol
ICT	Information and Communication Technologies
ID	Identification
IDC	International Data Corporation
IEEE	Institute of Electrical and Electronic Engineers
IP	*Infraestruturas de Portugal*
ISI	International Statistical Institute
ISO	International Standards Organization
ITS	Intelligent Transportation Systems
JMS	Java Message Service
JSON	JavaScript Object Notation
KDD	Knowledge Discovery in Databases

LPP	*Ljubljanski Potniški Promet*
LTSM	Low-Term Short Memory
MAP	Multi-Dimensional Aggregation Pyramid
MB	Mega Bytes
MIDAS	Motorway Incident and Automatic Signalling
ML	Machine Learning
MNO	Mobile Network Operators
MobiTrafficBD	Mobility- and Traffic-related Big Spatiotemporal Data
MPP	Massively Parallel Processing
NER	Name Entity Recognition
NERD	Name Entity Recognition and Disambiguation
NIST	National Institute of Standards and Technology
NTIS	National Traffic Information System
ODBC	Open Database Connectivity
OECD	Organisation for Economic Co-operation and Development
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
OODA	Observe, Orient, Decide, Act
OSM	Open Street Map
OTLIS	*Associação de Operadores de Transportes de Lisboa*
POS	Part of Speech
RM	Reference Model
RDBMS	Relational Database Management System
RDD	Resilient Distributed Datasets
REST	Representational State Transfer
RFID	Radiofrequency Identification
RITMOs	Road Infrastructure and Traffic Management Operators
RUN	*Repositório da Universidade Nova de Lisboa*

SARIMA	Seasonal Autoregressive Integrated Moving Average
SEMMA	Sample, Explore, Modify, Model, Assess
SOLAP	Spatial Online Analytical Processing
SQL	Structured Query Language
ST Event	Spatiotemporal Event
SVM	Support Vector Machine
TCP/IP	Transmission Control Protocol/Internet Protocol
TMU	Traffic Management Unit
UI	User Interface
UK	United Kingdom
UML	Unified Modelling Language
UN	United Nations
UNL	*Universidade Nova de Lisboa*
URL	Uniform Resource Locator
USA	United States of America
UX	User Experience
WGS	World Geodetic System
XML	Extensible Markup Language

1

INTRODUCTION

This chapter presents the context and motivation of this doctoral thesis, the research challenges, opportunities and goals, the expected scientific and technical contributions, and the structure of this document. Furthermore, this chapter describes the research methodology that supports this thesis and the relationship between its steps and the expected results and the overall contribution of relevant research endeavours for the development of the presented work.

1.1 Context

Human population reached 7.5 billion in 2017 and it is expected to rise to 10 billion by 2050, from which 66% will live in cities or urban areas (United Nations, Department of Economic and Social Affairs, Population Division, 2014). Urban areas with more than 10 million inhabitants are expected to grow from 28, in 2017, to 41 by 2030 (United Nations, Department of Economic and Social Affairs, Population Division, 2014). Many social and economic areas will be greatly affected by this growth, but perhaps one of the most affected areas will be mobility, especially urban mobility. One of the major concerns of the United Nations (UN) of the impact of this exponential migration to cities is its management regarding spatial distribution and mobility of the population (United Nations, Department of Economic and Social Affairs, Population Division, 2014). Mobility may be defined as the ease of movement from one location to another with the help of transport networks and services available within and between the two locations (Beimborn, Horowitz, Vijayan, & Bordewin, 1999).

More than the growth of population, the rise of middle-class population in cities is also challenging, when considering that more than 2 billion people will likely enter the middle-class demographic group, namely in cities in emerging markets, such as China or India. The new middle class will want to buy their own cars, with an expected growth in automobile sales of 70 million in 2010 to 125 million in 2025 (Goldman Sachs, 2017). Some even argue that today's global car fleet could double by 2030, mainly in cities (Dargay, Gately, & Sommer, 2007). The existing urban infrastructure cannot support such an increase in vehicles. Congestion is already close to unbearable in many cities and can cost as much as 2 to 4 percent of countries' Gross Domestic Product, caused by lost time in traffic, fuel waste and increased costs. Moreover, the environmental hazards that are linked with congestion are a tough reality. The World Health Organization estimated in 2014 that seven million premature deaths are attributable to air pollution, and a significant share is the result of urban transit (World Health Organization, 2014).

Therefore, urban productivity and lifestyle are highly dependent on the efficiency of its transport system to move labour, consumers and freight between multiple origins and destinations. But what does this mean for individual commuters? Among the most notable urban transport challenges are traffic congestion, parking difficulties, longer commuting times, public transport inadequacy and lack of efficiency, difficulties for non-motorized transport, loss of public space, high infrastructure maintenance costs, environmental impacts and increase in energy consumption, more accidents and less safety, to name a few (Rodrigue, 2017).

So, with all the above in mind, what are the main solutions, presently available or future prospected, to the evolution of mobility, particularly urban mobility? Besides the implementation of new policies regarding infrastructure and space management, urban design, public transport optimization (e.g., last mile public transportation schemes) or fuelled automobile restrictions (e.g., diesel-based vehicle ban in major cities), which will be further pushed forward during and after the COVID-19 pandemic situation lived across the world (McKinsey & Company, 2020), Information and Communication Technologies (ICT) solutions, supported by novel sensing capabilities, are being developed and implemented by cities worldwide.

In fact, the introduction and diffusion of ICT, and the emergence of an information society, resulted in several economic and social impacts, notably for activities depending on information processing. This has impacted both the service and manufacturing sectors. Transportation is a service that requires and processes large amounts of data. For instance, transportation users make decisions about where and when to travel, which mode to use and, if they are operating their own vehicle, which route to take. Inversely, transportation services' providers must manage their assets so that they effectively match the demand of data by various transportation markets (Rodrigue, 2017).

Some examples of such ICT-supported solutions are electric and autonomous vehicles, vehicle-to-vehicle and vehicle-to-infrastructure communication, electronic toll collection or automatic road enforcement, for instance. These constitute applications of a new paradigm coined Intelligent Transportation Systems (ITS). ITS is a research field in the domain of ICT which addresses problems on the transportation sector and the daily journeys made by citizens. Specifically, "Intelligent Transport Systems (ITS) refers to the integration of ICT with vehicles and transport infrastructure to improve economic performance, safety, mobility and environmental sustainability for the benefit of all citizens" (European Telecommunications Standards Institute, 2012).

Ultimately, the main goal of ITS is to enable optimization and to promote the efficiency of mobility services, by providing added value through processing and analysis of mobility-related data. This goal is tackled by solving data-related problems, coping with the spatiotemporal aspects of mobility-related information and creating insights and new knowledge from such data, by applying analysis, correlation, event and anomaly detection and enhanced visualization techniques. The application of the above analyses and optimization procedures is seen as part of Machine Learning and Data Mining research fields. The goal of Machine Learning and Data Mining is to deliver a set of tools and methods to extract automatically or semi-automatically relevant and easy-to-understand insights, such as rules, patterns, irregularities, or associations, based on the characteristics and interactions between the data being analysed (Pujari, 2001).

Hence, the main concerns involved in such analysis and optimization processes comprise the whole lifecycle of data, from its collection, cleaning, transformation and storage to its processing, analysis and visualization. For instance, how to get quality data in enough quantities to realize the mobility of the future? The acceleration in both the volume and speed of exploitable data will have a significant and disruptive impact in transportation (OECD/ITF, 2015). Massive amounts of mobility data must be collected in real-time, in order to have an up-to-date view of mobility.

The collection of large datasets – coined as "Big Data" – is not a new concept and is not part of a single technological development. The collection of great volumes of data is supported by new data gathering mechanisms on ubiquitous devices, better storage capabilities, enhanced computing power and novel sensing and communication technologies (Monino, 2021).

Mobility data is often divided into three main categories: Moving object data, i.e., the movement of objects (e.g., people, vehicles) in space over a period of time, non-moving object data, which corresponds to data captured throughout time in a specific location, such as, for instance, road sensor data, and event data, defined by the collection of mobility data variables on specific points in space and time, such as in the case of traffic events (Andrienko & Andrienko, 2009). Therefore, ITS should be able to efficiently collect and store mobility-related Big Data, considering its high-volume, high-speed characteristics.

Moreover, in order to have a good understanding of mobility within cities, ranging from traffic information to individual commuters' routes, infrastructure status or public transport demands, reliable data is needed. Data may come from sensors, both in-vehicle or in-infrastructure, Global Positioning Systems (GPS) tracking devices, public transit routes, times, delays, etc., or even users' mobility behaviours and tendencies. Some of these datasets have a well-defined purpose and are collected to address well-defined questions or to resolve specific tasks. However, the potential value of mobility data lies in the combination of different, heterogeneous data sources (Silveira, de Almeida, Marques-Neto, Sarraute, & Ziviani, 2016; Peixoto & Moreira, 2013). The issue is that mobility data is like "digital dust" that is gathered from humans' interactions with a panoply of computing systems or services and digital infrastructure. When effectively combined and merged, these data streams may help revealing unsuspected or unobserved patterns and insights in day-to-day mobility that can be used in benefit of all (OECD/ITF, 2015). ITS should cope with this heterogeneity aspect of mobility data, by introducing effective data standards and common formats for mobility and interoperability services, to merge and harmonize data coming from different data sources.

There are several compelling cases on the value brought forward by Big Data, data mining and visual analytics solutions in ITS for urban planning (Chen & Englund, 2016), intelligent and connected transport (Jia & Ngoduy, 2016) and better safety (González, Pérez, Milanés, & Nashashibi, 2016). However, there is a big question mark on the ability for relevant stakeholders, such as road infrastructure and traffic management operators, to keep the pace with the proliferation of newly available data and with the tools to efficiently analyse it. Big Data, data mining and visual analytics are promising research fields for improving planning, management and decision-making processes related to transport, by applying new data analysis and analytics frameworks (OECD/ITF, 2015). These advancements in large-scale data analysis are the best option to extract insights and knowledge from mobility data in a timely manner and to, ultimately, support decision-making processes. Hence, ITS should make use of efficient Big Data, data mining and visual analytics tools and technologies over large-scale mobility data to support the decision-making processes of relevant stakeholders.

But maybe the biggest issue of spatiotemporal data has to do with its inherent characteristics, which make it different from information originating in other areas and domains: its space-time features. The space-time dimension of mobility data introduces direct consequences in the ways such data is collected, stored, processed, analysed, correlated or visualized (Atluri, Karpatne, & Kumar, 2018). All these inherent characteristics of spatiotemporal mobility data should be considered when collecting, harmonizing, processing and visualizing mobility data.

In the literature, spatiotemporal data often refers to data that presents variability and dynamicity in both time and space (Heuvelink, Pebesma, & Gräler, 2017). But, spatiotemporal data may also define any data that includes a static or dynamic representation in both time and space, i.e., the measured variables are represented in both space and time, but these dimensions may or may not be static (have the same value across readings/measurements), such as presented in (Atluri, Karpatne, & Kumar, 2018), (Shekhar, Zhang, & Huang, 2010), (Pebesma, 2012) and (Kisilevich, Mansmann, Nanni, & Rinzivillo, 2009), for instance. For the sake of the presented work, the different types of spatiotemporal data that will be considered are defined by (Kisilevich, Mansmann, Nanni, & Rinzivillo, 2009), and will be further explored in Chapter 2. For now, it is worth to highlight that, although all spatiotemporal data types may be used for the purpose of managing traffic and road infrastructure, most public operators for road infrastructure and traffic management use road sensors' (geo-referenced time series) and traffic events' (spatiotemporal events) data to accomplish this goal.

Road sensors are relatively cheap and can be placed in prioritized, carefully selected positions to produce more accurate data on the different arteries of the road network, whereas traffic events' data is easy to capture by local authorities, through gathering of real-time information about events that happen in the road infrastructure by police authorities or road infrastructure operators. Hence, such data types enable a more comprehensive view of mobility, by providing an abstraction from individual drivers to an overall mobility panorama in specific points of interest within an urban area or city.

Hence, for the sake of this thesis work, the focus will fall upon spatiotemporal event (ST Event) and geo-referenced time series (GRTS) data, since these are directly linked with traffic-related data that do not fall into the categories of moving objects, presenting some advantages, in terms of data lifecycle management and analysis, due to the fact that they are considered spatially static objects, i.e., their location does not change across time. A ST event is often characterized by fixed location and fixed time, representing where and when the event happened, whereas GRTS data is composed by measurements of continuous or discrete spatiotemporal fields, recorded at fixed locations in space.

1.2 Motivation and Research Goals

Road infrastructure and traffic management operators (RITMOs) face several limitations to effectively extract value from the exponentially growing volumes of spatiotemporal data that are being acquired and gathered. The cheer volume and increased heterogeneity of collected data from traffic sensors and events, in the form of GRTS and ST events, undermines the process of answering important questions about day-to-day mobility in a timely manner, so as to support data-driven decision-making processes of these stakeholders. RITMOs need to be aware that traditional data lifecycle management strategies, such as conventional data warehousing systems or geographic information systems, are unable to solve all these issues and that new strategies and technologies must be employed to address the specificities of both Big Data and spatiotemporal characteristics, present in mobility data.

The literature does not offer a concrete methodological approach to design, implement and deploy a complete Big Data-based approach to manage the lifecycle of mobility-related spatiotemporal data, extract value from it and support decision-making processes of RITMOs. Research about these topics is scattered and there is no common approach to design, implement and deploy a generic, fully fledged solution to collect, store, process, analyse and visualize mobility- and traffic-related Big spatiotemporal Data (throughout the document, the abbreviation MobiTrafficBD will be used; the "spatiotemporal" dimension of data is already comprised in data coming from the Mobility and Traffic domains), in the form of ST Events and GRTS. Most works focus on specific use cases and consist of finding the best Big Data technology and/or spatiotemporal data model depending on use case-specific requirements, instead of a data-driven approach (Clegg, 2015) (e.g., (Anbaroglu, Heydecker, & Cheng, 2014; Wu, Zurita-Milla, & Kraak, 2015; Reich & Porter, 2015)).

Furthermore, several works point the way to some guidelines, best practices and implementations in specific contexts, but these only partially cover the aspects and characteristics of both Big Data and spatiotemporal data lifecycle management identified in the literature. This issue is a consequence of the multidisciplinary nature of these research areas, as works often present approaches on general guidelines and best practices for one research discipline but not for the others (MacLeod, 2018) (e.g., Big Data best practices (Marz & Warren, 2015)), while other works focus on particular technological advances in one specific area, such as Big Data collection [26] or analysis (Nallaperuma, et al., 2019), spatiotemporal models (Cheng, 2016; Wang, et al., 2017) or visualization (Surkhovetskyy, Andrienko, Andrienko, & Fuchs, 2017; Gatalsky, Andrienko, & Andrienko, 2004), to name a few examples. There is no integrated approach focusing on all the layers, logical and physical, that are needed to implement and deploy a solution as a whole that copes with the MobiTrafficBD lifecycle along with the correct evaluation measures (e.g., benchmarking, data modelling, deployment evaluation, etc.) which would provide a general-purpose data-driven approach. Such approach will enable the prescription of models and methods as well as best practices and guidelines to researchers and practitioners, so that they can effectively implement and deploy solutions that fit different needs and scenarios, opting for a data-driven approach, rather than a use case-specific one.

Hence, the main gap in the current state-of-the-art is the lack of definition of a prescriptive, methodological approach that describes in depth how a Big Data-based solution for the lifecycle management and analysis of MobiTrafficBD should be designed, implemented and deployed, in contrast with the existing approaches that RITMOs have at their disposal, in terms of MobiTrafficBD processing and analysis.

The proposal of a prescriptive methodology will provide not only the set of steps towards a generic data-driven framework for MobiTrafficBD lifecycle management and analysis in a rigorously justified manner, but also the non-functional requirements that are crucial for the envisaged framework, such as setup time, easy deployment, benchmarking and validation. Such methodology enables models (representations of logical and infrastructural components), methods (structured practices), and instantiations (prototypes or implemented systems) that are tightly coupled and grounded on evaluated practices. Hence, the following research question and hypothesis are proposed:

Q.: *How can a methodology enable easy design, development and deployment of generic data-driven approaches for the lifecycle management and analysis of MobiTrafficBD, in order to help RITMOs to extract value from such data to support their decision-making processes?*

H.: *A prescriptive and data-driven methodology for the design, development and deployment of MobiTrafficBD Frameworks that comprises logical components, data flows and technological infrastructure models, along with guidelines, may ease and accelerate lifecycle management and value extraction necessary to support RITMOs' decision-making processes.*

The main goal of this doctoral thesis is the proposal of prescriptive, data-driven methodology to design, implement and deploy Big Data-based frameworks for the management and analysis MobiTrafficBD, in the form of GRTS and ST events, which take into account the characteristics of both Big Data and spatiotemporal data, providing prescriptive models, methods and infrastructural components to manage these complex data assets, in the form of a structured practical guide to practitioners and stakeholders. In this context, data-driven means that, instead of being use case-driven, i.e., the solution is built solely according to particular use cases' specifications, a data-driven approach abstracts from use cases and focuses on the data itself, in this case GRTS and ST Events, creating a generic approach that can be applied to different use cases that provide these types of data, regardless of format or schema and spatial and temporal distribution of the available data sources.

The proposed methodological approach is foreseen to be used in the following situations: relevant stakeholders do not currently have a solution for MobiTrafficBD management and analysis and want to deploy one; relevant stakeholders already have some form of legacy solution for MobiTrafficBD management and analysis (e.g. traditional Data Warehousing, Geographic Information Systems, etc.) but need to replace it; or relevant stakeholders have a use case-driven solution, relying on a non-interoperable integration of technologies and models, and prefer to have a data-driven approach, based on highly interoperable components and well defined models and methods for MobiTrafficBD management and analysis.

Furthermore, the proposed approach presents itself as a valuable contribution to the scientific community and to practitioners in the areas of ITS, spatiotemporal data analysis and Big Data for Mobility by providing tightly coupled and scientifically evaluated models, methods and instantiations. With all the above in mind, the research objectives of the presented work are:

- A prescriptive, step-by-step, data-driven methodology that will enable the creation of models and methods for MobiTrafficBD management and analysis will be presented. This methodology will provide the basis for the development and deployment of frameworks that will encompass the entire lifecycle of MobiTrafficBD, from data collection, harmonization and storage to processing, analysis and visualization, by providing guidelines and best practices that are independent of the use cases at hand. Nevertheless, in some cases, the ultimate choice of method or technology will be based on the use case, without affecting the overall data-driven philosophy of the framework itself, as will be noted throughout the document. Such guidelines and best practices will guide stakeholders through the decision process around the choice of the most suitable data models, methods and algorithms to handle, process and analyse MobiTrafficBD, and towards the selection and deployment of Big Data technologies.
- Validation and evaluation of the models and methods through the application in several demonstration scenarios and comparison with known benchmarks: The validation of the methodology will be based on several real-world use case scenarios that present similar MobiTrafficBD sources, but have different formats and contexts, as well as different overall goals. Furthermore, the methodology will be evaluated on other scenarios that are not fully related to traffic management and monitoring, such as in the case of public transportation use cases. Moreover, requirements for each of the use cases will help in the benchmarking and evaluation process that will validate that the methodology is generic enough to be applied in different MobiTrafficBD-related scenarios but is specific enough to support these scenarios to achieve their goals, while conforming with the requirements of each of them.

Regarding the proposal of models and methods the methodology offers:

1. A model for logical components and layers using general guidelines and best practices defined by several Big Data institutes and standards, such as the Big Data Value

Association (BDVA) Strategic Research Agenda and Reference Architecture (Big Data Value Association, 2020), and apply it to the Mobility and Traffic Management sector. The model further represents how MobiTrafficBD, namely GRTS and ST events, flow through the different components.

2. Guidelines and best practices for collecting, harmonizing and enriching MobiTrafficBD, in the form of GRTS and ST events, providing methods to collect data from different sources and with different formats, and accounting for the Big Data characteristics of the collected data. These methods also provide ways for new mobility-related GRTS and ST event data sources and formats to be added, enabling new deployment scenarios.
3. A technological infrastructure model, defining the configuration, organization, deployment and usage of Big Data technologies in a shared-nothing architecture for the specific case of MobiTrafficBD frameworks.
4. Data modelling guidelines that enable the representation and description of GRTS and ST event data types related to traffic and mobility, regardless of their structure and format. Such modelling methods will be part of the presented use cases and are supported by relevant mobility data standards, such as DATEX II (Easyway, 2011), enabling direct mapping between raw data formats and standardized formats.
5. Guidelines and best practices on the choice of Data Analytics and Visualization methods and tools according to the data and the use case at hand.

In terms of the evaluation of the models and methods through the application in several demonstration scenarios, the methodology aims to:

1. Evaluate the suitability of the proposed approach when applied to several real-world scenarios concerning RITMOs. This objective focuses on the validation of the methodological approach across four concrete traffic management use cases with different requirements, data sources, data formats and deployment needs, by providing a set of data models, methods and guidelines for the application of a Big Data-based framework for MobiTrafficBD lifecycle management and analysis.
2. Demonstrate the suitability of the proposed approach for solving real-world problems in other related sectors, such as in the case of public transport operators' data analysis requirements, through the application of the methodology in one concrete use case focusing on the public transportation sector.
3. Provide an informed guide for practitioners and researchers, in the form of practical examples of the employment of the overall methodology to design and develop MobiTrafficBD frameworks, derived from the above use cases, by presenting a thorough

description of the use cases, their objectives and the choices taken in each one of them, regarding the entire lifecycle of the data available for each specific use case.

It is worth mentioning that, throughout this doctoral thesis, the adaptation, customization or creation of new Big Data technologies, machine learning algorithms or data quality methods are not seen as contributions. The main goal is to propose a cohesive and prescriptive methodology to build generic, data-driven MobiTrafficBD Frameworks for lifecycle management and analysis (within the document, these frameworks will be abbreviated to MobiTrafficBD Frameworks: Mobility- and Traffic-related Big Spatiotemporal Data Frameworks for data lifecycle management and analysis) and evaluate their usability through demonstration cases that will use Big Data technologies and machine learning algorithms already developed.

Moreover, in the context of the presented work, some concerns should be considered. First, although the title and goal of this work, i.e., the methodology to develop and deploy MobiTrafficBD Frameworks, does not point to ITS directly, it is strongly related with ITS, since these frameworks are considered ITS: the use of ICT technologies and paradigms to tackle challenges in the Mobility and Transportation domains. Thus, a MobiTrafficBD Framework is a fully-fledged ITS instantiation. Effectively, the title for the presented work, "A Prescriptive Methodology for the Design and Implementation of Mobility- and Traffic-related Big Spatiotemporal Data Frameworks ", can be decomposed in the following concepts:

- *Prescriptive methodology:* A set of step-by-step methods, procedures, guidelines, best-practices and useful directions to design and develop a data framework.
- *Data Frameworks:* In the context of the presented work, a data framework is an instantiation of a set of components to achieve specific data-driven objectives. Components may be software components (e.g., databases, processing engines, visualization tools, etc.), hardware components (e.g., distributed environments, cloud environments, etc.), conceptual components (e.g., modules in a modular architecture). Data-driven objectives are any objective accomplished by means of ICT-based systems that focus on the data at hand to tackle the specific objective (e.g., data lifecycle management, decision support, business insights and understanding, advantage over competitors, etc.). The word "frameworks" is in the plural form since the methodology will not focus on a particular type of framework; rather, it will strive to be generic and modular enough to encompass several types of data-driven frameworks, from simple data lifecycle management frameworks, encompassing data collection, harmonization, cleaning, curation and storage tasks, to fully-fledged data-driven frameworks that manage all stages of the data lifecycle and comprise Data Analytics, Visual Analytics and Decision

Support tasks. The idea is that the methodology is used to support the design and implementation of one framework at a time!

- *Design and implementation:* The methodology will focus on both the design phase, covering design considerations' discussion, requirements elicitation and analysis, conceptual architecture definition and initial technology surveys, and the development phase, comprising the final choice of technologies, data models, methods, algorithms, and standards and the development and deployment of the final system.
- *Mobility and Traffic:* The methodology will focus on the design and implementation of ICT-based frameworks that target the Mobility and Traffic Management research areas. Thus, the methodology enables the creation of ITS frameworks.
- *Big Spatiotemporal Data:* The main goal of the methodology is to guide researchers and practitioners in the design and development of data-driven frameworks that can tackle the lifecycle management and analysis of Big Data sets with spatiotemporal characteristics, under the scope of Mobility and Traffic domains.

Second, this relation between the frameworks and ITS brings yet another misconception to the discussion: the distinction between architecture and framework. Although this work proposes a conceptual architecture for the development of MobiTrafficBD Frameworks, in the form of a model of logical components and data flows (Chapter 3), it is based on software architectures developed for more generic purposes and domains, such as the NIST (NBD-PWG, 2015) and BDVA (Big Data Value Association, 2020) reference architectures for Big Data. So, MobiTrafficBD Frameworks can be considered, under the scope of the presented work, as ITS instantiations of existing Big Data, Data Warehousing and other generic domains' reference architectures, taking full advantage of guidelines and best-practices extracted from these generic architectures and repurposed to cope with the specificities of ITS, besides ITS-specific guidelines and best-practices collected throughout the author's work on several research projects in the ITS area.

Hence, the presented prescriptive methodology is intended to be used and, perhaps, further improved in future research endeavours, as it aims to represent one step forward towards the creation of easy-to-design, easy-to-develop, easy-to-deploy MobiTrafficBD Frameworks for research and academic applications and, ultimately and essentially, a guide for other researchers and practitioners that, like the author, struggle to find the best ways to handle, manage and extract value from high-volume, high-speed, high-variety MobiTrafficBD sources.

1.3 Research Methodology and Context of Work

This dissertation is supported by a generic research methodology, based on Design Science Research, which is used as the means to build a proof of concept. Since this dissertation is encompassed within the Computer Science and information Systems domain, the selected methodology was the Design Science Research Methodology (DSRM) for Information Systems, proposed in (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007), represented in Figure 1.1. Such methodology is suitable for Computer Science research, and comprises six stages, which are also suitable for the objectives and expected results.

The methodology starts by identifying the problem at hand and the motivations behind it, although it is designed to be flexible and can be applied to any research project in any of the phases. Also, the methodology provides retroactive connections, in order to allow several types of process iterations. In this case, research starts by identifying the problem and motivations. Instantiating this dissertation with the DSRM and considering the time limits for each intermediate phase, the following tasks are expected as outcome:

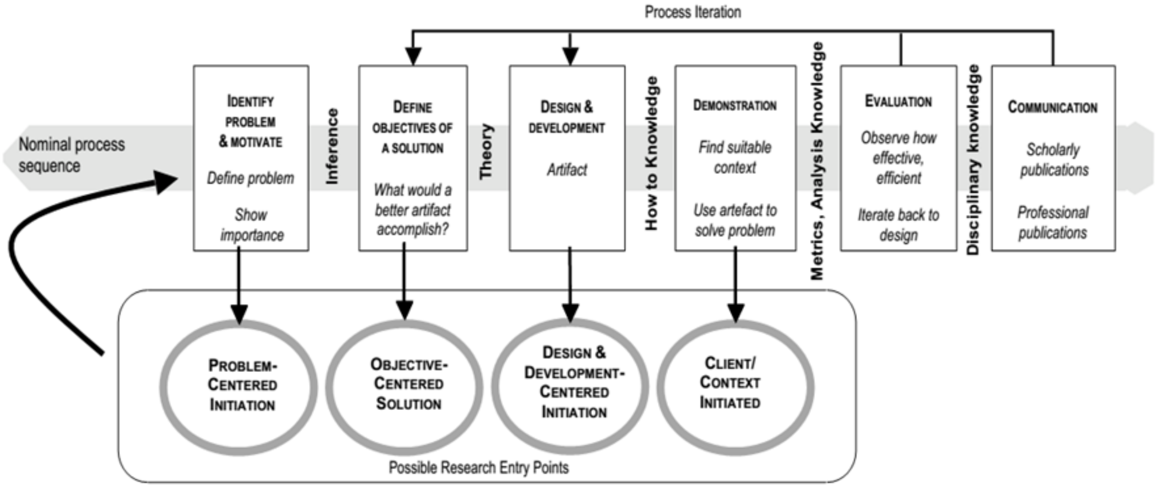

Process Iteration
Nominal process sequence
IDENTIFY PROBLEM & MOTIVATE
Define problem
Show importance
Inference
DEFINE OBJECTIVES OF A SOLUTION
What would a better artifact accomplish?
Theory
DESIGN & DEVELOPMENT
Artifact
How to Knowledge
DEMONSTRATION
Find suitable context
Use artefact to solve problem
Metrics, Analysis Knowledge
EVALUATION
Observe how effective, efficient
Iterate back to design
Disciplinary knowledge
COMMUNICATION
Scholarly publications
Professional publications
PROBLEM-CENTERED INITIATION
OBJECTIVE-CENTERED SOLUTION
DESIGN & DEVELOPMENT-CENTERED INITIATION
CLIENT/ CONTEXT INITIATED
Possible Research Entry Points

Figure 1.1 — Design Science Research Methodology (DSRM) for Information Systems (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007)

- Problem and motivation identification: identify the specific research problem and justify the value of the resulting solution. The first step towards successful scientific research consists in choosing a meaningful topic. As already explained, there is a significant lack in the literature regarding a prescriptive, data-driven methodology to design, implement and deploy Big Data-based frameworks for mobility-related spatio-temporal data management and analysis. The main problem and motivation arise from this gap and will help researchers and practitioners in the Mobility and Traffic management areas.

- Define solution outcomes: define the outcomes to achieve to solve the problem and considering the knowledge of what is doable and achievable. This task comprises the formulation of the hypothesis. It should be stated in a declarative format, which brings clarity, specificity and focus to a research problem. In the context of this doctoral thesis, the outcomes are presented in Section 1.2 and aim at delivering a prescriptive, data-driven methodology towards the design and development of generic MobiTrafficBD Frameworks that can aid RITMOs in the MobiTrafficBD (GRTS and ST Events) lifecycle management and value-extraction processes.
- Design and implementation: This task encompasses the design and development of a proof-of-concept. The proof-of-concept is frequently related to engineering research and the development of a prototype. It is the evidence that demonstrates that an idea is feasible. This way, and because many times the complete validation of the hypothesis in a real-world environment involves resources (both time and money) that few have access to, this step relates directly to the design of an experiment in a controlled environment. In the scope of the presented work, this task corresponds to the development of the data-driven, prescriptive methodology, along with all its components, challenges, guidelines and best practices.
- Testing and validation: This is the step where the testing of the hypothesis will be done. It includes the validation of the proof-of-concept and execution of tests according to the pre-defined validation method. Considerations regarding the implementations and ultimately the hypothesis will be drawn. At this stage, the researcher may find evidence that the hypothesis needs to be reformulated, thus it will need to jump back to step 4 or might need to propose adaptations to the prototype design (previous step). This task corresponds to the validation and demonstration scenarios that will serve as examples of the application of the presented methodological approach.
- Demonstration and dissemination: If the hypothesis is validated, it is a corroborated hypothesis, and can be published. It is mandatory to publish final findings and provide peers from the scientific community the chance to verify, comment, and use the developed work. Nonetheless, and despite appearing only as the final step of the adopted research method, intermediate findings can also be published.

This methodology was used throughout the research phases of the proposed work, as it was developed and applied to several European Commission- and Portuguese Government-funded research projects, which is, *per se*, a form of peer validation, as well as a way for proof-of-concepts to be tested and validated by industrial/commercial stakeholders in real world scenarios. This is an important step, since it is crucial for future technology transfer from research to industry. Hence, this PhD work has contributed for (thus being validated in):

- The European Commission's (EC) Seventh Framework Program (FP7) MobiS (Personalized Mobility Services for energy efficiency and security through advanced Artificial Intelligence techniques) research project (Grant Agreement 318452) (European Commission, 2012), funded by the EC from October 2012 to October 2015, focusing on creating a new concept and solution of a federated, customized and intelligent mobility platform by applying novel Future Internet technologies and Artificial Intelligence methods that monitor, model and manage the urban mobility complex network of people, objects, natural, social and business environment in real-time.
- The scientific research project Horizon 2020 OPTIMUM project (Grant Agreement 636160) (European Commission, 2015), funded by the EC from January 2015 to August 2018, which aims at unveiling state-of-the-art information technology solutions to improve transit, freight transportation and traffic connectivity throughout Europe. Through tailor-made applications, OPTIMUM strives to bring proactive and problem-free mobility to modern transport systems by introducing and promoting interoperability, adaptability and dynamicity. OPTIMUM establishes largely scalable architecture for the management and processing of multisource big data, which enables the continuous monitoring of transportation system needs while facilitating proactive decisions and actions in a semi-automated way.
- The national research P2020 Mobile Security Ticketing project (Grant LISBOA-01-0247-FEDER-011388), funded by the Portuguese Government, with the objective of developing an alternative ticketing support solution, based on contactless technology present in smartphones. This project also aims at delivering Big Data analytics solutions for traffic management and public transport operators.

All the above research projects joined relevant European and national stakeholders from the Traffic and Mobility domain as final end-users. Some examples are national road infrastructure operators (e.g., Infraestruturas de Portugal[1], the Portuguese road infrastructure operator, an end-user in OPTIMUM, and Trafikverket[2], the Swedish road infrastructure authority, an end-user in MobiS) or public transportation operators (e.g., OTLIS[3], the Lisbon's public transportation operators association, an end-user in Mobile Security Ticketing, and LPP[4],

[1] https://www.infraestruturasdeportugal.pt/

[2] https://www.trafikverket.se/en

[3] https://www.portalviva.pt/pt/homepage/sobre-a-otlis/a-otlis.aspx

[4] https://www.lpp.si/en

Ljubljana's public transportation operator, an end-user in OPTIMUM) , just to name a few. Hence, the work done under the scope of these projects was validated, demonstrated and put into use as proofs-of-concept in real-world scenarios, supported by relevant stakeholders. Throughout the 8-year research work performed by the author under the scope of the aforementioned projects, there was always a necessity to benchmark the most suitable architectures, methodologies, methods, tools and technologies for handling and capitalizing on both the input data sources and the underlying use cases originated from the projects' end-user scenarios, and to build roadmaps for the creation of MobiTrafficBD Frameworks that could provide the required functionalities to tackle the end-user scenarios' challenges.

Thus, the benchmarking and roadmapping processes mentioned above served as a basis for a symbiotic relation between the presented prescriptive methodology and the research work performed under the scope of these projects. On one hand, the systematic utilization of these processes across projects have paved the way to the formalization of all the steps of the methodology, from the generic requirements' elicitation and design considerations' specification for MobiTrafficBD Frameworks and the design of the logical components and data flows model to the catalogue of guidelines and best practices for the design and development of MobiTrafficBD Frameworks. On the other hand, the empirical research knowledge acquired by the author throughout the execution of the above projects has served as a proving ground for the creation, application and validation of the methodology, along with its models, guidelines and best practices, in real-world scenarios, reducing the design time of and guiding the creation of MobiTrafficBD Frameworks across projects.

This symbiosis is at the heart of the scientific contribution of the presented work: The methodology was created from the research work performed throughout the projects and the projects attested for the validity of the methodology and served as real-world demonstration scenarios for the application of the methodology to design and develop MobiTrafficBD Frameworks.

1.4 Document Structure

This section will overview the structure of this document. Figure 1.2 presents the bottom-up approach considered for the writing of the presented work. To fulfill the offerings mentioned above, the presented work will initially go through a detailed overview of the main literature works revolving around the disciplines of Spatiotemporal Data and Big Data, and the interdisciplinary conjunction of both these areas that forms the specific domain of Mobility- and Traffic-related Big Spatiotemporal Data, going through all phases of the data lifecycle, such as collection, modelling, cleaning and interoperability, processing and data and visual analytics.

The literature review, more than providing evidence of the already mentioned lack of an established methodology, with its guidelines and best practices, to handle and manage MobiTrafficBD, will try to encompass the main tools, challenges and relevant academic works regarding the above topics, to serve as an initial guide for researchers and practitioners towards a thorough understanding about specific tools, methods, models and solutions to manage and extract value from MobiTrafficBD. The literature review will be presented in Chapter 2.

Second, the prescriptive, data-driven methodological approach will be presented, starting from the generic characteristics, functional and non-functional requirements and design considerations for MobiTrafficBD Frameworks, and going through the thorough description of the model of logical components and data flows that is the basis of the prescriptive methodology, along with the technological infrastructure model that fulfills the logical components' technological stack. The presentation of the methodological approach will be concluded with a list of general guidelines and best practices for the design and development of MobiTrafficBD Frameworks.

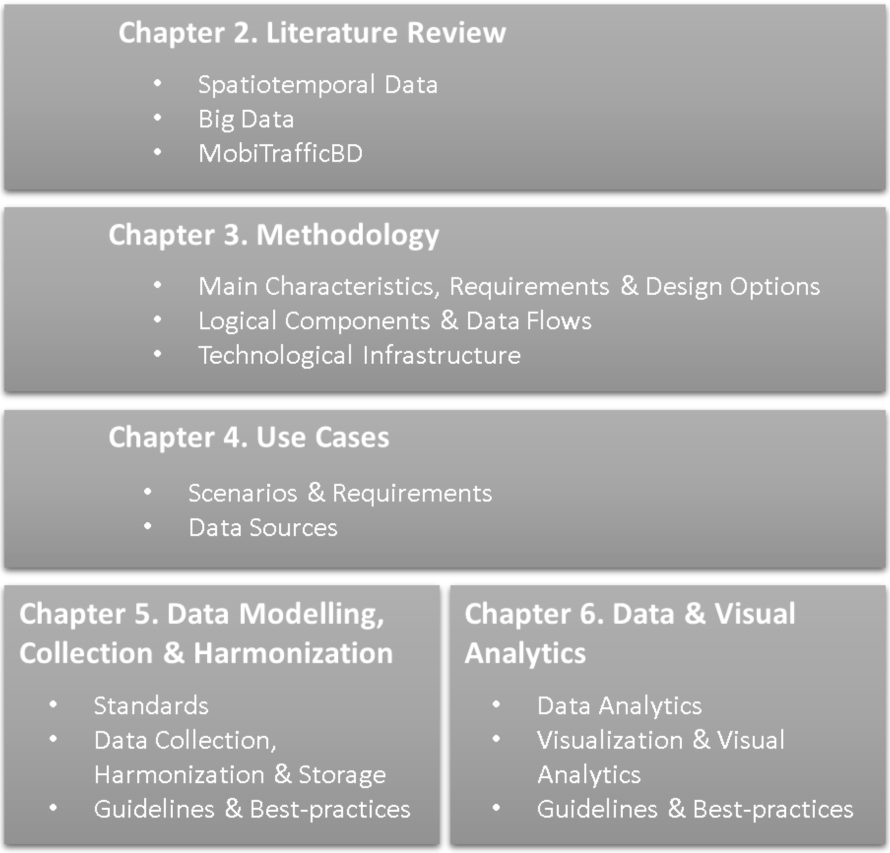

Chapter 2. Literature Review
• Spatiotemporal Data
• Big Data
• MobiTrafficBD
Chapter 3. Methodology
• Main Characteristics, Requirements & Design Options
• Logical Components & Data Flows
• Technological Infrastructure
Chapter 4. Use Cases
• Scenarios & Requirements
• Data Sources
Chapter 5. Data Modelling, Collection & Harmonization
• Standards
• Data Collection, Harmonization & Storage
• Guidelines & Best-practices
Chapter 6. Data & Visual Analytics
• Data Analytics
• Visualization & Visual Analytics
• Guidelines & Best-practices

Figure 1.2 — Document structure

Third, real-world use cases of MobiTrafficBD Frameworks put into practice will be highlighted through the overall descriptions, general requirements and in-depth listing of the data sets at hand for each use case. The use cases cover several MobiTrafficBD-related challenges, such as pipelines for Big Spatiotemporal data collection and harmonization (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018; Figueiras, et al., 2016; Figueiras, et al., 2016), complex-event processing for traffic event detection (Figueiras, Antunes, Guerreiro, Costa, & Jardim-Gonçalves, 2018; Antunes H. A., 2017), real-time analysis of traffic flows (Figueiras, et al., 2018; Rosa, 2017), public transport network status analysis and visualization (Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019) and social-media data mining for traffic event detection (Gutiérrez, Figueiras, Oliveira, Costa, & Jardim-Gonçalves, 2015). The use cases were tested, validated and demonstrated under the scope European Commission-funded and nationally funded research projects and provide not only validation and demonstration cases implemented and deployed in real-world scenarios, backed up by relevant Traffic and Mobility domain stakeholders and experts, but will also serve as example guides for researchers and practitioners towards the application of the presented prescriptive, data-driven methodology.

Fourth, the specificities of the methodological approach will be reviewed in detail, by dividing the logical components into two main groups:

- The first group concerns all data-driven processes spanning from data collection, cleaning, transformation, harmonization, etc. up to data storage. This group corresponds to Chapter 5 of this document and will focus on data modelling strategies, the use of standards for this purpose and on the data collection, harmonization, transformation and storage processes.
- The second group will go into MobiTrafficBD Analytics and visualization and will be the subject of Chapter 6. Since Data Analytics and Visualization tend to be more use case-driven than data-driven (in the sense that the type of data analytics and/or visualization methods chosen depends more on the final objective of the use case and less on the data at hand), this group will focus on more general guidelines and best practices that can be applied across use cases.

The discussion about the above groups will include detailed examples extracted from the presented use cases and will strive to guide researchers and practitioners through the different phases of the data lifecycle, from its collection, initial pre-processing and storage to the value-extracting processes of data analytics and data visualization.

2

Literature Review

This chapter presents a literature review on the most relevant concepts in the context of this dissertation. Section 2.1 introduces notes regarding the literature review process, while the following sections focus on the characteristics and methods related to spatiotemporal data and Big Data, finishing with a thorough review on the main topic of the presented work: mobility- and traffic-related Big Spatiotemporal Data.

2.1 Notes About the Literature Review Process

The creation of a systematic process for the retrieval of pertinent literature within the conceptual framing phase is essential not only to ease the research and reading tasks but also to maintain a consistent work method. Therefore, an initial phase of the literature review process was to define search keywords, reference databases and paper selection methods. Since the idea is to access the most up-to-date literature available, several temporal filters were used, depending on the novelty factor of each concept to study. Hence, for Big Data, for instance, due to the concept's youth, no time filters were used, but in the case of GRTS, a filter comprising the last ten years was used. Moreover, several alerts were enabled in reference databases, to maintain a tight monitoring on present and future published articles that may be important within the scope of the dissertation. Citations of works prior to 2010 occur if it is cited in another work under analysis, or if the work is a reference in the field.

Thus, the selected search keywords are composed by an aggregation of two main groups of keywords: keywords linked to the main concepts, and keywords which represent grammatical action nouns. The first group has keywords such as "Big Data", "time series", "geo-referenced time series", "spatiotemporal data" and "Intelligent Transportation Systems", among others, which were used individually or combined, and the second group is comprised by "optimization", "storage", "processing", "visualization", "harmonization", "indexation", “standards” etc. Searches were made by using keywords in the first group, individually or aggregated to keywords of the second group. The reference databases used for such searches include “Scopus”, “ISI Web of Science”, “RUN FCT-UNL”, “Google Scholar”, “IEEE Xplorer” and specific publishers’ repositories, as “Springer-Verlag” and “Elsevier”, just to name a few.

The selection of relevant articles was achieved in two stages. The first stage consisted in a quick “diagonal” reading of the initial set of papers, with emphasis given to the title, abstract and introduction, in which the papers were considered relevant or not relevant. The second stage was a thorough reading of the papers deemed relevant, to establish the degree of relevance and novelty of the presented work. Other literature sources were directly gathered, by recommendation of the supervisors, the citation in relevant articles and official technology sites. The filtered articles were thoroughly analysed, and important information and concepts were retrieved and used/cited throughout this document.

Finally, it is worth mentioning that this Literature Review strives not only to demonstrate the usefulness and value of the proposed approach, by pointing out the lack of a prescriptive methodology, comprising models, guidelines and best practices for the design, development and deployment of MobiTrafficBD Frameworks, but also to serve as a valuable source for practitioners and researchers that condenses works in the main areas that compose the MobiTrafficBD Framework ecosystem, from Spatiotemporal Data and Big Data as individual research domains, to the symbiotic relationship between these two domains, creating the more specific MobiTrafficBD research domain. Hence, this chapter intends to be more thorough and in-depth than regular literature reviews.

2.2 Spatiotemporal Data

Everything that occurs in space also occurs in time, i.e., phenomena always occur at some location in space within some time period. This implicit relation between time and space is all around, even more with the explosion in geographic and other types of spatially bound data, such as sensor data, social network data, etc. The analytical observation of movement by us, observers, or our observing and sensing technologies, produces data that can be better represented and defined as data types that have both space and time contexts or definitions (Lamigueiro, 2014).

On one hand, the time dimension represents the evolution of objects in time and defines the extent of such evolution present in data. The most trivial case is when the observed object does not evolve at all, meaning that the data only represents a static snapshot of each object. Another, more complex, case is when the object can change its status, which corresponds to an updated snapshot, but there is no information about its historical evolution. The extreme opposite of static objects would be to know the complete history of an observed object, thus creating a time series of the evolution it experienced.

The time dimension possesses an underlying semantic structure, comprised by its system of time granularities, hierarchically organized (e.g., seconds, minutes, hours, days) (Andrienko, et al., 2010), and represented in different calendar systems. These granularities hide natural cycles and re-occurrences, which may have a more regular (e.g., seasons) or less regular nature (e.g., social cycles, such as work and school holidays, economic cycles) and, while some of these cycles can be natural (e.g., seasons, volcanic activity), others are inherently human (e.g., seasonal public transportation routes). Furthermore, time may be viewed as continuous or discrete, depending on the data at hand. For instance, data may be captured at particular time instants, or points, such as in the case of discrete events (e.g., car crashes, sensor readings) or during extended, continuous time intervals (e.g., GPS routes of vehicles).

On the other hand, the space dimension describes position and movement properties of an object, whether it is fixed at a specific location or if its location can dynamically change over time. The spatial dimension may also discriminate the spatial extension of the objects observed. This dimension enables the location of observations and variables in a two-dimensional space, with one of the dimensions based on the North-South relation (latitude) and the other dimension based on the East-West relation (longitude). The most popular, simple, case is based on point-wise objects, which defines an object's location as a single point. More complex cases consider extended objects, such as lines and areas. This spatial feature might also be perceived as an extra dimension of space.

Any process performed on spatial information is both controlled and reinforced by Tobler’s fundamental first law of geography (Tobler, 1970): “everything is related to everything else, but near things are more related than distant things.”, i.e., spatial information is fundamentally characterized by the existence of spatial dependency. This means that often, two similar or nearby locations present close values for a particular variable or that a phenomenon in a specific location is probably a function not only of underlying factors, but also of the intensity of that phenomenon in nearby locations. Other important feature of spatial data is its heterogeneity, or the lack of stability in terms of the behaviour of relationships over space since spatial data rarely has stationary characteristics. This is also known as non-stationarity (Brundson, Fotheringham, & Charlton, 1998).

These inherent characteristics create several challenges for spatial data analysis, due to the local sensitivity aspects of the interaction between processes in space (Zhou & Lin, 2017). On one hand, global models and statistics do not constitute good tools, since they tend to fade out complex interactions between processes in small, nearby places, such as in the case of traditional Geographic Information Systems applications (Miller & Han, 2009). On the other hand, errors and uncertainty in spatial data are often spatially aggregated (Miller & Han, 2009). There are still other issues, such as discrepancies between the real world and its representations, the fuzziness of spatial data, or the problems of spatial data collection (e.g., sampling, extensiveness, redundancy and aggregation issues) (Heuvelink & Brown, 2017; Pauly & Schneider, 2017).

Further, there are many commonalities between space and time dimensions of data. On one side, Tobler's law for space interdependence and autocorrelation has similar twin concepts for relationships with expression on the temporal dimension (Andrienko, et al., 2010). These dependencies constrain the use of standard statistical analysis techniques, since these often assume independence between observations. On the other hand, spatiotemporal processes exist and evolve at different space and time scales or granularities (Silva R. A., 2017). On the space dimension, and due to Tobler's law, the scale of spatial analysis may significantly affect analysis results, because certain phenomena not detected at a particular spatial scale, may be clearly visible at another, bigger or smaller, scale. This also happens with time, as the time and space scales chosen may affect analysis in similar ways (Laube & Purves, 2010).

2.2.1 Spatiotemporal Data Types

Spatiotemporal data is defined in the literature as data that presents some level of dynamicity and variability in one or both of its space and time scopes (Heuvelink, Pebesma, & Gräler, 2017), i.e., data that comprise both space and time scopes, with these independent scopes presenting or not changes for different measurements (Atluri, Karpatne, & Kumar, 2018; Shekhar, Zhang, & Huang, 2010; Pebesma, 2012; Kisilevich, Mansmann, Nanni, & Rinzivillo, 2009), and these differences define the different spatiotemporal data types available. In the context of the presented work, the different types of spatiotemporal data that will be considered are shown in Figure 2.1.

These relations can be represented in a Cartesian plane, in which each observation comprises a location described by two points that represent the geographic coordinates. This representation on the Cartesian plane enables the calculation of distances between two points, each point representing the location of a particular observation, which can then be used to determine the intensity of the relations between the observations (Dubé & Legros, 2014). As presented in (Wu, Zurita-Milla, & Kraak, 2015), from the panoply of point-wise spatiotemporal observations gathered, several data types may be differentiated, regarding their spatial and temporal dimensions (Kisilevich, Mansmann, Nanni, & Rinzivillo, 2009): (1) spatiotemporal events; (2) geo-referenced variables; (3) GRTS; (4) moving objects (points) and (5) trajectories.

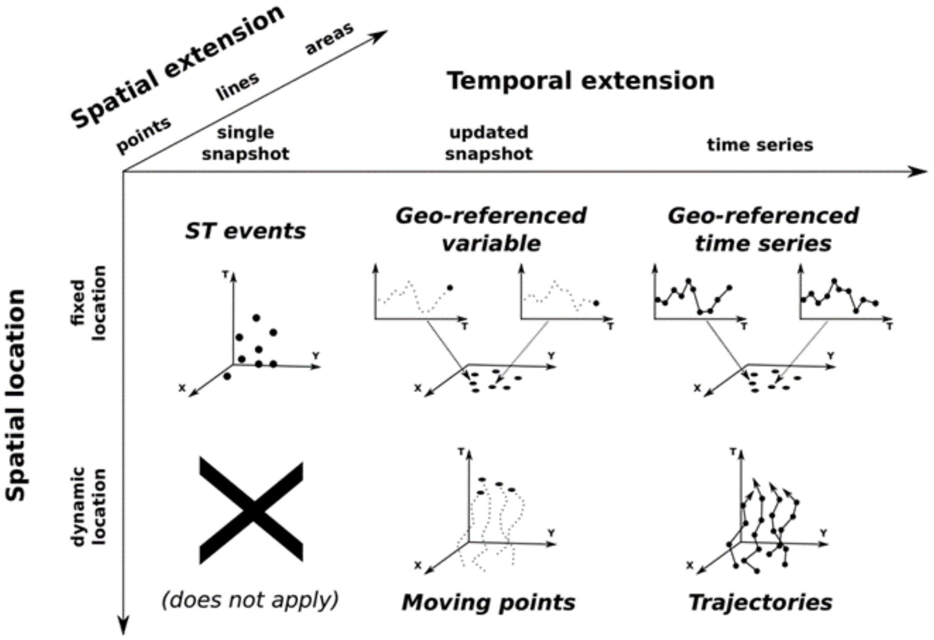


Figure 2.1 — Spatiotemporal data types, as defined by (Kisilevich, Mansmann, Nanni, & Rinzivillo, 2009)

The spatiotemporal data types depicted in Figure 2.1 arise from the static or dynamic nature of each dimension's scope, space and time: when both scopes are static, the data describes a single event, clearly defined in both time and space (e.g., place and time of an accident); when space is static and the time scope is dynamic, the data describes the evolution of measurements about phenomena on a single, well-defined place, throughout time, also coined as geo-referenced time series, or GRTS (e.g., road sensor values for instant speed, vehicle flow, etc.); and finally, when both time and space are dynamic, the data depicts moving objects (e.g., moving weather balloons continuously capturing weather data at different points of space and time) and trajectories (e.g., the movement of individual vehicles or people throughout a day).

A spatiotemporal event (ST event, as an abbreviation) is often characterized by fixed location and fixed time, representing where and when the event happened. Apart from these two dimensions, spatiotemporal events include other variables that provide additional information about the event, denominated marked variables (Silva R. A., 2017). For instance, in the case of traffic events, a common marked variable is the type of event that occurred (e.g., accident, traffic jam, etc.), providing a categorization of the event. Although most ST events may be defined by a single point in space and time, there are simple extensions that are common in real-world applications, such as the definition of the space scope of the event not by a point, but rather by a line or a polygon (e.g. traffic jam), or the depiction of the event in a time interval, such as the start and end times of the event (e.g. the time at which an accident occurred and the time at which the accident ended). Even so, most applications and methods for analysing this type of data are based on spatiotemporal points.

GRTS data is composed by measurements of continuous or discrete spatiotemporal fields, recorded at fixed locations in space. The measurement points may be regularly or irregularly spaced in both time and space (e.g., road sensor networks). While some examples of this data type comprise observations at point vertices (e.g., measurements collected by a sensor network), others make aggregated measurements over a region. Further, when analysing GRTS with varying resolutions, data is often converted from its native resolution to a finer or coarser resolution, so that a seamless analysis of all GRTS can be performed at a common resolution (e.g., temporal aggregation of sensor data to a single resolution, such as five-minute intervals, spatial aggregation of sensor data, to get regional measurements).

Moving point data comprises observations of continuous spatiotemporal fields throughout a set of moving points in space and time, such as weather balloons that capture meteorological observations. In both cases, finite, discrete samples of spatiotemporal points are used to depict the behaviour of a continuous spatiotemporal field. Normally, this kind of data is gathered through the aggregation of data captured within smaller space and time scopes.

Finally, trajectories denote the movement paths of objects or bodies in space throughout time, and are often collected by placing spatial sensors, such as GPS devices, on the moving objects and periodically measure and transmit their position over time. Examples of trajectories include routes taken by vehicles in a road network or migration patterns of animals and humans. Trajectories are a common problematic in several works in the literature, focusing on moving object data on the different stages of data handling and analysis: collection, storage, processing and querying (Silva & Santos, 2010; Xiaofeng, Ding, & Xu, 2012; Raza, 2012), analytics and data mining (Atluri, Karpatne, & Kumar, 2018; Andrienko, Andrienko, Fuchs, & Wood, 2017) and visualization (Bach, Dragicevic, Archambault, Hurter, & Carpendale, 2014), as well as on their application on real-world scenarios (Atluri, Karpatne, & Kumar, 2018).

Looking at Figure 2.1, the spatial extension dimension is not contemplated in the above descriptions. This dimension enables the definition of more spatiotemporal data types, by providing combinations of the spatial and temporal properties on objects that comprise a spatial extension, such as lines (e.g., roads) and areas (e.g., catastrophe extension). This third dimension and its implications will not be a subject of this work.

Moving point data is mainly used for applications in which the uncertainty of the spatial position of the measurement or observation does not affect the overall data analysis, since the produced fuzziness does not affect the analysis of a continuous phenomenon on a wider scale. Examples are the evolution of meteorological (e.g., wind, sea currents, etc.), biological (e.g., fish stocks and movements) or other variables across space and time. It is difficult to apply reference point data to traffic analysis and management due to the produced fuzziness and spatial uncertainty of the observations.

On the other hand, trajectories are crucial for traffic applications, but there are some limitations for its use by public operators. First, most public traffic-related operators do not have access to trajectory data from drivers and commuters, due to public privacy laws. Although some private companies have created applications using trajectory data from users, namely Google with Google Maps and Waze, public authorities and operators do not have access to it. Also, there is a problem of accuracy: Moving object and trajectory data may represent accurately moving behaviours, but to represent the reality of the entire road network involves having access to big volumes of trajectory and moving point data, otherwise this representation may be inaccurate. One example of poor accuracy on mobility analyses from moving objects data has to do with average speeds errors due to the lack of data: If the data is collected from just one vehicle, and the vehicle is going at a low speed, although there is no traffic in the road, then the overall speed on that road will be wrongly considered as the speed of the single vehicle. Another example has to do with heavy traffic depictions on roads with traffic lights, although it is normal for vehicles to slow down or stop due to the traffic lights, and such decrease in speed does not account for heavy traffic.

As already stated in Chapter 1, the focus of the presented work will be ST Event and GRTS data since these are directly linked with traffic-related data that do not fall into the categories of moving objects or trajectories and take advantage of the static nature of the space dimension's scope. In the literature, different approaches were presented to analyse and mine the different spatiotemporal data types, meaning that data mining and analysis methods over spatiotemporal data are applied on one or more spatial and temporal models, to extract further insights from the data. Within the scope of the presented work, the focus will be given to spatial object-based models and to temporal snapshot and event and process models.

2.2.2 Spatiotemporal Data Storage

Spatiotemporal data storage is not a new research field. Since the end of the 90's, researchers sought to find the best ways to store data with both time and space dimensions in database systems (Kim, Ryu, & Kim, 2000). Although such research endeavours produced some prototypes of spatiotemporal databases (e.g., (Kim, Ryu, & Kim, 2000; Sözer, Yazici, Oğuztüzün, & Taş, 2008)), for the best of the author's knowledge there are few off-the-shelf spatiotemporal database systems available. Some examples of proprietary systems are Spacetime (Mireo d. d., 2020), a relational database management system (RDBMS) for spatiotemporal analytical workloads, or ArcGIS Data Store (ESRI, 2020), the data storage application for hosting servers of the ArcGIS Enterprise software. It is worth to mention that neither of the examples are free or open-source, and the latter can only be used in conjunction with ArcGIS Enterprise, which is a major downside.

Hence, most approaches opt for building extensions to be applied on already existing database systems, in two different ways: spatial extensions for databases that only handle temporal information, such as in the case of time series and relational database systems, or temporal extensions for databases that only handle spatial information, such as in the case of spatial and Geographic Information Systems (GIS) databases (Fan, Yang, Zhu, & Wei, 2010). Normally, the first option is more popular, simply since there are more time series and relational database systems available than spatial database systems, and GIS databases are often related to trajectory and raster data storage (Yue & Tan, 2018). Thus, this section will focus on the spatial extensions for time-driven and relational database systems.

Regarding traditional RDBMS, several systems provide spatial extensions to integrate spatiotemporal data into relational tables. The most popular is PostGIS (PostGIS Project Streering Committee, n.d.) for PostGreSQL (The PosgreSQL Global Development Group, 1996). It adds support for geographic objects allowing location queries to be run in SQL and enables yet other spatial extensions, such as the pgrouting extension (pgRouting Community, 2007), which offers a range of methods to calculate any type of route (e.g., land vehicle routes, sea ship routes, etc.). Another example is Oracle Spatial (Oracle Corporation, 2019), which is also an extension to the Oracle RDBMS (Oracle Corporation, 1979). Some more examples can be found in (Chen & Xie, 2008). Also, in the academic domain, several works were published since the 90's about this subject. Regarding an example of recent works, the authors of (Martinez-Llario & Gonzalez-Alcaide, 2011) propose Jaspa, a Java spatial extension for RDBMS that mimics the offerings of PostGIS but can be integrated with other RDBMS, such as Oracle and HSQLDB (The HSQL Development Group, 2001).

There are also new data storage paradigms that already present the capability of handling both time and space, such as in the case of MongoDB (MongoDB, Inc., 2015), which represents time through a timestamp format and uses the GeoJSON (The Geographic JSON Working Group, 2016) spatial data representation standard to handle the spatial dimension, and ElasticSearch (Elasticsearch B.V., 2010), which has the ability to save both coordinates and geohashes, along with temporal attributes. As in the case of RDBMS, some data storage tools that do not have the capability to handle spatial data can be extended to be able to perform spatial operations and indexing (Fox, Eichelberger, Hughes, & Lyon, 2013; Brahim, Drira, Filali, & Hamdi, 2016). The former work gave rise to a widely used tool, called GeoMesa (The GeoMesa Project , 2013), which is an open-source, distributed, spatiotemporal index built on top of Bigtable-style databases, such as Google's BigTable (Google, Inc., 2005) and Apache Accumulo (The Apache Software Foundation, 2008), using an implementation of the Geohash algorithm presented in (Fox, Eichelberger, Hughes, & Lyon, 2013). But there is still some room to improve the performance of NoSQL databases for spatial operations, in relation to RDBMS and spatial extensions, as is evident in several comparison works in the literature between RDBMS and NoSQL databases, often using PostGreSQL and PostGIS versus MongoDB as benchmarking subjects (McCarthy, 2014; Agarwal & Rajan, 2016; Bartoszewski, Piorkowski, & Lupa, 2019). Ultimately, the selection of the appropriate database system for spatiotemporal data is inherently linked to the nature of the data and the domain at hand.

2.2.3 Spatiotemporal Data Interoperability

Due to the amount and heterogeneity of data sources available nowadays, there is a need to agree on common grounds when it comes to collect and exchange data between different systems. These common grounds or agreements are known as interoperability. Interoperability is defined as the ability of two or more software components/systems to cooperate independently of the individual characteristics of such systems (Grilo & Jardim-Gonçalves, 2010). There are different levels of interoperability (van der Veer & Wiles, 2008): technical interoperability, which is centred on (communication) protocols and the infrastructure needed for those protocols to operate to enable machine-to-machine communication, syntactical interoperability, which is associated with common data formats for data exchange, semantic interoperability, which is associated with a common understanding between systems and the people using them of the meaning of the content (information) being exchanged, and organizational interoperability, which is the ability of organizations to effectively communicate and transfer (meaningful) information even though they may be using a variety of different information systems over widely different infrastructures, possibly across different geographic regions and cultures.

It is easy to understand that, nowadays, effective technical interoperability is already in place, since both new and legacy systems easily exchange data via communication protocols, such as in the case of the well-known TCP/IP protocol for data exchange over the Internet. The main issues lie in syntactical and semantic interoperability levels. On one hand, data formats are increasing in number and complexity to cope with the exponentially growing volume and heterogeneity of data sources, and there are few standardized, one-size-fits-all approaches for syntactical interoperability. There are two main types of syntactical conflicts (Sonsilphong, Arch-int, Arch-int, & Pattarapongsin, 2016; Park & Ram, 2004). The first are data-type conflicts, which occur when data values of equivalent attributes are defined with different data types (e.g., a date attribute may be represented in one system as a string and in another as a timestamp) and the second are data-format conflicts, which account for the disparity of formats between equivalent data values (e.g., a date attribute may be represented in the format "yyyy-MM-dd" in one system while being represented in the format "dd.MM.yyyy" in another).

On the other hand, semantic conflicts appear in two different levels, data and schema (Sonsilphong, Arch-int, Arch-int, & Pattarapongsin, 2016; Park & Ram, 2004). On the data level, conflicts arise from the existence of multiple representations and interpretations of similar data sources. Data level conflicts may also be divided into data-value, data-unit and data-precision conflicts. Data-value conflicts occur when a specific value may have different meanings (e.g., locations may be represented in different reference systems, meaning that the same value for a specific latitude and longitude may represent different references) or two different values may have the same meaning (e.g., the attributes "coordinates" and "location" may have the same meaning, both defining the spatial dimension). Data-unit conflicts are related to the specific units in which a data value is represented (e.g., the distance attribute may be represented in kilometres in one system whereas in other system may be represented in miles). Finally, data-precision, or data-scaling, conflicts arise when data values are represented with different granularities (e.g., the time interval for sensor readings stored in a system may be five minutes while in another system may be one minute).

Finally, schema-level conflicts are divided into schema discrepancies and naming, entity-identifier, schema-morphism, generalization and aggregation conflicts. Schema discrepancies arise when the logical structure of attributes and corresponding values of an entity in one database are organized to produce a different structure in another database. The most common example is when data in one database correspond to metadata in one database (e.g., the values of an attribute in one system correspond to actual attributes in another system).

Particularly in mobility- and traffic-related applications, data harmonization and aggregation techniques of heterogeneous datasets are not easy tasks, mostly because most sensor data or existing services in ITS use an isolated approach of interoperability, which is an important problem to tackle to achieve better management of the global transportation network. Most interoperability approaches are made on a technical level, which is defined as the common understanding in messaging specifications, communication protocols, data formats and service discovery specifications, between others (European Commission, 2010). Hence, ITS interoperability's future goal is to enable interoperability at data level, since the objective is to use a large diversity of transportation-related data/services developed in different formats. The adoption of common standards seems to be the easiest way not only to meet the interoperability requirements at data level, but also to provide a higher level of data harmonization. The advantages of using a standard/specification to achieve data interoperability and harmonization is backed up by several studies.

In (Westerheim, 2014), experiences of The Norwegian Public Roads Administration to achieve interoperability are presented, and it is understood that, regardless of the approach for an ITS system, the solution must be provided with technical interoperability standards and specifications. The more widespread and used the solution is, the better is the understanding and the meaning of the information between different ITS systems, improving management of the overall network. In (Nowacki, 2012) and (Samper, Tomás, Soriano, & Pla Castells, 2013) the need of harmonizing all types of ITS-related data due to the proliferation of information and telematics systems is demonstrated, along with the major role of the European Transport Policy that, through ITS directives, can influence and contribute towards a concerted interoperability and harmonization of data exchange between new intelligent transport systems. The authors of (Samper, Tomás, Soriano, & Pla Castells, 2013) highlight the European role with the EasyWay programme (European ITS Platform, 2016), which since 2007 is working "for harmonized deployment of ITS across Europe", joining multiple key players. Fundamentally, the objective is the deployment of guidelines for stakeholders to spread best practices and knowledge, supporting projects and studies in the area. Therefore, it is important to explore the standard that best suits the goals of the presented work.

DATEX (Easyway, 2011) was first published in the end of 2006 and acknowledged in 2011 by the European Technical Specification Institute (ETSI) (European Telecommunications Standards Institute, 2015) for modelling and exchanging ITS-related information, being a European standard for ITS since then. It has been developed to provide a way to standardize information covering the communication between traffic centres, service providers, traffic operators or media partners. Thus, the European Commission made EU Directives 885/2013/EU and 886/2013/EU that require Member States to adopt the DATEX II standard or an equivalent one. Since the first release many aspects have been improved. It is developed and maintained by the EasyWay project (Easyway, 2011) and supported by the European Commission. Some of the main uses are:

- Routing/ rerouting using traffic management.
- Linking traffic management and traffic information systems.
- Multi-modal information systems.
- Information exchange between cars or between cars and traffic infrastructure systems.
- Applications where the exchange of measured data is important.

Almost any traffic-related issue is covered, but as an example, some of the situations that the message covers are:

- travel times.
- all types of traffic events and accidents.
- road works and infrastructure status.
- road weather events and status /measurements.
- traffic related measurements (speed, flow, occupancy).

- events with impact on traffic.
- CCTV and parking.

There are four main design principles involved in the creation of DATEX II: separation of concerns, in terms of their application domains, a rich domain model, which allows a comprehensive and well-defined modelling of data, extensibility, allowing specific extensions depending on country or area, and data exchange. As described in (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018), DATEX II supports and informs many ITS applications, particularly when cross border trips are concerned. DATEX II provides the platform and technical specification for harmonized and standardized data modelling and data exchange in ITS applications.

The former DATEX specification was a 'closed' standard, i.e., a potential user could use the data concepts provided by the Data Dictionary, but if the application required data concepts that could not be found in the Data Dictionary, there was no way to extend the data model without breaking the standard. Many potential users thus ignored DATEX entirely and produced non-interoperable solutions. It is one of the major requirements for the evolution of DATEX to overcome this problem. The solution for this issue is in part solved by the introduction of data model levels into the DATEX II data structure.

DATEX II delivers a data model, called "Level A" data model, which is the result of studies on data which are shared by users across Europe. Nevertheless, there will be situations in which data concepts required by a particular user are missing in the Data Dictionary, for example because they only make sense in a national context. To cater for this future proofing aspect of modelling it is desirable to have a formal mechanism by which the "Level A" model can be extended. For these new applications requiring extensions to the "Level A" model, the concept of "Level B" compliance has been created. This will allow development of specific models that will enrich the "Level A" model with additional, application specific information. These models/applications will remain interoperable with "Level A model compliant suppliers/consumers: they can exchange objects structured according to these enriched models.

After consideration of "Level A" and "Level B" compliance rules some users within the ITS domain may still find that there is no way that their specific data models can be accommodated. They are just too different from the "Level A" model or else cover completely different contents. That is why the concept of "Level C" has been created. "Level C" implementations are to be considered as not compliant with the DATEX II "Level A/B" content models. However, they are to be compliant in all other aspects of the DATEX II specifications.

More recently, in 2018, the DATEX II Light (DATEX II, 2018) specification was released. DATEX II light, also coined D2Light, is a JSON representation model that fully corresponds to the former XML-based DATEX II model. D2Light is aimed at the exchange between a Traffic Centres and small Service Providers as well as the exchange with end-user apps directly. Since, the DATEX II model was heavy, in terms of its XML structure, there is a strong demand from road operators to open their data to app developers in order to achieve maximum usability of their data in the Open Data domain. To support the ability to meet this demand, the Activity Group behind DATEX II agreed on creating a simpler, lighter version of the standard model.

2.2.4 Spatiotemporal Data Mining and Analysis

Although spatiotemporal data mining is a prolific research field, it is often divided into spatial and temporal methods, which are applied separately to produce distinct insights and results that are then merged to enable a spatiotemporal analysis on data. This is a common process when analysing and mining spatiotemporal data. True spatiotemporal data mining methods, which encompass both the temporal and spatial characteristics of data, are rare in the literature, and are often linked to the analysis of moving objects and trajectories (Silva & Santos, 2010; Xiaofeng, Ding, & Xu, 2012; Raza, 2012; Bach, Dragicevic, Archambault, Hurter, & Carpendale, 2014), which are not the focus of this work.

The main temporal counterpart of GRTS are time series, while ST events can be temporally translated to discrete temporal events, or single observations within time series. Hence, it may be worth defining what time series are, and what are their main characteristics. Time series are series of data points which were sequentially collected throughout a period of time, which makes them discrete points ordered and equally spaced in time. Time series arise from monitoring and tracking processes within several contexts of our society. One example is traffic sensor data, which depicts the quality-of-service level of roads, by providing observations evenly spaced in time of several traffic-related variables such as average speed, number of vehicles, time gap between vehicles, etc. One can consider these data points as events happening on a time instant. Such points can be extracted from time series (e.g., in the case of a specific sensor reading) or can be aggregated to create a new time series (e.g., traffic events' history during a year).

Generally, time series can be divided into four main components, which can be separated individually from the observed data: Trend, Cyclical, Seasonal and Irregular components (Adhikari & Agrawal, 2013). A time series is said to have a trend when there is a slowly evolving change, thus, the trend is a long-term movement in a time series, and it is said to have a seasonal component when some cyclic pattern emerges within the period of one year, with the period being the amount of time for a cyclic phenomenon to repeat itself. The cyclical variation in a time series describes the medium-term changes in the series, caused by circumstances, which repeat in cycles. The duration of a cycle extends over longer periods, usually two or more years. Finally, irregular or random variations in a time series are caused by unpredictable influences, which are not regular and do not repeat in a particular pattern.

Two main data processes are normally associated to time series: time series analysis and time series mining. Time series mining uses statistical and data mining techniques to extract insights and inherent characteristics of data, such as trends, seasonal behaviours, outliers or abnormal observations, leading to improved knowledge. Time series analysis uses a model built to represent the time series and employs techniques to forecast what will be the future behaviour of these series.

On the other hand, spatial data mining aims at extracting insights in the form of spatial interactions and other properties linked with the spatial component of data (Zeitouni, Yeh, & Aufaure, 2007). Like traditional, temporal data mining, the goal of spatial data mining is to deliver a set of tools and methods to extract automatically or semi-automatically relevant and easy-to-understand insights, such as rules, patterns, irregularities, or associations, based on the spatial interactions in the data being analysed. Such spatial interactions are directly linked to the first law of geography, or Tobler's law (Tobler, 1970), already defined in this section. In fact, spatial data properties for a particular location are often related and can be explained in terms of the surrounding neighbourhood's properties. Temporal aspects are also crucial but are rarely considered when analysing spatial data (Zeitouni, Yeh, & Aufaure, 2007).

Regarding the data inputs for spatial data mining, these are also more complex than the ones used in classical temporal data mining, since they comprise extended spatial attributes such as points, lines and areas, which contain the information about spatial locations (e.g., longitude, latitude, elevation, shape), along with non-spatial attributes, such as names and numeric variables corresponding to some kind of measurement or observation (e.g., number of cars per minute, unemployment rate in a city) (Shekhar, Zhang, & Huang, 2010). Furthermore, and contrasting with non-spatial objects' explicit relationships, relationships between spatial objects are generally implicit, such as in the case of overlapping, containing, intersecting or positioning of objects in relation to each other. One possible way to overcome the implicit nature of spatial relationships is to convert the relationships into normal data columns and then apply classical data mining techniques. This technique can result in loss of information. Another workaround is to develop and apply models and methods to incorporate spatial information into the data mining process (Shekhar, Zhang, & Huang, 2010).

Spatiotemporal data mining, as well as its spatial counterpart, is prone to many fundamental issues (Miller & Han, 2009). First, all spatiotemporal analyses are sensitive to the scale, or support, of the spatial and temporal scopes. This means that the spatial and temporal scales of data, usually selected upon data collection, determines which phenomena can be identified in the data. For instance, a phenomenon that is observed at smaller spatial or temporal scales, may not be observed at bigger scales, and vice-versa. This issue is called Modifiable Areal Unit Problem or the Ecological Fallacy. To bypass this issue, a clear understanding of the phenomenon's scale, or support level, is crucial.

Second, the nature of space and time differs quite a lot, since time is often perceived as unidirectional and linear, whereas space is bi-directional and nonlinear. Furthermore, although both scopes are continuous phenomena, time is generally represented as discrete and isomorphic integers at a higher granularity (e.g., hours, days, years), while space is often represented as isomorphic real numbers, with a generally smaller granularity associated. Hence, it is crucial to recognize the significant differences between spatial and temporal dimensions, even when analyses focus on static phenomena. Finally, the representation of spatial phenomena on a Cartesian plane is a poor representation of reality because it limits the inherent relationship between space and time. Time is essentially a spatial phenomenon, and the most common representations of time, such as continuous and linear, discrete, monotonic, or cyclic time, are limited aspects of time and have limited applications. For a thorough explanation on these issues and possible workarounds, please refer to (Miller & Han, 2009).

The following sections will present some of the more recent works on spatiotemporal data mining, namely on querying, clustering, classifying, summarizing, detecting anomalies and patterns, as well as predictive models for spatiotemporal data.

2.2.4.1 Querying

Spatiotemporal querying, just like the temporal counterpart, enables for spatiotemporal data search based on their spatiotemporal attributes, by defining both spatial and temporal query windows. (Luyi, Yan, & Ma, 2014) proposed an approach for querying fuzzy spatiotemporal data using XQuery, by making extensions to the XQuery language, while (Cheng, 2016) also proposed a model for representing fuzzy spatiotemporal objects and their topological relations. He proceeded to investigate how to design basic and complex fuzzy query operators so that it is possible to describe the evolution of fuzzy spatiotemporal objects over time. In two consecutive works, Magdy presents two systems for real-time spatiotemporal queries on microblogs with high efficiency: Mars (Magdy, et al., 2014) and Mercury (Magdy, Mokbel, Elniteky, Nath, & He, 2014). Mars supports a wide variety of spatiotemporal queries, while Mercury is based on top-k spatiotemporal queries. Both systems support high throughput of up to 64K microblogs per second and average query latencies of 4 milliseconds. For a thorough overview on indexing and querying techniques for spatiotemporal data, please refer to (John, Sugumaran, & Rajesh, 2016).

In (Eldawy, Mokbel, Alharthi, Tarek, & Ghani, 2015), a system for querying and visualizing spatiotemporal satellite data, based on the MapReduce paradigm (Dean & Ghemawat, 2008), is presented. (Doraiswamy, Vo, Silva, & Freire, 2016) presented a GPU-based indexing scheme for spatiotemporal queries on historical batch data. More recently, (Alarabi, Mokbel, & Musleh, 2018) proposed a full-fledged MapReduce framework for supporting spatiotemporal data queries. In (Galić, 2016), an overview of query mechanisms for spatiotemporal data streams is presented.

2.2.4.2 Clustering

Spatiotemporal clustering is directly linked with the first law of geography, but with a difference: Everything is related to everything else, but things that are closer to each other, in both time and space, are more related than distant things. This means that when clustering spatiotemporal data, both temporal and spatial proximity is of essence (spatiotemporal autocorrelation). This is one of the most prolific subjects on Spatiotemporal Data Mining, with several surveys available throughout the years (Atluri, Karpatne, & Kumar, 2018; Kisilevich, Mansmann, Nanni, & Rinzivillo, 2009; Senožetnik, Bradeško, Kažic, Mladenic, & Šubic, 2016; Shi & Pun-Cheng, 2019; Ansari, Ahmad, Khan, & Bhushan, 2020).

In (Anbaroglu, Heydecker, & Cheng, 2014), the authors present a spatiotemporal clustering method for non-recurrent traffic congestion detection, based on link journey times clustering on adjacent urban road links. The authors of (Saeedmanesh & Geroliminis, 2017) also aim at studying the spatiotemporal relation of congested links, observing congestion propagation from a macroscopic perspective, by applying a dynamic clustering method to capture spatiotemporal growth and formation of congestion. More recent relevant works include (Wu, Zurita-Milla, & Kraak, 2015) and (Wu, Zurita-Milla, Verdiguier, & Kraak, 2018), which present novel algorithms for georeferenced time series clustering.

2.2.4.3 Classifying

Spatiotemporal classification is often based on coupling or tele-coupling. Spatiotemporal coupling describes the occurrence of spatiotemporal objects in close geographic and temporal proximity, whereas tele-coupling patterns represent temporal correlations between spatiotemporal objects that are spatially further apart (Shekhar, Jiang, Ali, Eftelioglu, & Tang, 2015). Some examples of coupling and tele-coupling patterns are presented in (Shekhar, Jiang, Ali, Eftelioglu, & Tang, 2015). In (Steiger, Westerholt, Resch, & Zipf, 2015), the authors apply semantic and spatiotemporal classification to analyse the spatiotemporal autocorrelation between geo-referenced Twitter data and official census data for the city of London. Further, classification processes are often based on spatiotemporal clustering techniques and posterior classification of the resulting clusters (Reich & Porter, 2015; Lin, Chang, Wang, Huang, & He, 2019).

2.2.4.4 Summarizing

Spatiotemporal summarization provides a reduced representation of spatiotemporal data (Atluri, Karpatne, & Kumar, 2018). Summarization is important not only for data compression, but also for easing pattern analyses. In the case of GRTS, the summarization may be achieved by defining sets based on the partition of the series and posterior representation of the sets, using spatiotemporal nodes, paths or trees (Oliver, 2016), or by simply removing the spatial and temporal redundancy due to the effect of autocorrelation (Atluri, Karpatne, & Kumar, 2018).

The authors of (Pan, Demiryurek, Banaei-Kashani, & Shahabi, 2010) proposed a family of summarization methods for direct application on resource-efficient summarization and accurate reconstruction of historic traffic sensor data, based on high temporal and spatial redundancy/correlation among sensor readings from individual sensors and sensor groups. (Oliver, et al., 2012) explored summarization of GRTS using a k-full trees method, featuring an algorithmic refinement for partitioning regions that led to computational savings without affecting result quality.

2.2.4.5 Detecting Anomalies

Spatiotemporal outliers are objects with spatial and temporal scopes' references whose non-spatiotemporal attributes greatly differ from other objects in their spatiotemporal neighbourhood, denoting a spatiotemporal discontinuity (Shekhar, Jiang, Ali, Eftelioglu, & Tang, 2015). To detect outliers in spatiotemporal data in an optimal way, all three types of dependencies, spatial, temporal and attribute-based, should be incorporated. (Shahid, Naqvi, & Bin Qaisar, 2015) proposed that such a technique for outlier detection should exploit the above dependencies by firstly use temporal-attribute correlations to identify outliers in specific locations, and then should invoke a spatial consensus to determine the presence of outliers in neighbouring nodes, to detect the presence of events.

Namely in GRTS, basic spatial outlier detection approaches, like visualization-based (e.g., variogram clouds, Moran scatterplots) or neighbourhood-based (e.g., scatterplots, neighbourhood spatial statistics) methods, can be used as a generalization for application in spatiotemporal neighbourhoods. Visualization approaches plot spatial locations on a map or graph, to find spatial outliers by visual inspection, whereas neighbourhood approaches define spatial or spatiotemporal neighbourhoods, to which a spatial statistic is applied as the difference between the non-spatial attribute of the current location and that of the neighbourhood aggregate (Shekhar, Jiang, Ali, Eftelioglu, & Tang, 2015). A thorough review on spatial outlier detection can be found in (Aggarwal, 2017). More recently, authors of (Zhao, Qu, Zhang, Xu, & Liu, 2017) proposed a data mining technique applied on passenger smart card data to understand the hidden regularities and anomalies of the travel patterns. (Shi, Deng, Yang, & Gong, 2018) proposed an approach that detects anomalies in spatiotemporal flow data, such as in the case of traffic flow data, by constructing dynamic neighbourhoods.

One special case of anomaly detection is complex event detection (CEP). CEP is a defined set of tools and techniques for analysing and controlling the complex series of interrelated events that drive modern distributed information systems (Luckham, 2008). These tools are often based on a set of rules that are applied over data streams to detect complex events. Recent works on CEP applied to Mobility- and Traffic-related spatiotemporal data include (Figueiras, Antunes, Guerreiro, Costa, & Jardim-Gonçalves, 2018) and (He, et al., 2020).

2.2.4.6 Discovering Motifs/Patterns

Motifs and patterns discovery is the process of mining data sets to find patterns that occur frequently over several instances within the data set. A common research trend is based on the discovery of structural patterns in spatiotemporal data sets that extract complex spatiotemporal dynamics (Atluri, Karpatne, & Kumar, 2018). In the case of GRTS, motifs are temporal observations that repeat across several spatial locations. A thorough survey on this kind of motif discovery can be found in (Atluri, Karpatne, & Kumar, 2018). In the case of spatiotemporal event points, patterns can occur in co-occurrence or sequentially.

Co-occurrence patterns denote subsets of spatiotemporal events that appear in close spatial and temporal proximity. One example of co-occurrence patterns are hot spots (Levine, 2017), which can be defined as locations where the number of events is unexpectedly high within certain time intervals (e.g., points in a road network where accidents occur often) (Shekhar, Jiang, Ali, Eftelioglu, & Tang, 2015). Sequential patterns happen when the occurrence of spatiotemporal events of a specific type trigger a sequence of other types of spatiotemporal events (e.g., traffic jams caused by an accident).

The authors of (Akbari, Samadzadegan, & Weibel, 2015) proposed a new generic method for co-occurrence pattern mining on complex applications such as air pollution patterns' discovery. This method enables the extraction of implicitly contained spatiotemporal relationships over different spatial feature types, such as points, lines and polygons. In (Wu, Zurita-Milla, & Kraak, 2015), authors present a GRTS co-clustering technique that enables simultaneous analysis of spatial and temporal patterns. (Nguyen, Liu, & Chen, 2017) presented a set of algorithms that implement causality trees from congestion data sets and estimate congestion propagation probabilities based on temporal and spatial information, revealing not only recurring interactions across spatiotemporal congestions, but also bottlenecks or flaws in the designs of existing road networks. Finally, (Andrienko, Andrienko, Fuchs, & Wood, 2017) proposed an approach to ease exploration of long-term flow data by means of spatial and temporal abstraction, to study the spatiotemporal patterns and trends of mass mobility based on origin-destination data sets.

2.2.4.7 Prediction

Spatiotemporal prediction is based on the construction of a model that can predict some spatiotemporal output variable (dependent variable) present in the data from the spatiotemporal input features or attributes (independent variables) also contained in the original data (Shekhar, Jiang, Ali, Eftelioglu, & Tang, 2015). When the output variable is discrete, the prediction is called spatiotemporal classification, whereas when the output variable is continuous, the prediction is spatiotemporal regression. As previously presented, time series prediction is a prolific research field, but there is the need to develop new approaches that incorporate the spatial dimension of GRTS (Atluri, Karpatne, & Kumar, 2018).

Several works on this challenge have been published recently. Different recurrent neural network prediction methods that comprise spatial features have been proposed in (Jain, Zamir, Savarese, & Saxena, 2016), (Jia, et al., 2017) and (Jia, et al., 2017). Other Deep Learning spatiotemporal prediction approaches were also presented in (Zhang J. , Zheng, Qi, Li, & Yi, 2016), (Wang, Gu, Wu, Liu, & Xiong, 2016) and (Polson & Sokolov, 2017), with the latter two being concerned with traffic prediction. Other methods, such as latent space models that use topological and temporal attributes of locations (Deng, et al., 2016) and improved variants of SARIMA (seasonal autoregressive integrated moving average) and genetic algorithm models for spatiotemporal prediction (Luo, Niu, & Zhang, 2018) were also proposed.

2.2.5 Spatiotemporal Data Visualization

Since the beginning of human history, people have used graphical and pictorial data representations to understand and disseminate information in a way that was appealing to other humans. The information that our ancestors collected about the movement of celestial bodies and their seasonality can be defined as spatiotemporal series of data. Some monuments were used to visualize these spatiotemporal series throughout the year, such as in the case of Stonehenge (Pearson, 2013), which was built as an astronomical observatory and a computational calendar. Interest in spatiotemporal data analysis and visualization has been present for several decades, if not centuries (Surkhovetskyy, Andrienko, Andrienko, & Fuchs, 2017), with the first known graphical representation of spatiotemporal series in literature dating back to the 10th or 11th centuries, depicting planetary orbits as functions of time (Funkhouser, 1936), as shown in Figure 2.2.

Figure 2.2 depicts not only the time dimension, but also the space dimension associated to the movement of celestial bodies. This intrinsic connection between time and space is very common, since represented data often possesses a spatial or geographical dimension associated to time. Some examples are floating car data, physical sensor data or weather data. Thus, the current exponential growth in data collection has brought the need to build tools that can help humans to effectively understand and share data by means of visual representations (OECD/ITF, 2015).

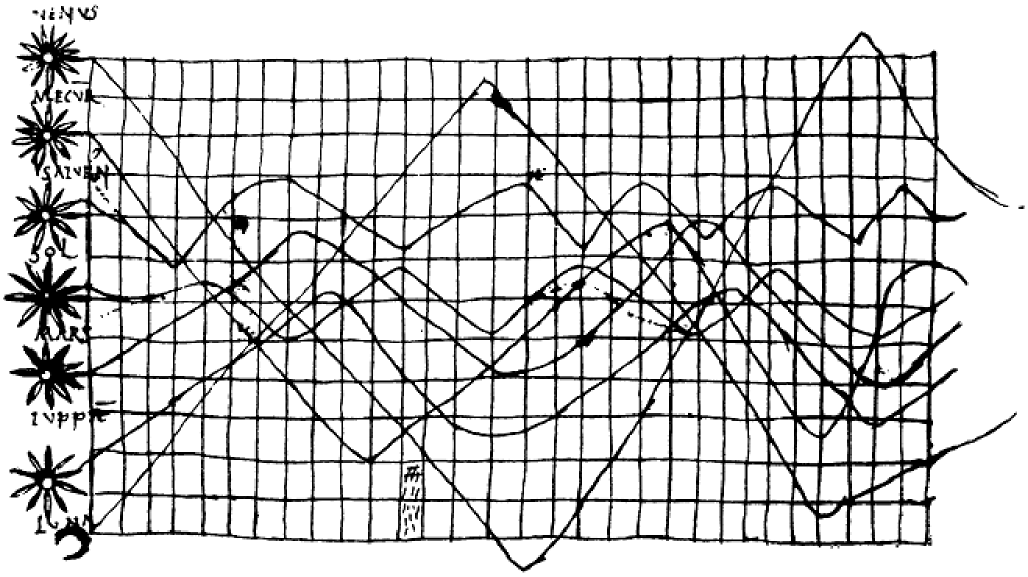


Figure 2.2 — Plot depicting planetary orbits (10th/11th century). The illustration is part of a text from a monastery school and shows the inclinations of the planetary orbits as a function of time (Funkhouser, 1936)

Data Visualization is the process of creating a visual representation of information using algorithms (Shrestha, 2014). The need for such algorithms is the characteristic that separates computer science-based visualizations from other visualizations in different fields of study, as it ensures reproducibility, i.e., if the original data is the same and the algorithms' rules are fully complied, visualizations may be gathered independently from users, platforms or choice of programming language. In essence, data visualization encodes information into a visual representation using graphical symbols, or glyphs (e.g., lines, points, rectangles, and other graphical shapes). Then, human users visually decode the information by exercising their visual perception capabilities (Steed, 2017).

In Data Science, the ideal scenario would be that systems could automatically discover knowledge from data without human supervision. Nevertheless, data analysis is generally an exploratory and complex process that fully automated solutions cannot achieve without posing trust issues upon the results. Data visualization methods help to discover patterns and relationships to find the best model that fits the data at hand, enabling automation of the data analysis process (Steed, 2017). When this happens, data visualization's goals shift from the exploratory process to confirmation and dissemination of resulting insights and knowledge.

But, due to the sheer volume and complexity of modern datasets, the process of data exploration through manual inspection is not feasible.

Hence, new solutions are needed that can enable interactive data visualization and provide analysis techniques that combine the power of computers with the strengths of human visual understanding, by developing new data analysis and visualization methods, technologies and practices. Such solutions are part of a new "umbrella" term, called Visual Analytics (Andrienko, et al., 2010). Visual analytics is a branch of data visualization, which focus on the orchestration of interactive visualizations with data mining algorithms (Steed, 2017). The key characteristics of visual analytics are (Andrienko, et al., 2010):

- emphasis given to data analysis, decision-making and problem-solving tasks.
- application of automated methods for data processing, knowledge discovery and data mining.
- support for direct human supervision of the analytical process, through interactive visual interfaces.
- support for the provenance of analytical insights.
- support for communication of the insights to relevant audiences.

Visual analytics draws its research directly from information visualization and adds the need for interactivity and application of knowledge discovery methods in the resulting visual representations. Information visualization can be divided into three categories: 2D visualization, which spans along two axes (e.g. bar charts, pie charts, line charts, 2D maps), 3D visualization, which spans along three axes (e.g. Google Earth visualization), and colour theory, which studies the most suitable colour pallet to improve data understanding or aid in the visual analysis of data (Silva R. A., 2017).

Particularly in the ITS domain, access to new visual analytics tools is key for understanding spatiotemporal data. These tools must be able to cope both with the spatiotemporal nature and Big Data characteristics of data coming from ITS. The following sections overview the spatiotemporal- and Big Data-related necessities of visual analytics techniques to support decision-making processes in a timely manner by the relevant stakeholders, such as public agencies and road infrastructure operators (Steed, 2017).

Besides its interactive features, spatiotemporal visual analytics must conform to the characteristics of both time and space. In the case of time, (Aigner, Miksch, Schumann, & Tominski, 2011) defined the inherent characteristics of time-oriented visualizations, to which some spatial characteristics were added (Silva R. A., 2017):

- *Frame of Reference:* Abstract data is collected without spatial context, i.e., is not connected to any spatial locations, whereas spatial data contains an inherent reference to spatial locations.
- *Number of Variables:* Univariate data comprises one data value for each time primitive, whereas multivariate data contains multiple data values for each temporal primitive.
- *Time Primitives:* Time may be categorized as instants, which do not have a duration, or intervals, which have temporal extents greater than zero.
- *Time Arrangement:* Time may be linear, meaning that time proceeds in a straight line from past to future, or cyclic, which comprises a finite set of recurring time elements, such as seasons.
- *Visualization Mapping:* Static mapping maps time to spatial locations or to visual attributes, whereas dynamic mapping maps time to time.
- *Dimensionality:* Representation of time and space dimensions can either be two- or tree-dimensional.

On the other hand, other relevant features for spatiotemporal visual analytics are (OECD/ITF, 2015):

- *Geo-spatial:* Data should be plotted on customisable maps with additional geographical information.
- *Time Resolution:* Data patterns should be observable in different time granularities (hourly, daily, weekly, etc.) by easily switching between time resolutions.
- *Animation:* Users should navigate freely across different time periods and spatial extensions and should be able to draw comparisons.
- *Interaction:* Users should have the ability to pan or zoom to particular objects and interact with them to extract additional insights.

In the context of spatiotemporal data visualization, and particularly for GRTS and spatiotemporal event points, there are three main visualization categories: multiple views, animations and *isosurfaces*, besides hybrid techniques that extend two or more of these categories and other visualization methods. Multiple views present time changes of a certain parameter on the same location, with each view corresponding to the state of a specific parameter or group of parameters at a given time. Recent examples of research works using multiple views are (Jern & Franzen, 2006; Maciejewski, et al., 2010; Plug, Xia, & Caulfield, 2011; Harris, Brundson, & Charlton, 2013). Animations show the temporal change of one or more parameters at a specific location, with the temporal change being perceived to happen in a single frame by displaying several snapshots after each other. Some examples on animated techniques for spatiotemporal data comprise (Anwar, Nagel, & Ratti, 2014; Bouattou, Laurini, & Belbachir, 2017). *Isosurface* methods map locations in the X-Y axes and time in the Z axis. Literature for this category is mainly based on space-time cubes and its variants, such as in the case of (Kraak, 2003; Gatalsky, Andrienko, & Andrienko, 2004; Kristensson, et al., 2008; Nakaya & Yano, 2010). Finally, examples of hybrid approaches are (Shrestha, 2014; Andrienko, Andrienko, Mladenov, Mock, & Pölitz, 2012; Landensberger, Bremm, Andrienko, Andrienko, & Tekušová, 2012).

The above categories describe single visualizations, but one of the major innovations that characterize spatiotemporal visual analytics tools is the ability to provide multiple interactive visualizations in one interface, each of which representing one or more dimensions and attributes from underlying data (Figure 2.3). Hence, one interactive environment may combine interactive maps, animated, temporal or timeless graphic charts or any other type of visualization, depending on the information to be presented (Kraak & Ormeling, 2010).

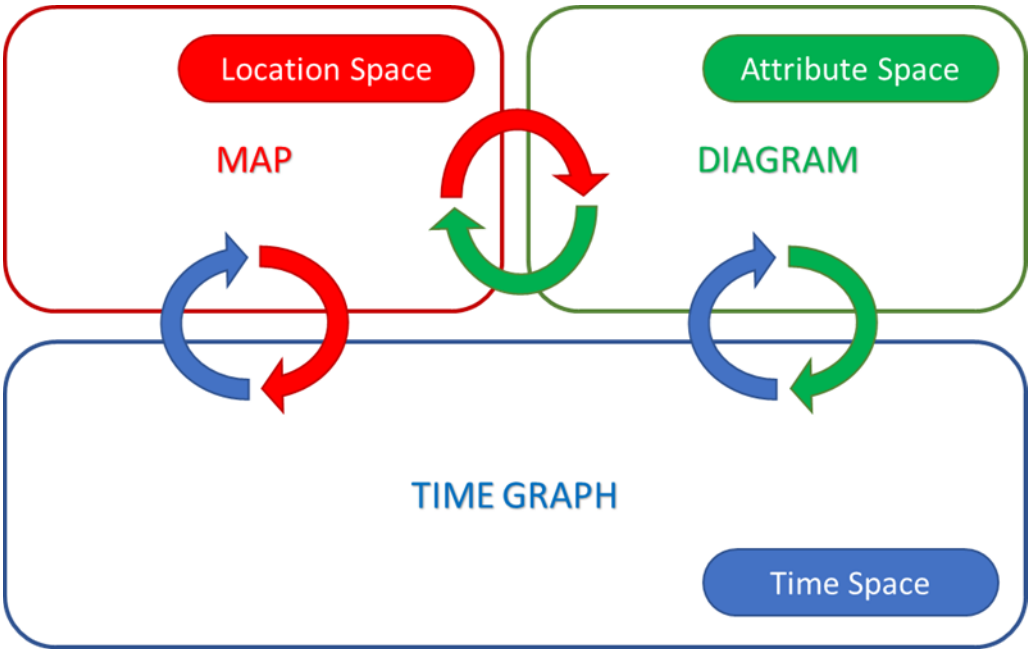


Figure 2.3 — Multi-window and synchronization example on geo-visualization interfaces (Kraak & Ormeling, 2010)

Multiple visualizations are often scattered across different windows, which are dynamically linked with each other, and their operation is based on the principle of synchronization and direct interaction with the user. So, if a visual object in one window is chosen or highlighted, the highlighting is applied to all the elements located in the other windows that are related with the selected object. One example of a research work on visual analytics based on multi-window synchronization is presented by (Chae, et al., 2012). In this work, the authors present a visual analytics approach that enables scalable and interactive social media data analysis and visualization through the exploration of abnormal events within various social media data sources. The visual analytics interface has a map for geographic reference, a temporal window choice, social media content window and topic count window.

The authors of (Robinson, Peuquet, Pezanowski, Hardisty, & Swedberg, 2017) proposed a visual analytics tool called STempo, which includes coordinated-view visualization components designed to support visual exploration and analysis of event data, and patterns extracted from those data, in terms of time, geography, and content. More recently, (Quinn & MacEachren, 2018) proposed the Crowd Lens system for OpenStreetMap (OSM), designed to help professional users of OSM make sense of the characteristics of the "crowd" that constructed OSM in specific places. Crowd Lens is an interactive Web tool with a single display containing multiple linked views, providing a filterable overview of the OSM contributor crowd in any given place, as well as a range of drill-down options to explore detailed aspects of data.

As a final example, and one of the most widely used spatiotemporal data visualization tools, GeoServer (Open Source Geospatial Foundation, 2001) is a Java-based server that allows users to view and edit geospatial data. Designed for interoperability, GeoServer allows data publishing from any major spatial data source using open standards. GeoServer has evolved to become an easy method of connecting existing information to virtual globe apps, such as Google Earth. It is primarily used for raster data but can be extended to enable visualizations of MobiTrafficBD.

2.3 Big Data

Recent years have witnessed a dramatic increase in our ability to collect data from various devices, in different formats, from independent or connected applications. This data flood has outpaced our capability to process, analyse, store and understand these datasets. 5 Exabytes (1018 bytes) of data were created by humans until 2003, while ten years later the same amount of information is created in two days (Sagiroglu & Sinanc, 2013). According to a report from the International Data Corporation (IDC), in 2011, the overall created and copied data volume in the world was 1.8 Zettabytes (≈ 1021B), which increased by nearly nine times within five years, and in 2012, the digital world of data was expanded to 2.72 Zettabytes. This figure will double at least every other two years in the near future (Sagiroglu & Sinanc, 2013; Gantz & Reinsel, 2011).

Considering Internet data, the web pages indexed by Google were around one million in 1998, but quickly reached 1 billion in 2000 and have already exceeded 1 trillion in 2008. This rapid expansion is accelerated by the dramatic increase in acceptance of social networking applications, such as Facebook, Twitter, Weibo, etc., that allow users to create content freely and amplify the already huge Web volume. Furthermore, with mobile phones becoming sensory gateways to get real-time data on people from different aspects, the vast amount of data that mobile carriers can potentially process to improve our daily life has significantly outpaced our past call data record-based processing designed only for billing purposes (Fan & Bifet, 2013).

Much influenced by technology-driven economies, in which innovation and creativity are encouraged, and by an easier access to new technologies, a menagerie of digital devices has proliferated and gone mobile—cell phones, smart phones, laptops, personal sensors—which in turn are generating a daily flood of new information. More business and government agencies are discovering the strategic uses of large databases. And as all these systems begin to interconnect with each other and as powerful new software tools and techniques are invented to analyse the data for valuable inferences, a radically new kind of "knowledge infrastructure" is materializing. A new era of "Big Data" is emerging, and the implications for business, government, society and culture are enormous.

The explosion of mobile networks, cloud computing and new technologies has given rise to incomprehensibly large amounts of information, often described as "Big Data". Big Data is a broad terminology for extremely large and complex data sets, which cannot be adequately handled by traditional data processing tools and mechanisms. The term 'Big Data' first appeared in 1998 in a Silicon Graphics slide deck by John Mashey with the title "Big Data and the Next Wave of InfraStress" (Diebold, 2012). Most common definitions for Big Data are Gartner's 3 V's definition for Big Data and De Mauro's variant for the 3 V's:

- Big data is high-volume, high-velocity and high-variety information assets that demand cost-effective, innovative forms of information processing for enhanced insight, decision making and process automation (Gartner, Inc., 2013).
- Big Data represents the Information assets characterized by such a High Volume, Velocity and Variety to require specific Technology and Analytical Methods for its transformation into Value (De Mauro, Greco, & Grimaldi, 2014).

In both definitions, Volume represents the ever-augmenting amount of data collected, Velocity corresponds to the exponential growth on data acquisition speed, and Variety stands for the growing heterogeneity of data formats and communication protocols that exist to share and spread data. More recently other definitions were presented, which included more V's. Some of them include:

- *Variability:* there are changes in the structure of the data and how users want to interpret that data (Fan & Bifet, 2013).
- *Veracity:* there is an inherent uncertainty on great volumes of data, whether it comes from low quality data, or simply flawed or untrustworthy data (Ward & Barker, 2013).

- *Value:* business value that gives organizations a competitive advantage, due to the ability of making decisions based in answering questions that were previously considered beyond reach (Fan & Bifet, 2013), (Chen, Mao, & Liu, 2014).

From the above, the most important is Value. The real value of Big Data is in the insights it produces when analysed—finding patterns, deriving meaning, making decisions, and ultimately responding to the world with intelligence. Data is not just a back-office, accounts-settling tool anymore. Rather, it is increasingly used as a real-time decision-making tool. Researchers using advanced correlation techniques can now tease out potentially useful patterns of information that would otherwise remain hidden in petabytes of data. Companies running social-networking websites conduct "data mining" studies on huge stores of personal information in attempts to identify subtle consumer preferences and craft better marketing strategies.

But how can Big Data be collected, transformed and managed, in order to extract such insights? From what we have seen so far, data may come in completely different formats at completely different time intervals and cover a panoply of data topics, completely different from one another. Furthermore, other challenges driven by a Big Data context must be considered, such as inconsistencies on the data itself, outdated data, the bandwidth of the connection and the formats that data may take.

Such inconsistencies lead to the lack of quality in data sets and, adding to that, there is still the challenge of fusing and harmonizing large volumes of data from many sources at the same time (volume to variety ratio). Furthermore, Big Data is not a singular construct; rather, it is a process spanning data acquisition, processing and interpretation. In other words, Big Data can be represented by a lifecycle shown in Figure 2.4, which starts from fast, voluminous, heterogeneous data and ends with Value, in the form of insights that support decision-making processes (OECD/ITF, 2015), through the proper interpretation and dissemination of the resulting insights and gathered knowledge.

As pointed out in (Chen & Zhang, 2014), (Garber, 2012) and (Marz & Warren, 2015), Big Data solutions often comprise the following objectives:

- Presentation of high-level architectures, which address the specific role of each technology.
- Take advantage of the application of different tools for specific tasks.
- Comprise a set of data science processes, such as statistics, data mining, machine learning, and visualization.
- Close the gap between data and analysis processes, by bringing these processes to where the data is stored, and not the opposite.

- Enable distributed processing and storage across several nodes arranged in clusters.
- Coordinate the distribution of data and processing tasks in the nodes, to assure scalability, efficiency and fault-tolerance.

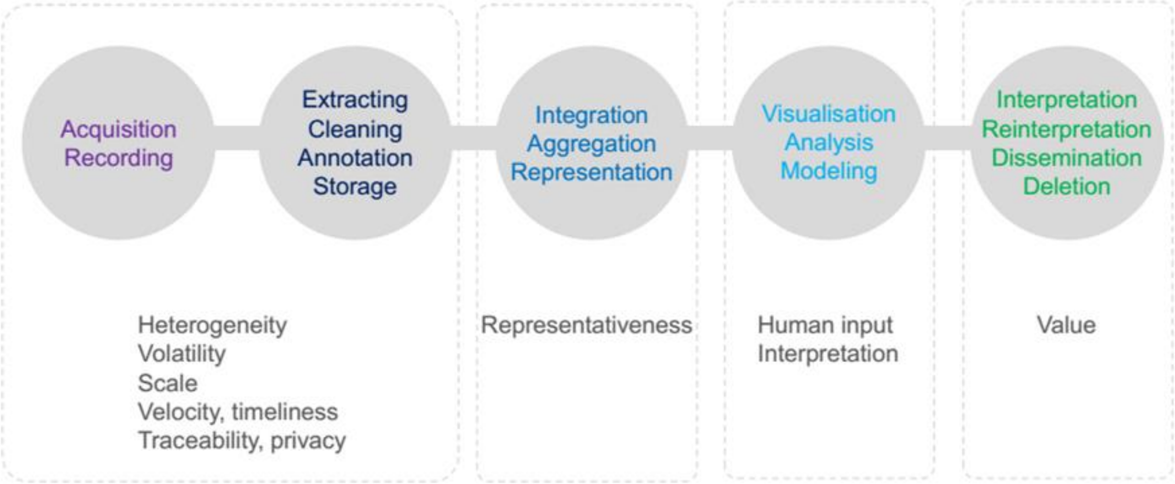


Figure 2.4 — Big Data Lifecycle (OECD/ITF, 2015)

But, in most scientific research on Big Data, works mainly focus on some component technology or solution that reflect a small part of the whole Big Data ecosystem field (Demchenko, de Laat, & Membrey, 2014). Big Data is not just related to data storage or processing, although these concepts are fundamental components for large scale data analysis and extraction of insights. Big Data is the fuel that powers all processes, sources, targets and outcomes related to data. The paradigm shift for the creation of a true Big Data ecosystem is dependent on several requirements and features, which have been the focus of a panoply of research endeavours (OECD/ITF, 2015; Marz & Warren, 2015; Manyika, et al., 2011; Costa & Santos, 2017; Demchenko, de Laat, & Membrey, 2014). Some of the most referred are:

- Big Data Infrastructure: Key aspects of the required infrastructure are the existence of necessary infrastructure components and management tools that allow fast infrastructure and services composition, adaptation, scalability and provisioning on demand for specific projects and tasks. The infrastructure should support decentralized architectures, due to the volume of data, i.e., data should be replicated and shared across multiple nodes, to support fault-tolerance, multistep processing and multi-partitioning. Data transformations should use scalable, efficient and fault-tolerant mechanisms. The results should be stored in adequate systems, such as distributed file systems or non-relational database systems. Data reads should be efficient.
- Data Management: Several key aspects of data management are required for supporting Big Data. On one hand, there is a growing need for data quality and interoperability, in the form of new methods for improving data quality and standards for data

models and formats. On the other hand, data must be managed and leveraged across the entire Big Data lifecycle, ranging from data collection and cleaning, Big Data interoperability and transformation approaches and long-term data storage and access, to integration of analytics, data mining, visualization and interaction with users. Finally, the inclusion of a clear paradigm of Data-as-a-Service, to aggregate both data and software processes for Big Data, is needed to offer Big Data solutions as a single package.

- Data Processing and Analytics: Key requirements are the automation of all data production, consumption and analysis processes including data collection, storing, processing, classification, indexing and other components of the general data analytics and mining fields, scalable and performant systems to process and analyse data either in real-time or in batches, and dynamic data-driven approaches, such as improved models and algorithms for data processing and analysis, machine learning, clustering, pattern mining, network analysis and hypothesis testing techniques and high performance data analytics. All of these requirements have to address the improvement of the scalability and processing speed for the aforementioned algorithms in order to tackle linearization and computational optimisation issues.
- Data Visualisation and User Interaction: Techniques must consider the range of data available from diverse domains as well as support user interaction for the exploration of unknown and unpredictable data within the visualisation layer. These requirements entail the need for new methods and tools for scalable visual data discovery, exploration and querying, personalized and interactive visual analytics of large-scale data and domain-specific data visualization approaches as, for instance innovative ways to visualise data in the geospatial domain, such as geo-locations, distances and space/time correlations.
- Data Security and Trust: Although not part of the presented work, data protection is also a crucial requirement of Big Data. Such requirements encompass advanced security and access control technologies that ensure secure operation of the complex research and production infrastructures and allow creating trusted secure environment for cooperating groups of researchers and technology specialists, robust anonymization algorithms and complete data protection frameworks. Although security requirements, both functional and non-functional, are of vital importance for data-driven frameworks, they will not be subject of thorough discussion within this document. Nevertheless, these and other requirements that are not covered by this work, will be revisited when needed.

These and other requirements paved the way for the creation of a panoply of Big Data Reference Models and Architectures, such as the Lambda Architecture (Marz & Warren, 2015), the National Institute of Standards and Technology (NIST) Big Data Reference Architecture (NBD-PWG, 2015) or the Big Data Value Association (BDVA) Reference Model (Big Data Value Association, 2020), which present similar and equivalent concerns. The latter, BDVA, is an industry-driven international not–for-profit organisation, with the mission of developing the Innovation Ecosystem that will enable the data and AI-driven digital transformation in Europe delivering maximum economic and societal benefit and achieving and sustaining Europe's leadership on Big Data Value creation and Artificial Intelligence. This mission is supported by the European Commission through the creation of a public-private partnership with the association to develop and implement a strategic roadmap for research, technological development and innovation in the Big Data Value and other ICT domains in Europe. The BDVA Reference Model is presented in Figure 2.5.

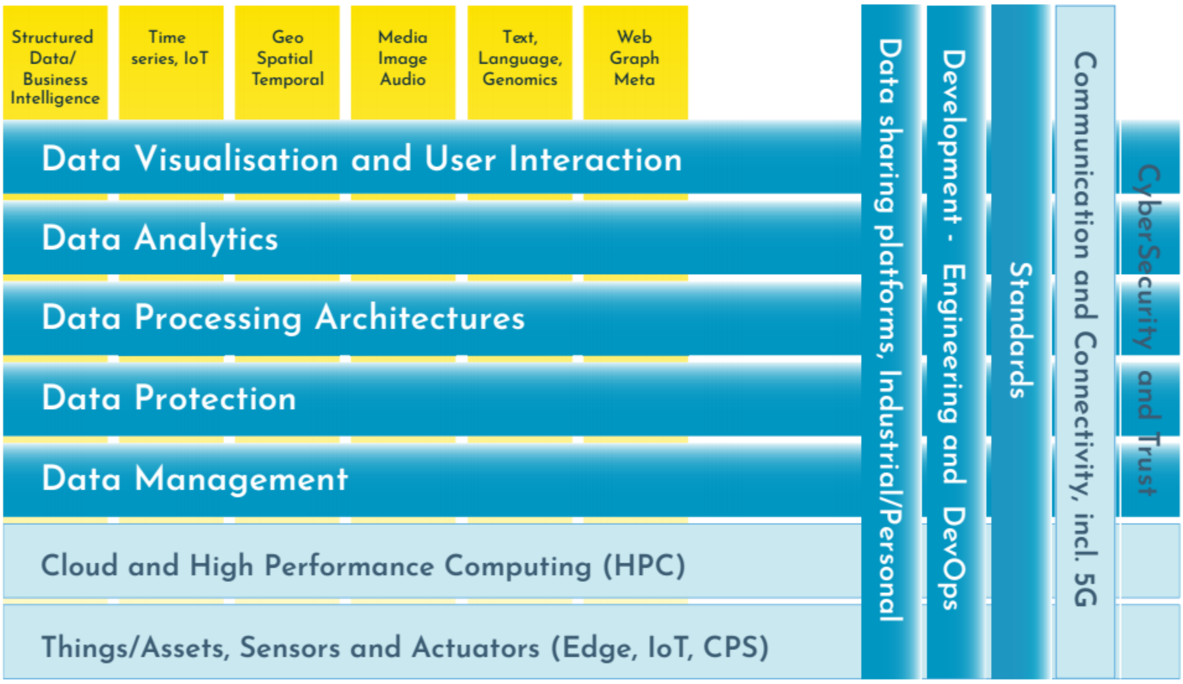

Structured Data/ Business Intelligence
Time series, IoT
Geo Spatial Temporal
Media Image Audio
Text, Language, Genomics
Web Graph Meta
Data Visualisation and User Interaction
Data Analytics
Data Processing Architectures
Data Protection
Data Management
Cloud and High Performance Computing (HPC)
Things/Assets, Sensors and Actuators (Edge, IoT, CPS)
Data sharing platforms, Industrial/Personal
Development - Engineering and DevOps
Standards
Communication and Connectivity, incl. 5G
CyberSecurity and Trust

Figure 2.5 — The BDVA Reference Model (Big Data Value Association, 2020)

The BDVA Reference Model (BDVA-RM) was chosen as the main reference architecture for this thesis work and as a basis for MobiTrafficBD framework design and development because, besides tackling the requirements brought by Big Data, it also points out the importance of data types, such as in the case of IoT data, time series and other spatial and temporal data, as shown in the yellow vertical data concerns. Nevertheless, and since this is a generic reference model, the spatiotemporal aspects of data must be added within the design phase of a framework, as will be further explored in Chapter 3.

The BDVA-RM comprises horizontal and vertical concerns. Horizontal concerns tackle specific aspects along the Big Data lifecycle of Figure 2.4, starting with data collection and ingestion, and extending to data visualization, whereas vertical concerns cover cross-cutting issues, which may affect all the horizontal concerns and may also involve non-technical aspects. Further, the BDVA-RM distinguishes between three different elements. On the one hand, it describes the elements that are at the core of the BDVA, represented in darker blue; on the other, it outlines the features that are developed in strong collaboration with hardware-related European activities, depicted in lighter blue.

Hence, Things/Assets, Sensors and Actuators (Edge, IoT, CPS) represent initiatives on hardware towards the collection of data from the physical world, Cloud and High-Performance Computing (HPC) also relates to the necessary hardware infrastructure used to process and manage highly voluminous and fast-paced Big Data, Communication and Connectivity includes the development, implementation and adoption of communication hardware technologies, including 5G and Cybersecurity comprises techniques that maintain security and trust beyond privacy and anonymisation.

Finally, six Big Data types have been identified, based on the fact that they often lead to the use of different techniques and mechanisms in the horizontal concerns, which should be considered, for instance, for data analytics and data storage: (1) Structured data; (2) Time series data; (3) Geospatial data; (4) Media, Image, Video and Audio data; (5) Text data, including Natural Language Processing data and Genomics representations; and (6) Graph data, Network/Web data and Metadata (Big Data Value Association, 2020).

Regarding horizontal core concerns in the BDVA-RM, Data Management relates to the Acquisition & Recording and the Extracting, Cleaning, Annotation and Storage steps of the Big Data lifecycle, and covers principles and techniques for data management. Data Protection does not have a direct connection to any individual step of the Big Data lifecycle; rather it should be accounted for in all the steps, since it represents privacy and anonymization mechanisms to facilitate data protection across the lifecycle. Data Processing Architectures concern optimised and scalable architectures for analytics of both data-at-rest and data-in-motion, with low latency delivering real-time analytics, and is linked to the Integration, Aggregation & Representation step of the Big Data lifecycle. Data Analytics and Data Visualisation and User Interaction horizontal concerns are mapped to the Visualisation, Analysis & Modelling step of the Big Data lifecycle, with the former representing data understanding, deep learning and the meaningfulness of data and the latter accounting for advanced visualisation approaches for improved user experience, in order to support users in value creation in the final step of the Big Data lifecycle, Interpretation, Reinterpretation, Dissemination & Deletion step.

The vertical core concerns are present across the horizontal concerns. Standards symbolise the standardisation of Big Data technology areas to facilitate data integration, sharing and interoperability. Development, Engineering & DevOps are at the basis of building Big Data Value systems, and Data Sharing Platforms comprise ecosystems for Data Sharing and Innovation Support facilitate the efficient usage of several horizontal and vertical Big Data areas, most notably data management, data processing, data protection and cybersecurity.

In the following sub-sections, the steps of the Big Data lifecycle presented in Figure 2.4 will be described, and the most recent technological and research trends in the field of Big Data will be overviewed. Finally, Section 2.4 will entail the present state of the art in MobiTrafficBD and guidelines and best practices in research works associated to this emerging field.

2.3.1 Data Acquisition and Recording

As already referred, everybody leaves data traces, like breadcrumbs, wherever they go, either voluntarily or not. This phenomenon will increase with the introduction of new sensing and data capture capabilities. Data comes in the form of phone calls, text messages, emails, social media posts, online searches or credit card purchases, among other data sources. Data is then relayed to central servers of service providers that enable such services. Making sense of this data requires familiarity with the technical aspects of data production methods as well as an understanding of how, or from whom, the data is sourced (OECD/ITF, 2015). Some key differences are listed:

- Digital vs. analogue: Data may be digital or analogue in its creation. Digital data is created specifically for use in a machine processing environment (PCAST, 2014). Digital data is produced by design to address specific needs. Some examples of digital data are GPS traces, timestamps and process logs, data produced by devices, vehicles or networked objects, data associated with access (badges, cards, RFID tags, etc.) or commercial transaction data, among others.
- Real-time vs. batch: Although data is collected in real-time, until recently data was recorded in the form of historical data, which is then stored as data batches. More recently, with the introduction of sensor technologies, the use of real-time data became common. Real-time data brings additional issues in its acquisition, since the recording methods must be ready to acquire data at great speeds, most of the times, in the order of the milliseconds.
- Unstructured vs. structured: Structured data is characterized by its definition through a well-established data model, thus being easier to analyse. Structured data is usually stored in relational databases, with discrete fields and enabling quick aggregation of data from several locations, or tables, in the database. Some examples are GPS latitude

and longitude data and commercial transaction data, among others. On the other hand, unstructured data has an inherent internal structure but is not organized in predefined data model. This results in irregularities and ambiguities that make it difficult to understand using traditional programs as compared to data stored in structured databases. Examples of unstructured data are emails, social media content, video and audio streams.

2.3.2 Extract, Clean, Annotate and Store

When selecting a data source to analyse, one crucial factor is the quality and scope of the data set produced by the source. Data sets resulting from different sources tend to be messy (i.e., heterogeneous), dirty (i.e., comprising missing, incorrect, mislabelled or potentially forged or fake data) and incompatible (i.e., not aligned with other data sources). One characteristic that is part of the above issue is the structural level of data, with some data sets being highly structured, which eases and quickens the analysis, whereas other data sets are highly unstructured, which makes them more difficult and time consuming to analyse. Recent advances in data processing and analysis enable the elicitation of new insights by mixing both structured and unstructured data for analysis, but these processes require efficient data cleaning (OECD/ITF, 2015).

Data cleaning tasks are very time consuming and not trivial since the preparation of data for analytical use involves several processes. In the case of unstructured data, this type of datasets must be correctly interpreted, contextualized, categorised and consistently labelled, while for structured data, datasets must be parsed and cleaned from missing and incorrect data. Furthermore, interoperability-based transformations for datasets that are contextually similar, but inherently different, in terms of structure and format, need to be applied for both unstructured and structured data. In fact, it is estimated that data cleaning tasks, manual or automatic, account for 50% to 80% of data scientists' time (Endel & Piringer, 2015).

These aspects bring particular importance to ETL (Extract-Transform-Load) processes when data needs to be loaded from sources into a harmonized data repository. ETL software houses have been extending their solutions to provide big data extraction, transformation and loading between big data platforms and traditional data management platforms, describing ETL now as "Big ETL" (Bala, Boussaid, & Alimazighi, 2016), i.e., Big ETL is the adaptation of ETL methodologies and techniques to Big Data, which is a relatively new research field. In (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018), an overview of some Big ETL approaches in the literature is presented.

2.3.3 Integration, Aggregation and Representation

A wide range of methods and approaches have been drawn from the expertise of several different fields, such as statistics, computer science, applied mathematics and economics, to manipulate, aggregate, process and visualize Big Data. Within the context of mobility, transportation and road traffic, the techniques for data analysis are divided into, but not limited to, Data Fusion, Data Mining, Optimisation and Visualization.

Data fusion is comprised by approaches that merge and consolidate data coming from multiple data sources, such as location data produced by GPS hardware and telecommunications networks (e.g., via cell triangulation). Data mining, as already described, is a set of techniques that are used to extract patterns and insights from large datasets, such as the inherent relationships between single nodes in a road network. Optimisation refers to methods to reorganise complex systems and processes, enabling improvements in their performance regarding one or more attributes, as in the case of fuel efficiency or travel times. Finally, visualization, which will be subject of sub-section 2.3.4, is based on the generation of images, maps, diagrams or animations to communicate the results of data analysis processes. These techniques are often used throughout the data analysis process, enabling understanding of information by humans.

The first category, data fusion, comprises techniques that enhance the representations of reality that can be then used for data mining, by matching and aggregating data coming from heterogeneous datasets and different data sources (Figueiras, et al., 2016), being a crucial processing step when dealing with inputs from multiple sensor platforms. But high-level data fusion tasks, which merge several heterogeneous, unstructured data inputs, such as in the case of analogue sensor inputs, is still challenging and constitutes a strong research focus, as explained in (Khaleghi, Khamis, & Karray, 2013). Examples of preliminary works on this research challenge are presented in (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018; Figueiras, et al., 2016; Figueiras, et al., 2016). Generally, data fusion's goal is to fuse data accuracy and semantics, and to solve issues related to data resolution and granularity. One challenge in data fusion is to extract shared features across multiple datasets, which were created for different purposes. Hence, data integration and fusion entail the matching and merging of datasets based on their shared attributes, while each dataset is retained. Integration and fusion methods enable knowledge discovery through contextual data analysis.

2.3.4 Visualisation, Analysis and Modelling

Although Big Data visualisation will be presented in sub-section 2.3.5.3, it is necessary to overview the analysis and modelling procedures associated with Big Data. The advantages brought by the evolution of Big Data technologies and distributed processing environments allow for the exploitation of very large datasets to extract relevant knowledge and insights. Classical approaches, such as statistical or traditional optimization techniques are still important but bring about data processing bottlenecks in the case of voluminous and high-velocity data. In the last years, knowledge discovery methods have suffered a significant evolution in terms of their ability to handle Big Data (OECD/ITF, 2015), such as in the case of data mining and modelling techniques.

Several works in Big Data Mining and Deep Learning, which take advantage of distributed processing environments to model and extract insights from large volumes of data, are available in the literature. Particularly for Big spatiotemporal datasets, some examples were already presented in former sections, and refer to stream processing (Andrienko, Andrienko, Fuchs, & Wood, 2017; Galić, 2016; Pan, Demiryurek, Banaei-Kashani, & Shahabi, 2010; Polson & Sokolov, 2017; Lin, Keogh, Lonardi, & Chiu, 2003; Giao & Anh, 2015; Giao & Anh, 2016; Guo, Huang, & Williams, 2015), Deep Learning (Jain, Zamir, Savarese, & Saxena, 2016; Wang, Gu, Wu, Liu, & Xiong, 2016; Chambon, Galtier, Arnal, Wainrib, & Gramfort, 2018; Qiu, Ren, Suganthan, & Amaratunga, 2017; Ryu, Noh, & Kim, 2017), application of known Big Data processing technologies and methods (Eldawy, Mokbel, Alharthi, Tarek, & Ghani, 2015; Alarabi, Mokbel, & Musleh, 2018; Fan & Bifet, 2013; Chen & Zhang, 2014), GPU-based and distributed processing (Doraiswamy, Vo, Silva, & Freire, 2016; Luo, Niu, & Zhang, 2018) and evolutions of classical approaches to cope with big datasets (Kim, Park, & Chu, 2001; Movchan & Zymbler, 2015).

2.3.4.1 Distributed Processing Tools

Distributed computing is a field of computer science that studies distributed systems. A distributed system is a software system in which components located on networked computers communicate and coordinate their actions by passing messages. The components interact with each other to achieve a common goal. Three significant characteristics of distributed systems are: concurrency of components, lack of a global clock, and independent failure of components (Coulouris, Dollimore, Kindberg, & Blair, 2012). There are three main types of distributed computing systems (Costa C. F., 2019):

- Batch Processing: This type of processing operation entails time-intensive data processing tasks, often applied on high-volume batch data. It involves latencies in the order of minutes or hours; hence they are better suited for running as background processes without the need for direct user intervention. Some examples of batch processing are the periodic cleaning, harmonization, enrichment and aggregation tasks of high volumes of historical data, the creation of complex reports and execution of complex *ad hoc* queries, the training of data science models, such as predictive or classification models, and other intensive data mining and machine/deep learning tasks.

- Stream Processing: This type of processing operation comprises fast processing tasks, with latencies in the order of milliseconds to a few seconds, since it is specifically designed for near real-time data throughput. In this case, examples are streaming data cleaning, harmonization and enrichment, as well as swift analytics and processing tasks, such as data aggregation in micro batches or fast detection of anomalies and patterns. Micro batches are very useful when performing specific streaming operations as, for instance, small aggregations, sliding window operations, data fusion operations, to find trends in the data, or data science models' application tasks. Micro batches are really small batches of data records that are used to optimize collection, storage and processing tasks, by handling a few records at a time, instead of handling them individually, improving the overall throughput of the data flow and enabling the optimization of database insertion by performing data insertion operations on micro batches, instead of on individual streaming records.
- Interactive Processing: This type of processing operation is focused on user interaction and is responsible for answering direct user queries. Like in the case of stream processing, query execution times should be in the order of milliseconds to a few seconds, depending on the volume of data in the answer, because it is expected that answers to user queries should be as swift as possible, due to direct user interaction. Some database solutions provide data distribution and organization strategies to tackle this level of latencies, even when data volumes increase exponentially, such as data denormalization, partitioning or inter-storage (Costa C. F., 2019).

When the term Big Data became a buzzword, it applied mainly to batch processing, because companies had lots of historical data already on their databases to process. But soon, companies realized that using distributed computing to process real-time streams of data was a necessity (Chen & Zhang, 2014). Below, we present three of the trendiest distributed processing technologies available today, although there are several other options (e.g., Apache Flink (The Apache Software Foundation, 2014)):

- Apache Hadoop (batch processing): The Apache Hadoop [193] software library is a framework that enables distributed processing of large data sets across clusters of computers using simple programming models, based on the MapReduce paradigm [116]. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. From Apache Hadoop, an entire ecosystem of tools has been created, to cope with the specificities of Big Data (Figure 2.6). The most well-known of such tools, in both academia and industry, will be presented in this and the following sections.

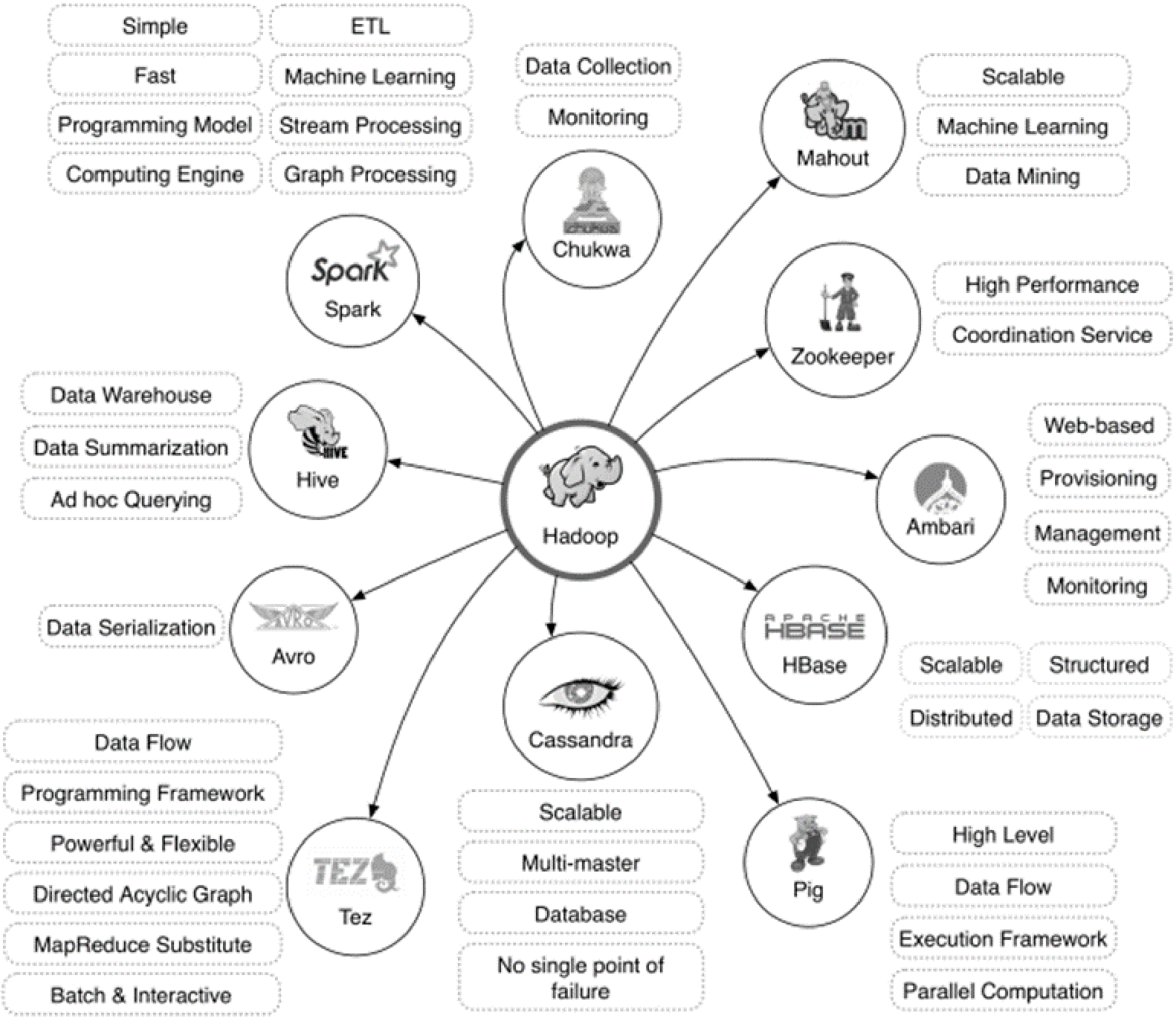


Figure 2.6 — The rich ecosystem of Apache Hadoop (Costa & Santos, 2017)

- Apache Spark (online-stream and offline-batch processing): Apache Spark (The Apache Software Foundation, 2018) is a fast and general-purpose cluster computing system. It provides high-level APIs in Java, Scala, Python and R, and an optimized engine that supports general execution graphs. It also supports a rich set of higher-level tools including Spark SQL for SQL and structured data processing, Spark MLlib for machine learning, Spark GraphX for graph processing, and Spark Streaming.
- Apache Storm (online-stream processing): Apache Storm (The Apache Software Foundation, 2018) is an open source distributed real-time computation system. Storm makes it easy to reliably process unbounded streams of data, doing for real-time processing what Hadoop does for batch processing. Storm is used for real-time analytics, online

machine learning, continuous computation, ETL, and more. It is scalable, fault-tolerant, guarantees data is processed and is easy to set up and operate.

2.3.5 Technological and Research Trends

The inferential techniques being used on Big Data can offer great insights into many complicated issues, in many instances with remarkable accuracy and timeliness. The quality of business decision-making, government administration, scientific research and much else can potentially be improved by analysing data in better ways (Bollier, 2010). Hence, Big Data is expected to have a great impact in several sectors. For instance, the McKinsey Global Institute specified the potential of Big Data in five main areas (Manyika, et al., 2011):

- Healthcare: clinical decision support systems, individual analytics applied for patient profile, personalized medicine, performance-based pricing for personnel, analyse disease patterns and improve public health.
- Public Sector: creating transparency by providing accessible related data, discover needs, improve performance, customize actions for suitable products and services, decision making with automated systems to decrease risks, innovating new products and services.
- Retail: in store behaviour analysis, variety and price optimization, product placement design, improve performance, labour inputs optimization, distribution and logistics optimization and web-based markets.
- Manufacturing: improved demand forecasting, supply chain planning, sales support, developed production operations, web search-based applications.
- Personal Location Data: smart routing, geo-targeted advertising or emergency response, urban planning, new business models.

In fact, with the advent of the Internet of Things (IoT), the potential of Big Data gets even more importance across sectors. In the IoT paradigm, an enormous amount of networking sensors is embedded into various devices and machines in the real world. Such sensors may collect various kinds of data, such as environmental, geographical, astronomical or logistics data. Mobile equipment, transportation facilities, public facilities, and home appliances could all be data acquisition equipment in IoT (Chen, Mao, & Liu, 2014). IoT brought to light “Smart” areas or scenarios, such as Smart Cities, Smart Buildings, Smart Grids, Smart Mobility, etc., in which the Big Data hype is also present, and is considered an important factor for Smart applications to flourish.

From all data-gathering areas and contexts, ITS-related data must be one data type in which all Big Data characteristics are present, from large quantities of data, captured every day in intervals ranging from hours to seconds, varying from real-time or simulated traffic data, floating-car and GPS data, weather and traffic forecasting and history data among several others. Big transportation data is also highly variable, since it still presents lots of inconsistencies, such as intermittent sensor data (traffic, parking spots, etc.) or outdated data from transportation providers (schedules, stops, etc.). These inconsistencies lead to a low veracity ratio, since the quality of data is always changing. Finally, complexity refers to Volume-to-Variety ratio, and to the difficulties to fuse large amounts of data coming from several different sources.

The following sections describe the most common Big Data technological tools to process, store and analyse Big Data, and present the state-of-the-art of the application of such technologies on whole systems, architectures and frameworks, designed to cope with the challenges posed by ITS scenarios.

2.3.5.1 Big Data Storage Technologies

Regular SQL Engines and databases are not built to support, manage and process today’s Big Data. Big Data refers to petabytes of information, and to interact with this amount of data through regular SQL solutions is difficult. The first solution for this issue was the creation of the NoSQL (Not Only SQL) concept (Schmid, Gálicz, & Reinhardt, 2015): A NoSQL database provides a mechanism for storage and retrieval of data that is modelled in means other than the tabular relations used in relational databases. Motivations for this approach include simplicity of design, presumed better "horizontal" scaling to clusters of machines, which is a problem for relational databases. NoSQL databases are increasingly used in big data and real-time web applications.

Lately, several solutions based on SQL but called SQL-on-Hadoop engines (Floratou, Minhas, & Özcan, 2014) are coming to the spotlight. With SQL-on-Hadoop technologies, it's possible to access Big Data stored in Hadoop by using the familiar SQL language. Users can plug in almost any reporting or analytical tool to analyse and study the data. Here are some examples of the latest NoSQL and SQL-on-Hadoop solutions:

- Apache Hive (SQL-on-Hadoop): The Apache Hive (The Apache Software Foundation, 2011) data warehouse software facilitates querying and managing large datasets residing in distributed storage. Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL. This language also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL.
- Cloudera Impala (SQL-on-Hadoop): Impala (The Apache Software Foundation, 2015) is a fully integrated analytic database architected specifically to leverage the flexibility

and scalability strengths of Hadoop - combining the familiar SQL support and multi-user performance of a traditional analytic database with the rock-solid foundation of open-source Apache Hadoop and the production-grade security and management extensions of Cloudera Enterprise.

- Apache Cassandra (NoSQL): Apache Cassandra (The Apache Software Foundation, 2016) is an open-source distributed database management system designed to handle large amounts of data across many commodity servers, providing high availability with no single point of failure. Cassandra offers robust support for clusters spanning multiple datacentres, with asynchronous masterless replication allowing low latency operations for all clients. Cassandra also places a high value on performance.
- MongoDB (NoSQL): MongoDB (MongoDB, Inc., 2015) (from humongous) is a cross-platform document-oriented database. Classified as a NoSQL database, MongoDB eschews the traditional table-based relational database structure in favour of JSON-like documents with dynamic schemas (MongoDB calls the format BSON), making the integration of data in certain types of applications easier and faster.

Another category of databases suited for Big Data is coined NewSQL, which is used to classify database solutions aggregate the advantages of the classical relational model with the concepts of scalability and fault-tolerance present in NoSQL (Grolinger, Higashino, Tiwari, & Capretz, 2013). These databases are based on distinct assumptions and architectures, when compared with traditional Relational Database Management Systems (RDBMS), but still support relational models and use SQL as their native query language. Some of the most used NewSQL databases are presented below:

- Clustrix DB: Clustrix DB (MariaDB, 2018) is distributed relational database built to scale horizontally, by adding cores and servers. ClustrixDB can execute one query with maximum parallelism and many simultaneous queries with maximum concurrency, due the application of the query fragments' concept. Query fragments are compiled to machine code, run on the nodes that contain the data and can perform operations from a very rich set available. Query fragments may be different components of the same query or parts of different queries. The result is the same: massive concurrency across the cluster that scales with the number of nodes.
- NuoDB: NuoDB (NuoDB, Inc., 2018) is a technologically advanced, distributed SQL database for cloud- and container-based environments that appears as a single, logical SQL database to the applications. Its two-layer, peer-to-peer architecture retains strict transactional consistency and can be deployed across multiple data centres or even

different clouds and is optimized for in-memory speeds, continuous availability, and elastic scale-out.

- VoltDB: VoltDB (VoltDB, Inc., 2019) is also based on SQL and the relational model, but it is tuned for context, real-time analytics understanding, low latency and strong consistency. VoltDB relies on horizontal partitioning down to the individual hardware thread to scale, k-safety (synchronous replication) to provide high availability and is packed with real-time stream processing capabilities to cope with the data velocities of the upcoming 5G networks.

Finally, there are database solutions that cope with Big Data challenges by focusing on specific types of data, as in the case of time series databases. A time series database is specifically optimized for time-stamped or time series data and for measuring changes in the data over time. Time series databases handle metrics and events or measurements that are timestamped. Time series databases have key architectural design properties that make them very different from other databases. These include time-stamp data storage and compression, data lifecycle management, data summarization, ability to handle large time series dependent scans of many records, and time series aware queries (InfluxData Inc., 2019). Some of the most common time series databases are:

- InfluxDB: InfluxDB (InfluxData, Inc., 2013) is an open-source time series database that supports a very large set of programming languages and operating systems and is optimized for heavy writing load and works amazingly well with concurrency. It is built on NoSQL concepts and allows for quick database schema modifications.
- TimeScaleDB: TimescaleDB (Timescale, Inc., 2017) is an open-source time-series database built for fast data ingestion and complex queries. It is fully SQL compliant while scaling in ways previously reserved for NoSQL databases. It is built as a PostGreSQL extension. It allows for high rates on data write, using batched commits, in-memory indexes, transactional support, support for data backfill and parallelized operations across servers.
- Apache Druid: Apache Druid (The Apache Software Foundation, 2019) is a real-time analytics database designed for fast analytics on large data sets. Druid uses column-oriented storage and enables massive parallel processing operations, since it can be deployed in clusters of tens to hundreds of servers and can offer ingest rates of millions of records per second, retention of trillions of records, and query latencies of sub-second to a few seconds.

2.3.5.2 Big Data Mining Suites and Libraries

There are several off-the-shelf technology suites and libraries to handle the Big Data lifecycle. Business Intelligence (BI) software is commonly used to analyse and visualize the data. This type of software also provides reporting, data discovery, data mining and dashboarding functionality. While most of the Cloud service providers offer their own Business Intelligence tools, many independent suites and libraries are available. Some of the most successful open-source integrated environments with ETL and BI capabilities are listed below:

- Talend Open Studio: Talend Open Studio (Talend, 2006) is a versatile set of open-source products for developing, testing, deploying and administrating data management and application integration projects. For ETL projects, Talend Open Studio for Data Integration delivers a rich feature set including a graphical integrated development environment with an intuitive Eclipse-based interface. The advanced ETL functionality including string manipulations, automatic lookup handling, and management of slowly changing dimensions and support for ELT (extract, load, and transform) as well as ETL, even within a single job.
- RapidMiner: RapidMiner (RapidMiner, Inc., 2013) is one of the leading data mining software suites. RapidMiner supports all steps of the data mining process from data loading, pre-processing, visualization, interactive data mining process design and inspection, automated modelling, automated parameter and process optimization, automated feature construction and feature selection, evaluation, and deployment. RapidMiner can be used as stand-alone program on the desktop with its graphical user interface (GUI), on a server via its command line version.
- GeoKettle ETL: GeoKettle (Open Source GeoBI, 2010) is a powerful, metadata-driven spatial ETL tool dedicated to the integration of different data sources for building and updating geospatial databases, data warehouses and services. GeoKettle enables the extraction of data from data sources, the transformation of data to correct errors, make some data cleansing, change the data structure, make them compliant to defined standards, and the loading of transformed data into a target Database Management System (DBMS) in OLTP or OLAP/SOLAP mode, GIS file or Geospatial Web Service.
- Pentaho Business Analytics: The Pentaho Business Analytics platform (Hitachi Vantara Corporation, 2017) covers the entire big data life cycle, from data extraction and preparation of diverse data to scalable processing on Spark and Hadoop, leading to end-to-end analytics solutions. Pentaho Business Analytics provides a spectrum of analytics for all user roles, from visual data analysis for business analysts to tailored dashboards for executives. Pentaho is fast to deploy, easy to use, and purpose-built for Big Data analytics.

Furthermore, several libraries and languages are available to bring Data Mining methods to the Big Data application area. These are often built upon existing Big Data processing engines, enabling parallelization of the Data Mining tasks. Some examples of these libraries are presented below:

- Apache Spark MLlib: MLlib (The Apache Software Foundation, 2018) is Apache Spark's Machine Learning library. MLlib contains high-quality algorithms that leverage iteration and can yield better results than the one-pass approximations used on MapReduce. The algorithms in MLlib feature common learning algorithms such as classification, regression, clustering, and collaborative filtering, featurization methods, tools for constructing machine learning pipelines, and other utilities, such as linear algebra, statistics and data handling techniques.
- Apache Mahout: Mahout (The Apache Software Foundation, 2014) is a distributed linear algebra framework that offers a mathematically expressive Domain Specific Language, allowing to quickly implement custom algorithms. It has implementations of clustering, classification, collaborative filtering and frequent pattern mining. It can run on top of Apache Hadoop or Spark.
- Weka: Weka (The University of Waikato, 2005) is a free and open-source machine learning and data mining software written in Java, containing a collection of machine learning algorithms for data mining tasks, as well as tools for data preparation, classification, regression, clustering, association rules mining and visualization. Weka is not scalable by default, but it has been extended with several connectors for scalable Big Data stores and packages for distributed processing engines (e.g., distributedWekaBase, distributedWekaHadoop and distributedWekaSpark).
- MOA and SAMOA: MOA (The University of Waikato, 2010) is a stream data mining software to perform data mining in real-time. It has implementations of clustering, classification, regression, frequent item set mining, and frequent graph mining. SAMOA (The Apache Software Foundation, 2015) is based on MOA and enables development of new ML algorithms without directly dealing with the complexity of underlying distributed stream processing engines.
- R Language: Developed by Bell Labs, R (The R Foundation, 2000) is a programming language for statistical values, complex data and graphical information that can handle large volumes of data. R provides graphical techniques and has many tools to perform data analysis. Like Weka, R can be extended with several connectors for scalable Big Data analysis and packages for distributed processing engines (e.g., SparkR, RHadoop, RHive).

2.3.5.3 Big Data Visualisation

With the exponential growth of datasets and the spread of data across society, classical data visualization approaches are not enough. As data sets gets larger and efforts to exploit this data seeks to reach more people, the language of data visualisation must adapt and improve (OECD/ITF, 2015). The visualization of Big Data entails a complete revision of traditional approaches, mainly due to the volume and speed at which data must be analysed. Advanced data visualizations are necessary enabling the extraction of real value from Big Data, by providing the capabilities to scale to millions of data points, handle multiple data types, and joining appearance and functionality (Costa & Santos, 2017). Leveraging Big Data visualization is challenging due to its inherent characteristics. In these visualizations, data from multiple sources is typically integrated into a single picture.

The authors of (Wong, Shen, Johnson, Chen, & Ross, 2012) identified the main challenges for extreme-scale visual analytics. Firstly, classical approaches that apply visual analytics on stored data are not feasible when datasets reach the petabyte scale. Hence, in situ visual analytics, i.e., performing as much visual analyses as possible over in-memory data, would greatly improve the costs in terms of input/output operations and disk use. Second, visual analytics interfaces need a revamp to account for the growing differences in data sizes, which are ever expanding, and the cognitive capabilities of humans, which remain unchanged. This means that better methods for data fusion, reduction, aggregation and summarization are crucial for humans to extract insights from great volumes of data.

Third, the presentation and visualization techniques for big data volumes must account for the data reduction pointed out in the last challenge to still present aggregated and fused data in ways that can transmit the reality of the whole data being represented. Moreover, even if ever-larger displays can be built, human vision accuracy has limitations, also limiting the effect of large-screen technologies for visual analytics. Fourth, both classical databases and algorithms were not designed to scale above a certain threshold (e.g., exabytes for new database technologies). In the case of databases, even cloud-based solutions may not meet the needs for extreme-scale visual analytics and cloud storage costs are still higher than traditional hard drive storage. For algorithms, many of these are computationally intensive and are mostly focused on post processing of data already available in memory or in disks. New methods need to be scalable, visually efficient and must be integrated with automatic learning so that the visualization output is highly adaptable. Fifth, data movement will become the most expensive component of visual analytics since computing costs continue to decrease. One of the main challenges for large-scale computing and visual analytics has to do with the efficiency of communication networks. Finally, as data sets' volumes continue to grow, the ability to process them will be severely limited. Many analytics tasks will rely on data subsampling to overcome the real-time constraint, introducing even greater uncertainty. Uncertainty quantification and visualization will be a big challenge, since new analytics methods will have to cope with the incompleteness of extreme-scale data, while considering data as distributions.

Particularly in ITS, the application of real-time and massive scale visualisation tools to analyse traffic has increased in recent years, due the availability of data collected by traffic management and road infrastructure operators. Perhaps the best-known examples are online map services (Google Maps, OpenStreetMap, etc). These services use color-coded paths to indicate traffic speeds derived from road sensors and GPS-enabled vehicles and mobile devices (OECD/ITF, 2015).

As described by (Andrienko, et al., 2010), three main approaches are being adopted to cope with the challenges posed by Big Data visualization. One approach is to use data aggregation and summarization techniques prior to graphical representation to modify the number of data points directly depicted. The second approach relies on more sophisticated computational methods based on data mining, in order to (partially or fully) automatically extract insights, features and patterns, specific to the problem at hand, before presenting the data through visualization techniques. This approach could apply data mining algorithms over aggregated and summarized data, thus capitalizing on advances in direct depiction. The last approach is based on the development of data projections that shift objects from their geographic locations to fill the visualization space in a more effective way.

2.4 MobiTrafficBD

The swift, ongoing sensorisation of the world, from our smartphones to *in-situ* sensors, has brought the two topics already covered in this chapter closer: Spatiotemporal Data and Big Data. This closeness is based on the spatiotemporal nature of our world, with more and more data, collected by sensing devices, possessing a spatiotemporal signature. The aggregation of Spatiotemporal Data with Big Data is often coined in the literature as Big Spatiotemporal Data.

As clearly presented in (Yang, Clarke, Shekhar, & Tao, 2020), Big Spatiotemporal Data has become an important research topic with a ten-year history (with the first publications starting in 2009), spanning several research disciplines, such as GIS (Wang, Zhong, & Wang, 2019; Song, Wang, & Zomaya, 2017), cloud computing (Yang, Yu, Hu, Jiang, & Li, 2017; Yang, et al., 2015), data processing (Yang, et al., 2015), Data Mining (Xu, Deng, Demiryurek, Shahabi, & van der Schaar, 2015; Vatsavai, et al., 2012), Deep Learning (Cao, et al., 2018; Wang, et al., 2017; Polson & Sokolov, 2017) and data visualization (Vatsavai, et al., 2012; Cao, et al., 2018), and with applications in many societal and economic sectors, as in the case of tourism (Baptista e Silva, et al., 2018), mobility and traffic (Xu, Deng, Demiryurek, Shahabi, & van der Schaar, 2015; Batran, Mejia, Kanasugi, Sekimoto, & Shibasaki, 2018), climate (Li, et al., 2013; Li, et al., 2017), environment (He, Gu, Wang, & Zhang, 2017; Zhu, et al., 2018), urbanism and land use (Li, Ye, Lee, Gong, & Qin, 2017; Comber & Wulder, 2019), just to name a few.

Therefore, spatiotemporal data is, at the same time, a prolific research topic and a fuzzy one, to say the least. It is fuzzy in the sense that spatiotemporal data is related to so many research areas and is applied in so many economic sectors, that it is hard to have a clear picture of the directions and approaches that are available in the literature. Moreover, research endeavours are more than often directed to application niches, such as in the case of traffic, mobility and ITS. Hence, the focus will fall upon the main theme for this document: the analysis and lifecycle management of large-scale GRTS (e.g., traffic sensor data) and ST event (e.g., accident or traffic jam data) data for traffic- and mobility-related applications.

Although mobility- and traffic-related Big Spatiotemporal Data (MobiTrafficBD), in the form of GRTS and ST events, may be considered a sub-field of the broader Big Spatiotemporal Data research topic, it is also a prolific topic, with several contributions in the areas of Data Mining, Machine Learning and Deep Learning and an emphasis on the prediction of traffic flow and congestion. The rest of this section will go through the more recent works regarding MobiTrafficBD and the general guidelines that can be extracted from the commonality analysis between such works.

2.4.1 Representation, Modelling and Interoperability

The representation and modelling of MobiTrafficBD is the first important factor when dealing with this kind of data. On one hand, MobiTrafficBD models and representations should account for the different attributes and heterogeneous nature of data captured by traffic sensors (sensors from different manufacturers may capture diverse attributes in many granularities, e.g. one sensor may capture travel speed for the entire road segment, in one-minute intervals, while another may capture road occupancy for each lane in the road segment, in five-minute intervals) or events reported by RITMOs (each RITMO may report traffic events in different ways, collecting different attributes), while providing a single, homogeneous model or representation that can handle the heterogeneity and diverseness of such data, by harmonizing different data sets into such single model. On the other hand, these data models should be lightweight, to meet the storage requirements of Big Data, human-readable, for easy understanding of the data, and standardized.

Let us consider a less recent work that paves the way for traffic sensor and traffic event data modelling and representation. In (America's Advanced Traveller Information Systems Committee, 2000), the authors propose some guidelines for data quality in the representation of data related to Advanced Traveller Information Systems, namely traffic sensor data, incident and event reports and road and environmental station data. As already referred, traffic sensor and road and environmental station data are defined as GRTS and incident and event reports as ST events. At the time, the author did not define the data attributes for road and environmental station data, due to the lack of maturity and consensus support as traffic sensor data and incident and event reports. Therefore, Figure 2.7 was adapted for the specific needs of this work.

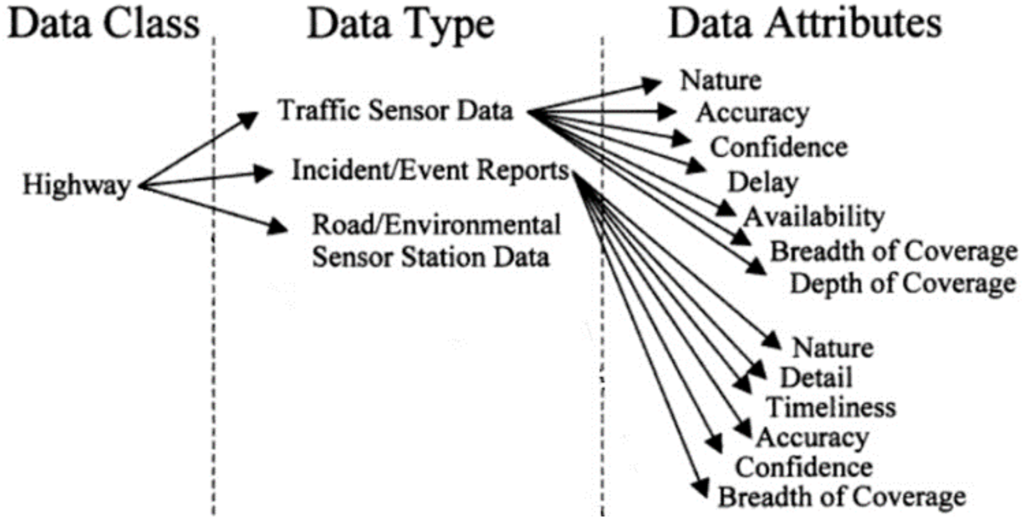


Figure 2.7 — Data types and attributes for traffic sensor and event data. Adapted from (America's Advanced Traveller Information Systems Committee, 2000)

The figure represents the data types and their respective attributes. Traffic sensor data is based on measurements used to categorize the flow of vehicles at a particular point or over a specific road or highway segment. Such measurements may collect speed, travel time, road volume and occupancy data or any other information related to traffic flow. This data type can be collected through a panoply of detection systems, such as loop detectors, microwave, sonic or infrared sensors or automatic vehicle identification, just to name a few.

Some of the attributes presented are vital for data quality assessment purposes, but not so much for the actual representation of the data. Nevertheless, an overview of such attributes is in place. Nature corresponds to the data parameter being collected. Four parameters are commonly collected through traffic sensors:

- *Volume*: the actual number of vehicles observed passing a point during a given time interval (i.e., the data collection interval).
- *Occupancy*: the ratio between the time of permanence of vehicles in the detection point and the time of sampling (i.e., the data collection interval).
- *Speed*: the average rate of motion, as distance per unit of time.
- *Travel time*: the elapsed time for a vehicle to traverse the road or highway segment.

From these parameters, volume, occupancy and speed are often collected at a single point in the roadway and are defined as point data. Travel time is collected over a section of roadway and is defined as section data.

Accuracy accounts for the matching between what is measured and the actual conditions on the road, since all traffic sensors are prone to inaccuracies, due to faulty measurements or to several other conditions (e.g., weather, interreferences or occlusion). This attribute is often represented as a percentage. Confidence describes the degree of belief on the quality of communicated data. Delay refers to the time between data collection and its availability for use in some application. Availability has to do with the amount of collected data that is made available. Finally, coverage relates to the span of road infrastructure in which data is being collected.

Regarding the incidents and event reports, they are characterized by descriptions of planned or unplanned occurrences that may affect traffic conditions. This type of data is usually manually inserted into a database by RITMOs or extrapolated from other types of data, such as traffic sensor data (e.g., high occupancy and low speed may be related to an accident and consequent traffic jam). In terms of the nature attribute of this data type, the possible events that can be collected are crashes, breakdowns or other unplanned vehicle stoppages, planned and emergency roadworks and maintenance, special planned events, general road and weather conditions or disasters.

The detail attribute describes additional data that is associated to an event, such as the reason of the event, location, severity, impact, status, advice and suggestions for travellers impacted by the event, duration and starting time. The timeliness attribute relates to the time it takes to detect, verify and update the status of incidents or events. Accuracy, confidence and coverage have the same definitions as in the case of traffic sensor data.

Although the attributes for road and environmental station data are not present in Figure 2.7, they are also overviewed in (America's Advanced Traveller Information Systems Committee, 2000). Road and environmental station data refer to data collected from a wide array of sensor stations, such as weather, roadway, surface and air quality conditions' monitoring stations. The nature of data collected by these stations may be elevation and atmospheric pressure, wind, temperature, humidity and precipitation, radiation and visibility data, air and water quality. Finally, coverage also relates to the definition of coverage for traffic sensor data.

MobiTrafficBD representation and modelling approaches are not often present in the literature, since the majority of the available literature works focus on the algorithms to be used and frameworks and architectures deployed in the analysis of MobiTrafficBD, and not the data models or representations used. Some exceptions exist, such as in (Iamwan, Indikawati, Kwon, & Rao, 2016), in which the authors present the Entity-Relationship diagrams for two datasets (Busan ITS Traffic Sensor Data set and Seattle Traffic Sensor Data set) and provide definitions for major concepts (e.g., Road, Road network, Traffic sensor data, etc.), with the final goal of querying and extracting timeline information from traffic sensor data. Nevertheless, there is no attempt to harmonize data into one single schema that can encompass both data sets. Also, the Entity-Relationship diagrams do not translate into the actual database representation of the data sets, nor present any metadata about the units for the measurements, coverage, accuracy, etc.

In fact, mobility- and traffic-related data harmonization, as a research topic, is somewhat a neglected topic. Searching for these exact keywords in research works' search engines, such as Semantic Scholar or Google Scholar, returns little to none results (as a side note, some of the top results are works from the author of this document, such as in the case of (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018; Figueiras, et al., 2016)). Nevertheless, there are a few works that attempt to use standards to represent and model traffic- and mobility-related data, and particularly of DATEX II, the European de facto standard for traffic-related data exchange, already introduced earlier in sub-section 2.2.3. DATEX II is being used to exchange a wide variety of mobility- and traffic-related data, such as origin/destination matrixes, which are connected with floating-vehicle data (Melo-Castillo, Canon-Lozano, Herrera-Quintero, Bureš, & Banse, 2016), vehicle parking systems (Melo-Castillo, Bureš, Herrera-Quintero, & Banse, 2017) or fully fledged ITS frameworks (Westerheim, 2014). But the focus will be on traffic flow and traffic event data management, exchange and harmonization. Some examples are described below.

In (Ruiz-Alarcon-Quintero, 2016), the author presents a data model for traffic flow sensor data, which follows the European Commission's INSPIRE (Infrastructure for Spatial Information in Europe) directive (European Commission, 2020) guidelines and is based on DATEX II. The data model is supported by an Entity-Relationship diagram for a PostGIS (PostGIS Project Streering Committee, n.d.) database and SQL scripts are used to transform heterogeneous data sources into the harmonized data model. The author presents two use cases for traffic flow data harmonization, with data collected from two cities in Spain, Sevilla and Málaga. However, the compatibility of the data model with DATEX II and the way that the harmonized data is exported to the DATEX II standard are not explained.

The authors of (Tomás, Castells, Samper, & Soriano, 2013) present an assessment on the harmonization of ITS, following the Deployment Guidelines proposed by the EasyWay project. These deployment guidelines are thoroughly explained, and the correspondent functional, organizational and technical requirements are presented in order to achieve full harmonization between different ITS core services that are combined to create added value ITS services. The authors present two of these added-value ITS services deployed in Spain, a cross-border traffic management planning tool for the Atlantic corridor between Spain and France and a web-based real-time traffic information tool called eTraffic. The authors further elaborate by analysing if these services comply with the deployment guidelines proposed by EasyWay. The main data exchange format is DATEX II. Even so, the specific usage of DATEX II is not presented and it is stated in the conclusions that a deeper analysis on the use of DATEX II to improve the information exchange is still underway.

As a final example, the authors of (Wei-Feng, Wei, & Jian, 2008) present a dynamic, real-time traffic information publication platform that uses DATEX II as the main data exchange format. Firstly, a comparison was made between DATEX II and its American counterpart, the National Transportation Communications for ITS Protocol (NTCIP) (National Electrical Manufacturers Association , 1996), from which the authors chose DATEX II as their main data exchange standard. Secondly, the framework supporting the data publication platform is presented. The framework is based on a publish-subscribe paradigm that enables users to subscribe to real-time streaming subsystems for each type of data available in the platform (e.g., traffic events, parking information, etc.). The publication of the selected data is done through a message queuing software that sends data to the subscribers using the DATEX II format. The authors conclude that using the message queuing system and the application of the DATEX II standard, the platform becomes highly scalable and the addition of new data exchange subsystems into the platform is possible with minor configurations.

2.4.2 Processing Engines

Although there are no Big Spatiotemporal data processing engines specific for the Mobility and Traffic domain, it is worth going through the existing Big Spatiotemporal Data processing engines. Since this category of processing engines does not fall in neither of the previous sections (2.2 and 2.3), they will be overviewed in this section.

There are two main types of Big Spatiotemporal Data processing engines: processing engines that already have built-in capabilities to handle Big Spatiotemporal Data or spatiotemporal extensions for processing engines. Some of the engines presented in sub-section 2.3.5, such as Apache Spark (The Apache Software Foundation, 2018) and Apache Storm (The Apache Software Foundation, 2018) have some spatial capabilities, namely through their built-in SQL querying languages (Spark SQL for Spark and Tiny Storm SQL for Storm), but they do not provide a full integration for spatiotemporal data.

A general-purpose Big Data processing engine that provides support for spatiotemporal data out-of-the-box is Apache Flink (The Apache Software Foundation, 2014). According to (Karim, Soomro, & Burney, 2018), Apache Flink provides several spatiotemporal data handling methods, such as intersection, containment and even clustering, and enables spatial partitioning with and without indexing. One example of works that use Flink as a Big spatiotemporal processing engine is proposed in (Galić, Mešković, & Osmanović, 2017). The authors present a framework for efficient real-time managing and monitoring of mobile objects through distributed spatiotemporal streams processing on large clusters. A proof-of-concept implementation is based in the Apache Flink stream processing model, which overcomes the challenges of current distributed stream processing models and enable seamless integration with batch and interactive processing like MapReduce (Dean & Ghemawat, 2008). Apache Storm can also be used for monitoring spatiotemporal streams, as proposed by the author of this thesis in (Figueiras, et al., 2018).

Regarding spatial extensions for already existing processing engines, there are several options to choose from. For Apache Hadoop, there are Spatial Hadoop (Eldawy & Mokbel, 2015) and ST-Hadoop (Alarabi, Mokbel, & Musleh, 2018). There are also GIS-specific extensions for Apache Hadoop, such as Hadoop-GIS (Aji, et al., 2013) and GIS-Hadoop (Abdul, Alkathiri, & Potdar, 2016). For Apache Spark, some examples of spatial and spatiotemporal extensions are Magellan (Sriharsha, 2017), which is officially supported by Spark, Apache Sedona (The Apache Software Foundation, 2015), formerly known as GeoSpark (Yu, Wu, & Sarwat, 2015), which is an processing engine project for spatiotemporal data based on Apache Spark, Stark (Spatiotemporal Spark) (Hagedorn & Tonndorf, 2016) and the already referred GeoMesa (The GeoMesa Project , 2013), just to name a few. For a more thorough survey on spatiotemporal extensions for Big Data processing engines and processing engines, please refer to (Karim, Soomro, & Burney, 2018).

2.4.3 Analysis, Mining and Visualization

The analysis and lifecycle management of big spatiotemporal data often depends on the domain, since the way big spatiotemporal data is handled will have consequences in the way the data is queried, analysed and visualized, and in the value of the knowledge extracted from such analysis. For instance, if the domain is related to land information or other macro-domains of Geographic Information Systems (GIS), the spatial and temporal dimensions must target big changes in both these dimensions to have a clear picture of the evolution of land use and transformation. Or if, as pointed in (Mahood, Burney, Rizwan, Shah, & Nadeem, 2017), the objective is to analyse cancer growth in human bodies, which is spatiotemporal in nature, then one also must consider the medical concepts and terminologies to formulate a suitable analysis.

Specifically, regarding MobiTrafficBD, namely traffic sensor and spatiotemporal event data, there is a panoply of analytic processes that can be applied to these data types, from clustering to anomaly detection, providing a set of answers to common questions and problematics that are generic but closely related to traffic management (Atluri, Karpatne, & Kumar, 2018). For spatiotemporal events, some examples of such common questions are:

- *How are spatiotemporal points (traffic events) clustered or organized, both in times and space?* This question refers to the autocorrelation between events, such as accidents and consequent traffic jams.
- *Are there any frequent patterns of spatiotemporal points?* This question has to do with the existence of so called "hot spots" (Levine, 2017), or zones prone to the occurrence of traffic events.
- *Is it possible to identify spatiotemporal points that do not follow the general behaviour of other points?* This question is related to the possibility of having special types of events, such as severe accidents or longer traffic jams, which do not correspond to the normal behaviour of most events.

In the case of traffic-related GRTS, some example concerns are:

- *Are there time series that present similar temporal activity and have nearby locations?* This question arises from the fact that traffic flow and speed observations that are contiguous in space and time tend to be related, and often have the same cause. Also, the combination of weather- and traffic-related time series may present similar patterns in terms of the traffic situation deterioration due to weather conditions.
- *Are there repeating time patterns for a set of time series?* This refers to seasonality, time-of-day or day-of-week influence in traffic occurrences, such as peak hours, weekday traffic or summer versus winter traffic conditions.
- *Can one find time intervals in which time series deviate from their normal behaviour, even if it is for a short period of time?* This question is linked to the detection of abnormal traffic events such as road obstructions, accidents, traffic jams, etc. in GRTS collected from road sensors, or due to abnormal weather patterns.

Another type of questions combines both GRTS and spatiotemporal events to find correlations about traffic situations. One example is:

- *Can one infer some temporal correlation between one or more spatiotemporal points and one or more time series that are spatially nearby?* This question refers to the consequences that traffic events have in the overall traffic status, such as the relation between accidents and traffic jams. It can be also related to the correlation between traffic events and weather patterns.

These, and other pertinent questions for RITMOs may be answered using several analytical processes, coming from the Data Mining, Machine Learning and Statistics areas, and by applying visual analytics techniques to detect such patterns and anomalies. First of all, it is worth mentioning that the fact that the spatial dimension in both traffic and weather sensor-based GRTS and spatiotemporal events is fixed is an advantage in their analysis, since the analysis of large volumes of spatiotemporal data without fixing any dimension is very difficult and complex (Rao, Govardhan, & Rao, 2012). Secondly, there are already some full-fledged framework and platform proposals in the literature for handling and analysing MobiTrafficBD.

The authors of (Nallaperuma, et al., 2019) present an online and incremental machine learning platform for Big Data-driven, near real-time smart traffic management, based on a three-layer architecture. The first layer is responsible for data collection and transformation, gathering and modelling data from traffic sensor networks, social media and other sources, such as CCTV and weather stations. The second layer is composed by an online, incremental and decremental machine learning model that is used to achieve unsupervised concept drift detection of recurrent and non-recurrent traffic congestions. The first two layers serve as the basis for a third layer in which several analyses are done.

First, an impact propagation analysis of the congestions in the neighbouring road network is done, based on an unsupervised data-driven approach. Second, a traffic forecasting approach based on deep neural networks is proposed to predict traffic congestions and their impact in road segments in the vicinity. Third, an intelligent traffic control approach, based on deep reinforcement learning, is used to optimize the performance of road networks in case of congestion and, finally, an emotion analysis algorithm is used when a congestion event is detected, in order to extract the emotional behaviour of commuters to improve transportation services. The downside of this proposal is that the authors focus on the algorithms but not on the technologies. The authors claim that the platform is capable of handling Big Data, but the presented scenario (the vicinity of a shopping centre in the city of Victoria, Australia) may not be considered a Big Data use case and no technology stack for the deployment of the proposed platform is presented.

On the other end of the scale, the authors of (Wu, Morandini, & Sinnott, 2015) present a cloud-based architecture supported by a technological stack for Big Data processing and visualization of traffic-related data. The framework is called SMASH and is, in essence, a distributed software stack that tackles the issues of data replicability, distributed storage and batch, offline processing capabilities and spatiotemporal indexing, querying and visualization. The technology stack is based on the Hadoop Distributed File System for raw data storage, Apache Spark for processing and analytics, Apache Accumulo (The Apache Software Foundation,

2008) and GeoMesa (The GeoMesa Project , 2013) for querying and indexing spatiotemporal data and GeoServer (Open Source Geospatial Foundation, 2001) for visualization of spatio-temporal data. Nevertheless, there is no account on the data models used, the way data is collected from heterogeneous data sources, nor the type of analyses that are realized by the framework except for the visualization of traffic flows. Further, although a benchmark on Big Data processing using Apache Spark or Hadoop is presented, it is not clear how the GeoServer software copes with effective Big Data volumes. The only comment on this subject is that every time a user performs a simple zoom/pan operation, a new request is made to the server and a new map is rendered for that request, which may not be the best approach when handling higher data volumes. Still, SMASH is the closest academic work to a fully-fledged MobiTrafficBD Framework.

Finally, another example of a Big Data-based architecture for ITS, and particularly for traffic and Mobility analysis is proposed in (Gohar, Muzammal, & Rahman, 2018). The proposed architecture has a built-in storage and analysis capability to work with ITS data and is composed of four modules, namely Big Data Acquisition and Pre-processing Unit, Big Data Processing Unit Big Data Analytics Unit and Data Visualization Unit. Both the modular approach and the individual modules' division is similar to the modular approach of the presented prescriptive methodology, as will be clear in the upcoming chapters. The architecture uses several Big Data technologies, such as Apache Hadoop for Big Data processing and MongoDB for NoSQL data storage. The authors finish with a benchmark analysis on the overall performance of the proposed architecture, in comparison with the centralized system in place in the present day. The benchmark analysis shows that the distributed architecture outperforms the traditional, centralized system in analysing ITS data.

The remainder of this section will go through the different types of analyses that are possible using Data Mining and visual analytics techniques, such as pattern discovery, clustering, prediction, classification or visualization, to name a few, in order to try to answer some of the questions above and others that might be useful for RITMOs to better understand traffic and mobility.

2.4.3.1 Pattern Discovery and Outlier Detection

Discovering both patterns and outliers in mobility- and traffic-related data helps RITMOs to better understand the spatiotemporal relationships between different traffic and mobility phenomena. On one hand, patterns may help RITMOs to better predict future occurrences of these same patterns, and, on the other hand, outliers may also point to some causality, helping RITMOs to be better prepared for the occurrence of future, similar anomalies.

In (Banaei-Kashani, Shahabi, & Pan, 2011), the authors present and test two hypotheses about traffic flows on road segments through the analysis and pattern discovery in traffic sensor data, collected from the Los Angeles County road network. The first hypothesis states that road segments may be categorized based on similar patterns present in their traffic flows and the second hypothesis postulates that road segments in each category possess not only similar traffic flows but are also similar in other types of characteristics, such as locality or connectivity. For the first hypothesis, the authors apply the X-Means algorithm, which is an extension of the K-Means clustering algorithm, to a dataset of traffic flows, comprised of volume, occupancy and speed attributes for all road segments, spanning the working hours of each day (6:00 to 21:00) and with a temporal granularity of 15 minutes between records. The result is eleven different signature patterns, corresponding to distinct categories for road segments. The authors go even beyond and characterize these eleven categories as residential, downtown, business, attraction and remote areas, and the subsequent types of road segments between these main categories (e.g., from residential to downtown, from downtown to business, etc.).

For the second hypothesis, the authors firstly present the intrinsic characteristics, or features, of different road segments: Length of the segment, direction of the segment, spatial capacity (or number of lanes), connectivity (with fan-in and fan-out, respectively, the number of nodes that end in the road segment (entries) and the number of nodes that start at the road segment (exits)) and density and locality, which are both connected to the neighbouring characteristics of the road segment. The authors developed a feature selection algorithm that uses a Bayesian network to evaluate the classification of the features that are strongly related within the same category of road segments. They found that the only feature that does not present a strong correlation with the type of road segment is spatial capacity, meaning that the number of lanes is not correlated with the type of road segment, neither with its traffic flow characteristics (the number of lanes does not affect the traffic flow behaviour in each type of road segment). The remaining features present tight correlations with the traffic flow signatures of each category of road segments. The idea is to extend this analysis to other road networks to check if these are general rules that can be applied throughout the globe and, in that case, to develop a traffic data generation tool that generates traffic flow data for those road networks for which traffic sensor data is not available.

2.4.3.2 Clustering, Classification and Prediction

Clustering can be used to infer about spatiotemporal relationships between traffic-related phenomena and to group together similar mobility-related behaviours and situations. Classification goes even further by classifying sets of patterns or clusters into groups that can be recognized as having the same characteristics. But the execution of methods for both clustering and classification over large volumes of data or fast streams is not a trivial task, since often the execution performance of these methods is directly correlated to the amount of data at hand, whether they are batches of historical data or sliding windows of data streams.

There are few research works in the literature that focus on the Big Data aspects of spatiotemporal data clustering and classification, and in most cases, these aspects are tackled not by parallelizing the training and execution of methods through the use of Big Data technologies, but by proposing new methods or extending existing ones to improve their overall performance, even if the execution remains centralized (Shao, Salim, Song, & Bouguettaya, 2016; Choi & Hong, 2021; Tang, et al., 2019). Nevertheless, some example works stand out when considering the application of Big Data and Deep Learning technologies to classify and cluster Big Spatiotemporal Data.

The authors of (Cuzzocrea, Gaber, Lattimer, & Grasso, 2016) propose a methodology for the design of a spatiotemporal clustering model, based on CRISP-DM and built on top of Weka for the analysis of spatial sectors' greenhouse gas emissions, particularly how many sectors are high emitters and how many sectors are low emitters. The clustering technique used is the K-Means method, which is one of the simpler clustering algorithms and was used in this case because the number of clusters (represented by K) was known a priori and equal to 2: low emitters and high emitters. The data was retrieved from the European Environment Agency's Web site and concerns all greenhouse gas emission data from all European countries, as per agreed in the Kyoto Protocol.

The main drawbacks of this work are the fact that, although Weka supports integration with Big Data processing engines, such as Apache Spark, there is no reference to its application in the methodology, and the selection of the algorithm, K-Means, may be a good match for this use case since the number of clusters is known, but other, more optimized algorithms for spatiotemporal clustering could be used, such as ST-DBSCAN (Birant & Kut, 2007) and its implementations for distributed environments, RT-DBSCAN (Gong, Sinnott, & Rimba, 2018) for real-time Big Spatiotemporal Data stream clustering, and MR-DBSCAN (He, Tan, & Luo, 2014) for Big Spatiotemporal Data offline batch clustering. While the former is based on the implementation of ST-DBSCAN for Apache Spark Streaming (The Apache Software Foundation, 2018) processing engine by using the SMASH (Wu, Morandini, & Sinnott, 2015) platform to leverage the execution of the algorithm using Apache Kafka (The Apache Software Foundation, 2017) as streaming source and Spark Streaming for processing, the latter is based on the MapReduce paradigm and was implemented Apache Hadoop (The Apache Software Foundation, 2018).

More recently, Deep Learning techniques have been employed for clustering Big Spatiotemporal Data. Although these techniques are widely used for prediction, learning and classification tasks, as will be discussed below, they have been repurposed for Big Spatiotemporal Data clustering with promising results. The author of (Konstantaras, 2020) proposes a distributed Deep learning-based spatiotemporal clustering algorithm that employed a Deep Learning neural network to cluster seismic events into distinct seismic zones. The neural network was developed using the CUDA C language for graphical processing units, which enables parallel execution of the neural network's training. This clustering technique detected an unknown seismic zone under the Ionian Sea.

In another work (Asadi & Regan, 2019), and focusing particularly on MobiTrafficBD, the authors present a deep embedded spatiotemporal clustering model for traffic-related GRTS. The model is based on a Deep embedded neural network with cluster weights obtained through the application of the K-Means algorithm to iteratively adjust the cluster centres. The model then extracts temporal clusters and from correlating these temporal clusters it computes the most relevant spatial clusters. The model was implemented using Keras (Chollet, 2015). The authors then validate and demonstrate the application of the model by training and executing it with traffic loop detector sensor data, and describe several interesting patterns extracted from the clusters, such as high correlation between Euclidean distance of latent features and Dynamic Time Warping distance of GRTS, distinguishable clustering probabilities for different timestamps, and dynamic spatial clustering for various hours of a day.

Although spatiotemporal-specific classification and prediction methods are rare or nonexistent, the existing generic versions can provide good results when applied to Big Spatiotemporal Data, and particularly to MobiTrafficBD. In these cases, the choice of method or model depends on the use case at hand, and on the individual performance benchmarks for each method. Namely, in the case of Deep Learning-based classification and prediction, the generic models can be modelled to cope with spatiotemporal features of MobiTrafficBD. Some examples of the application of Deep Learning techniques for MobiTrafficBD classification and prediction were already presented in previous sections (e.g., (Wang, Gu, Wu, Liu, & Xiong, 2016; Polson & Sokolov, 2017)).

As a final example that combines several of the above techniques and technologies to analyse MobiTrafficBD, the authors of (Dagaeva, Garaeva, Anikin, Makhmutova, & Minnikhanov, 2019) propose a Big spatiotemporal data mining framework for traffic- and mobility-related emergency management information systems. The framework comprises several Big Spatiotemporal Data methods, supported by Big Data technologies, that support a number of Data Mining tasks, such as i) spatiotemporal clustering to detect areas of high interest for traffic issues or emergencies, using FP-Growth and DBSCAN algorithms, ii) spatiotemporal co-location pattern mining to discover new spatiotemporal relationships between different data types, through Natural Language Processing methods, iii) spatiotemporal outliers' detection, by applying ARIMA and LSTM models and iv) spatial autocorrelation analysis and prediction to uncover causal relationships between discovered events.

The framework is based on a conjunction of several Big Data technologies, from a Apache Spark cluster that supports several Spark-based spatiotemporal extensions, such as the Spark MLLib (The Apache Software Foundation, 2018) and its implementation of FP-Growth, STARK (Hagedorn & Tonndorf, 2016) and its implementation of the ST-DBSCAN algorithm, and GeoSpark/Sedona (The Apache Software Foundation, 2015) and its spatial partitioning features, to Keras (Chollet, 2015) Deep Learning library and its implementation of LSTM. As a practical application scenario, the framework was used to extract rules on traffic incidents in the city of Kazan and demonstrated that these rules provide useful insights to local authorities and emergency services for their decision-making processes, and that the framework could be used for transportation-related incidents' management on a city-wide level.

2.4.3.3 Visualisation

MobiTrafficBD visualization is the last topic of this chapter. It is clear that the performance of visualization tools is directly correlated to the amount of spatiotemporal data objects to be visualized. Further, due to the visual composition of different data sources that produces spatiotemporal visualizations, such as in the case of the composition of world map data with regional boundaries (e.g., city limits) and with Mobility- and Traffic-related spatiotemporal data (traffic events, traffic sensors, trajectories, etc.), spatiotemporal data visualization performance is an even bigger issue (Wang, Zhong, & Wang, 2019; Wang S. , et al., 2018). Moreover, spatiotemporal visualization tools often make use of multiple linked displays to represent multiple aspects of spatiotemporal data, since map-based visualizations alone usually are not enough and need other visual displays, such as statistical graphs or timelines, to complement the complexities of the spatiotemporal phenomena (Meirelles, 2013). Thus, in recent years, several research and academic endeavours to overcome the obstacles of spatiotemporal visualization of large amounts of MobiTrafficBD have been ensued. These do not account for the already existing GIS visualization tools, some of them already overviewed in this chapter, such as GeoServer (Open Source Geospatial Foundation, 2001).

One of the more concrete examples is GeoSparkViz (Yu, Zhang, & Sarwat, 2018; Yu, Tahir, & Sarwat, 2019), which was built on top of and by the same authors of Apache Sedona (The Apache Software Foundation, 2015) (formerly GeoSpark (Yu, Wu, & Sarwat, 2015)). GeoSparkViz is a large-scale geospatial map visualization framework and extends a massively parallelized cluster computing system (Apache Spark) to provide native support for general cartographic design and seamlessly integrates with the GeoSpark spatial data management system. The main contributions of GeoSparkViz are i) the encapsulation of the main tasks of the geospatial map visualization process (e.g., spatial objects' rasterization, pixel aggregation, etc.) into a set of Apache Spark-specific massively parallelized Resilient Distributed Datasets (RDD), which are a fundamental data structure of Spark, comprised by immutable distributed collections of objects, ii) a map tile-aware data partitioning method that achieves load balancing for the map visualization workloads among all nodes in the cluster and, iii) an extensive experimental evaluation that compares and contrasts the performance of GeoSparkViz with state-of-the-art distributed map visualization systems over real large-scale spatial data.

The validation and demonstration of GeoSparkViz was performed with one set of Mobility- and traffic-related ST events: the New York city taxi trips dataset, which consists of 260 Gigabytes of data records containing pick-up and drop-off dates/times, pick-up and drop-off precise location coordinates, trip distances, itemized fares, payment method and travel distance. Pick-up and drop-off locations were represented in special spatial RDDs from GeoSpark and presented in a heat map to show the overall trends of taxi usage in New York city. More recently, GeoSparkViz was integrated with Apache Sedona and Apache Zeppelin to create a large-scale spatiotemporal data visualization system, enabling visualization of over 1 billion spatial objects (depending on the cluster size) (The Apache Software Foundation, 2021).

Other research works on this subject point to two main research paths to enable efficient visualization of MobiTrafficBD: novel visualization methods that better summarize and aggregate large-scale or extremely fast spatiotemporal datasets, without losing the meaningfulness of the underlying information and accounting for the performance aspects of such methods, or the extension of existing Big Data processing technologies to support large-scale spatiotemporal visual analytics, such as in the case of GeoSparkViz. Some examples of the former are presented in sub-section 2.2.5, such as multiple linked views (Jern & Franzen, 2006; Maciejewski, et al., 2010; Plug, Xia, & Caulfield, 2011; Harris, Brundson, & Charlton, 2013), animated visualizations (Anwar, Nagel, & Ratti, 2014; Bouattou, Laurini, & Belbachir, 2017) and space-time cubes (Kraak, 2003; Gatalsky, Andrienko, & Andrienko, 2004; Kristensson, et al., 2008; Nakaya & Yano, 2010). Two examples of applying novel visualization methods, to the particular case of MobiTrafficBD, are Traffic Origins (Anwar, Nagel, & Ratti, 2014) and TripMiner (Riveiro, Lebram, & Elmer, 2017).

Traffic Origins is presented as a simple, animated visualization technique that emphasizes the effects of traffic events on road congestion. The rationale behind this technique is that commercial mapping software typically draws attention to traffic events by placing markers at the events' locations. While this is useful for navigation, it is less useful for analysis since it does not focus the user's attention on the impact that these incidents have on traffic in the immediate vicinity before and after the incident happens. Traffic Origins was built with two main goals in mind: create an engaging visualization that used an attractive visual language to make traffic and congestion data accessible, enjoyable and easily understood by traffic management controllers, transportation expert and to be used in a walk-up-and-use setting that encourages members of the public to walk over and casually explore the data. The authors present a case study for loop detector and traffic event data from the city of Singapore, spanning one month. The case provided evidence of the applicability of Traffic Origins on a macro level, enabling observation of traffic incidents' variation over the course of a single day, week or month, and on a micro level, enabling observation of the visual relationship between traffic incidents and resulting congestion.

TripMiner is a visual analytics framework for road traffic data analysis and anomaly detection that combines linked views and other novel visualization techniques to enable analysis over large volumes of highly heterogeneous, feature-rich vehicle data. The dataset comprises trip data, contextual data (e.g., road type, maximum speed, weather, etc.) and sensor data (e.g., vehicle sensors, road sensors, etc.). The framework uses several Data Mining and Machine learning techniques to aggregate and cluster data, in order to optimize the delivery of visualizations to the users. The linked views are the Feature and Normal Model Viewer, which supports the analysis of anomalous events found by an anomaly detector module and allows the identification of the most informative or important features of a cluster, model and data set, the Temporal Viewer, which displays the selected features versus time, for one or more trips, and a 2D Interactive Map Viewer that allows zooming, panning and area selection, and complements the two viewers described above. TripMiner allows the analysis of multidimensional data, the identification of the most informative features of trips, the characterization and comparison of driving behaviours and the detection of anomalous behaviour.

Regarding the latter, there are several works that attempt to apply Big Data distributed technologies and architectures to produce visual analytics methods that can cope with large-scale or extremely fast data. The authors of (Wang S. , et al., 2018; Wang S. , et al., 2018) propose a visual analytics framework for Big Spatiotemporal Data, with each paper representing a different counterpart of the overall framework. In (Wang S. , et al., 2018), the authors propose a visual analytics framework for large-scale, batch Big Spatiotemporal Data and, in (Wang S. , et al., 2018), the same framework and workflow are redeployed for real-time streaming Big Spatiotemporal Data. The framework uses a conjunction of technologies to enable distributed parallel processing, rendering and provision of visualizations, from NoSQL databases to Big Data processing engines (Apache Spark, Apache Hadoop). Although these works represent a suitable example for the upcoming prescriptive methodology for developing MobiTrafficBD frameworks, in this case focusing on visualization, they present a major drawback: They are based on a commercial, non-open-source software product for GIS services provision, called SuperMap (SuperMap Software Co., Ltd., 1997).

The authors of the SMASH platform (Wu, Morandini, & Sinnott, 2015), already presented in this section, and described as a cloud-based platform architecture for Big Data processing and visualization of traffic data, perform a benchmarking on the performance of the overall platform, with an emphasis on data aggregation for visualization and in relation to the amount of data to be aggregated versus the number of nodes in the Spark cluster in which SMASH is deployed. The visualization tool integrated in SMASH is GeoServer (Open Source Geospatial Foundation, 2001). Last but not least, the authors of (Root & Mostak, 2016) present MapD (Massive Parallel Database), an in-memory database and Big Data analytics platform designed to be deployed on GPU-based environments that can query and visualize Big Spatiotemporal Data up to one hundred times faster than other Big Data platforms. MapD achieves its speed using a variety of novel techniques, such as data rendering and visualization in situ on the GPU without the need to copy query result set before rendering it, code vectorization that allows the compute resources of a processor to process multiple data items simultaneously and highly optimized GPU routines for common database operations.

To conclude, there are also works that follow simultaneously both research paths: the use of novel visualization techniques coupled with Big Data distributed technologies and architectures to render the final visualizations. The authors of (Perrot, Bourqui, Hanusse, Lalanne, & Auber, 2015) propose a visualization system for large interactive visualization of density functions for MobiTrafficBD. The framework is supported by Big Data distributed technologies, such as Apache Spark for data aggregation and Apache HBase for data storage, as well as GPU-based techniques to render novel density cluster visualizations in the form of geographical heatmaps. To benchmark the framework, the authors used four extremely large datasets, two of which corresponded to bike users' positions, one corresponded to the whole collection of points of interest from OpenStreetMap and another with GPS traces registered in the OpenStreetMap database. The benchmark analysis covered clustering, rendering and final image quality and concluded that the framework is able to interactively explore sets of points of any size, with the only limiting factor being the size of the Big Data infrastructure.

Finally, the authors of (Guan, et al., 2020) present MAP-Vis, a distributed Big Spatiotemporal Data visualization framework based on a novel visualization technique denominated Multi-Dimensional Aggregation Pyramid (MAP) model. The MAP visualization model is based on the Space-Time Cube (Gatalsky, Andrienko, & Andrienko, 2004; Bach, Dragicevic, Archambault, Hurter, & Carpendale, 2014; Kraak, 2003; Kristensson, et al., 2008), extended with the attribute dimension to Space-Time-Attribute Cube and providing the building blocks for the proposed MAP model, and on the 2D Tile Pyramid, which implements the idea of 2D spatial aggregation and provides a good implication for the simultaneous aggregation of space, time, and attributes. The MAP model enables the hierarchical aggregation to be achieved not only on the spatial dimension but also on the temporal and attribute dimensions. To validate and demonstrate the efficiency and usefulness of the proposed model, the authors developed a Big Data framework that uses Apache Spark, as the main processing engine to create the visualization model, and Apache HBase, as the distributed storage technology that houses the pyramid model. Further, the framework is validated by applying several different Big Spatiotemporal Data sets to build visualizations based on the MAP model, such as for instance, the 60 Gigabyte-New York city taxi data set. The MAP-Vis realizes millisecond-level multidimensional data querying and achieves good interactive visualization. Experimental results validate the efficiency of both the MAP model and the MAP-Vis framework, both of which can provide high scalability for processing capability and online visualization.

3

A Methodological Approach for MobiTrafficBD Frameworks

This chapter describes a data-driven and prescriptive methodological approach for the design and development of Mobility- and Traffic-related Big Spatiotemporal Data Frameworks (MobiTrafficBD Frameworks). The intended characteristics are further explored, giving rise to a set of functional and non-functional requirements and design considerations for the implementation of generic, data-driven frameworks that process, manage and analyse Big Spatiotemporal Data.

Based on such requirements and considerations, a generic model or conceptual architecture is presented and complemented with logical components, data flows and technological stacks that can fulfil the requirements. Finally, a set of general guidelines and best practices for the design of MobiTrafficBD frameworks is presented as a wrap-up conclusion for the chapter.

3.1 Characteristics, Requirements and Design Options

Chapter 1 overviewed some generic characteristics, or requirements, for MobiTrafficBD frameworks, regarding the Big Data and spatiotemporal nature of such frameworks in the context of Intelligent Transportation Systems. These can be seen as the necessary characteristics of generic ITS systems and frameworks with the objective of handling and analysing MobiTrafficBD, and are described as follows:

- Efficient collection and storage of MobiTrafficBD.
- Application of standards and interoperability tools to tackle the heterogeneity of MobiTrafficBD.
- Awareness towards the spatiotemporal nature of MobiTrafficBD.

- RITMOs' decision support through the application of efficient Big Data, data mining and visual analytics tools over MobiTrafficBD.
- Value extraction from MobiTrafficBD to support RITMOs' monitoring and decision-making processes.

These characteristics have a clear objective: To bring Big Data and spatiotemporal data together to support RITMOs with a clear perspective on mobility and traffic and to optimize their decision-making processes. Summarizing the general characteristics described above, a generic MobiTrafficBD framework should be able to collect high volumes of heterogeneous MobiTrafficBD swiftly, harmonize them into standard formats, store them in databases, apply suitable processing and analysis methods and present the results to RITMOs in a meaningful and valuable fashion through visual analytics methods.

Regarding collection and storage, the main requirements are linked to the online versus offline characteristics of the data at hand (i.e., if the data is a real-time stream or an offline batch of historical data), different functional requirements are imposed. The difference between stream and batch data will be further explored in the next chapters. One important requirement comprises the need for MobiTrafficBD frameworks to collect data from different data sources and data sharing mechanisms, such as Web services, databases or file systems.

Next, regarding data heterogeneity, requirements comprise the need for harmonize Big Spatiotemporal Data according to the data type and using existing data standards. Depending on the data type (e.g., traffic sensor data, traffic event data, weather station data), Big Spatiotemporal Data coming from different data sources and bearing different data formats should be harmonized to a single standardized format.

This means that, for instance, traffic sensor data from two different sources, and having different formats and even different attributes, should be harmonized to a single common schema that enables the different attributes of different data providers and sources to be stored without any loss of the original data. Another requirement is linked to the data collection mechanisms in place, since MobiTrafficBD frameworks should be able to access and collect data from data sources through different mechanisms, such as, for instance, Web services, file systems or database systems. Furthermore, MobiTrafficBD frameworks should enable the addition of new standard-based data formats for data types that are not initially considered within the frameworks. For instance, if a framework cannot handle public transportation data, then it should be possible to add new common or standard data formats for this new type of Big Spatiotemporal Data.

The third and fourth points account for both the Big and spatiotemporal natures of MobiTrafficBD: On one hand, MobiTrafficBD frameworks need to employ Big Data technologies that can efficiently process and analyse high-volume and high-speed MobiTrafficBD, in such a way that the resulting insights are useful enough and are gathered swiftly enough to be effectively used and capitalized upon within RITMOs' decision-making processes and mobility- and traffic-related optimization tasks. On the other hand, the value of these insights is directly linked to the spatiotemporal nature and relationships of the data at hand, hence processing and analysis actions must be directed towards the spatiotemporality of MobiTrafficBD. Hence, the tools and methods used to process and analyse MobiTrafficBD must not only be compliant with the Big Data technological paradigm but must also be suitable for spatiotemporal data.

Only when all the above characteristics are fulfilled, is the fifth and last point possible: if heterogeneous MobiTrafficBD, gathered from completely different data sources, is harmonized into common formats, and processed and analysed taking into account both the Big Data and spatiotemporal characteristics of MobiTrafficBD, then RITMOs will be able to extract insights with the necessary value to really be useful within decision-making processes. These characteristics give rise to the elicitation of MobiTrafficBD framework requirements.

Requirements may be divided into functional and non-functional requirements. Table 3.1 and Table 3.2 present respectively the generic functional and non-functional requirements for MobiTrafficBD frameworks.

As formally described in (Chung, Nixon, Yu, & Mylopoulos, 2012), a functional requirement is a system requirement that specifies a function that the system or one of its components must be capable of performing. Functional requirements define system behaviours, i.e., the fundamental processes or transformations that system's components (software plus hardware) perform on inputs to produce outputs. Conversely, a non-functional requirement is a system requirement that describes not what the system will do, but how the system will do it. This includes a system's performance requirements, external interface requirements, design constraints and quality attributes. Non-functional requirements are often evaluated subjectively because their veracity is hard to validate objectively.

These requirements are data-driven in the sense that all of them focus on data and on the generic actions carried out on such data by MobiTrafficBD frameworks, to produce meaningful and valuable insights for RITMOs. There are other functional requirements that must be considered when designing MobiTrafficBD frameworks, such as the case of registering and logging users in and out of the framework, so that data access can be secured and private.

Table 3.1 — MobiTrafficBD Frameworks functional requirements

Functional Requirement ID	Requirement Description
FR1	MobiTrafficBD frameworks should be able to collect Big Spatiotemporal Data, whether it is comprised by high-volume historical batch data or high-speed real-time streams
FR2	MobiTrafficBD frameworks should be able to collect Big Spatiotemporal Data from different data sources and through different mechanisms
FR3	MobiTrafficBD frameworks should store big volumes of Big Spatiotemporal Data and high-speed streams of real-time Big Spatiotemporal Data
FR4	MobiTrafficBD frameworks should harmonize data from similar categories (e.g., traffic sensor data) into standardized formats
FR5	MobiTrafficBD frameworks should enable the addition of new standardized formats for new data categories, not yet present in the framework (e.g., public transportation data)
FR6	MobiTrafficBD frameworks should enable Big Data processing and analysis of both high-volume (batch) and high-speed (streaming) MobiTrafficBD
FR7	MobiTrafficBD frameworks should also enable the analysis of MobiTrafficBD based on their spatiotemporal attributes
FR8	MobiTrafficBD frameworks should provide visual analytics tools for visualizing processing and analysis results and insights

The non-functional requirements presented in Table 3.2 correspond to intrinsic characteristics a framework must strive for, to achieve the fulfillment of the functional requirements presented in Table 3.1. As stated in (Sachdeva & Chung, 2017), non-functional requirements are vital to projects involving cloud and Big Data and need to be handled in a suitable manner early in the software lifecycle. Table 3.2 presents a description of each non-functional requirement and its link to the functional requirements presented in Table 3.1, when appropriate.

Most of these requirements are transversal subjects in several discussions and in the literature regarding Big Data. Big Data technologies are now in the spotlight for science, business and industry, because of the inherent promise for the creation of a paradigm shift for automation of all processes within these domains. In fact, Big Data is becoming the envisaged conceptual and technological ecosystem for solving different aspects of human activity in areas such as industry, society, healthcare, science and mobility, among others.

Table 3.2 — MobiTrafficBD Frameworks non-functional requirements

Non-functional Requirement ID	Non-functional Requirement	Requirement Description
NFR1	**Interoperability**	Addresses the ability of systems and frameworks that create, exchange and consume data to have clear, shared expectations for the contents, context and meaning of that data. Related to **FR2**, **FR4** and **FR5**.
NFR2	**Elasticity**	Refers to the ability of systems and frameworks to adapt to new data types, sources and formats, as well as new workload variations, opening the application boundaries to new use cases. Related to **FR1**, **FR2** and **FR5**.
NFR3	**Scalability**	Depending on the volume and speed of data at hand, systems and frameworks must be able to dynamically adapt their infrastructure to accommodate higher data volumes and higher data speeds. Related to **FR1**, **FR3** and **FR6**.
NFR4	**Robustness**	Concerns the ability of frameworks to recover from errors or issues during runtime. The robustness of the framework must be present across all stages and processes. Related to **FR2** and **FR6**.
NFR5	**Distributed & Flexible Storage**	Linked to the scalability and elasticity, refers to the necessary resilience and flexibility that storage mechanisms must possess to ensure data storage resilience, fault tolerance and availability. Related to **FR2** and **FR3**.
NFR6	**Parallel & Distributed Processing**	Also linked to scalability, addresses the performance, fault tolerance and resilience of data processing mechanisms by means of distributing and parallelizing workloads through several hardware nodes. Related to **FR1**, **FR3** and **FR6**.
NFR7	**Spatiotemporality**	Refers to the focus given to spatiotemporal characteristics in data, when applying processing, analysis and visual analytics mechanisms. Related to **FR7** and **FR8**.
NFR8	**High-Performance**	Addresses the necessary performance to deliver valuable results and insights in a timely manner to support decision-making processes, in both high-volume (batch data) and high-speed (streaming data) cases. Related to **FR1** and **FR6**.

NFR9	**Security & Privacy**	Refers to the necessary security and privacy mechanisms to protect personal and community data from data leaks, breaches and hacks.
NFR10	**Minimum Maintenance**	Accounts for the need for easy configuration, easy deployment and easy maintenance of the solution. Related to the overall framework
NFR11	**User Friendliness**	Points towards the easy-ot-use characteristics of the framework. Since most RITMOs do not have a IT background, MobiTrafficBD Frameworks should be intuitive and easy to use by other than researchers and practitioners. Directly related to **FR8** in terms of user-friendly data visualization.

As already pointed out in Section 2.3, an important step in the design and development of MobiTrafficBD frameworks is the choice of a suitable reference architecture, in the case of this thesis work, the BDVA-RM (Figure 2.5), and the addition, if necessary, of the spatiotemporal components to the reference architecture. Nevertheless, the proposed prescriptive methodology will only focus on some of the horizontal and vertical layers of the BDVA-RM, as represented in Figure 3.1. In the data domain, the presented methodology will mainly concern structured data, time series and IoT data, geographic, spatiotemporal data and map and geography data in the form of graph data.

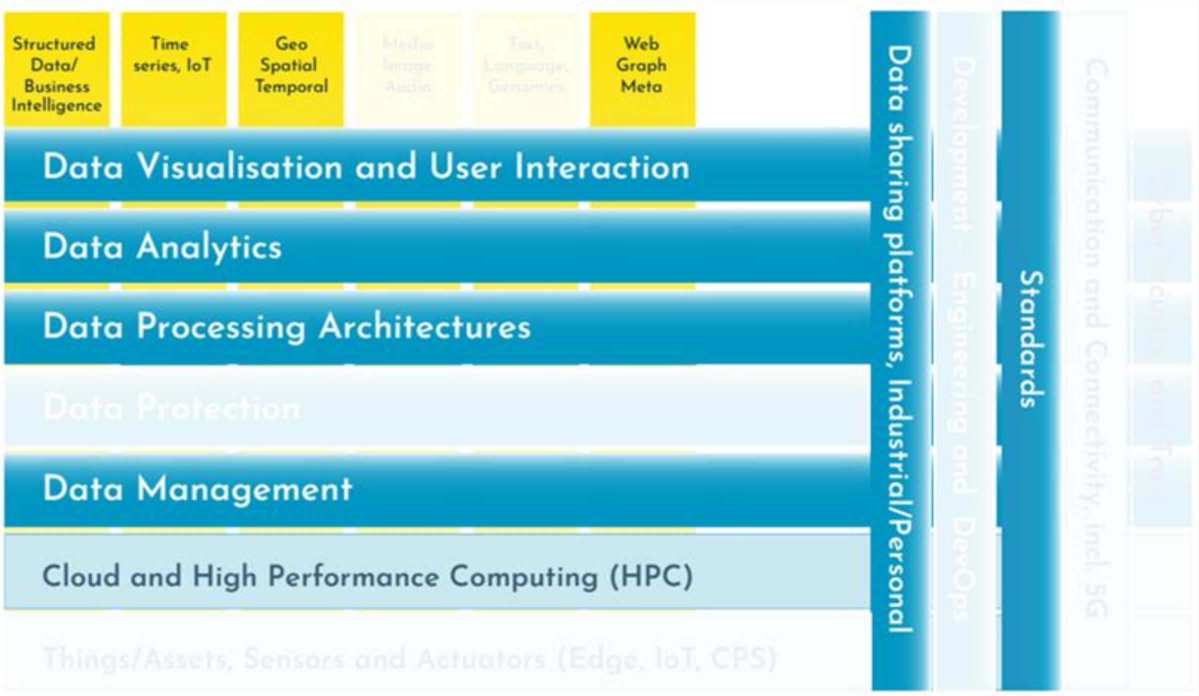

Structured Data/ Business Intelligence
Time series, IoT
Geo Spatial Temporal
Media Image Audio
Text, Language, Genomics
Web Graph Meta
Data Visualisation and User Interaction
Data Analytics
Data Processing Architectures
Data Protection
Data Management
Cloud and High Performance Computing (HPC)
Things/Assets, Sensors and Actuators (Edge, IoT, CPS)
Data sharing platforms, Industrial/Personal
Development - Engineering and DevOps
Standards
Communication and Connectivity, incl. 5G

Figure 3.1 — Positioning of the proposed methodology in relation to the BDVA-RM (adapted from (Big Data Value Association, 2020))

The necessity of addition of spatiotemporality to the chosen reference architecture has to do with the inherent generic nature of most reference architectures. Since, to the best of the author's knowledge, there is no reference architecture or model that satisfies both the Big Data and spatiotemporal data needs of MobiTrafficBD frameworks, the choice of a Big Data reference architecture and the posterior addition of spatiotemporal components seems to be the best way to tackle the design of these frameworks.

Spatiotemporal aspects must be present in almost all the layers of the BDVA-RM that will be tackled by the proposed methodology (Figure 3.1). MobiTrafficBD is collected from the Things/Assets, Sensors and Actuators layer, although the inner workings of this layer (e.g., sensor technology, IoT concepts, etc.) will not be the subject of this thesis work. The Data Management layer is responsible for the actual data collection process, as well as for cleaning, harmonization and storage processes. Although the data cleaning process is an important step towards data quality, it will not be thoroughly approached in this document, but will be discussed whenever needed. For now, it suffices to say that the spatiotemporal data cleaning process is not necessarily done upon the data's spatial and temporal attributes, but mainly performed due to unreliable readings on actual measured attributes (Zhou, Li, & Gu, 2020), such as those obtained by sensors, for instance.

Nevertheless, both harmonization and storage processes need to be prepared to handle both the spatiotemporal and high-volume, high-speed characteristics of MobiTrafficBD. In the case of harmonization, the chosen data modelling standards must comprise spatial and temporal attributes in their data models, and there is also the issue of harmonizing the spatial and temporal attributes themselves, whether to a uniform geographic reference system, for spatial attributes, or to a uniform temporal representation, as in the case of timestamps. For storage, there is also the need for the database system, or any other storage mechanism, to be able to cope with both Big Data and spatiotemporal data.

Next, the Data Processing Architectures layer is responsible not only for the actual data processing tasks, but also for the overall architecture that enables these tasks. This point is especially important due to the growing need of developing frameworks that enable integrated processing of data-at-rest (high-volume, batch data) and data-in-motion (high-speed, streaming data). The problem of achieving effective and efficient processing of data streams (data-in-motion) and the integration with already existing batch data in a Big Data context is far from being solved. Focusing on MobiTrafficBD, this layer must be able to cope with both spatiotemporal data streams and data batches, by leveraging the necessary infrastructural, orchestration and performance optimizations to integrate, fuse and aggregate both data batches and streams in meaningful and valuable ways, in order to support Big Spatiotemporal Data analytics tasks, such as prediction or pattern discovery, to name a few. In this sense, the focus must be to employ already existing Big spatiotemporal data processing frameworks, such as Apache Flink (The Apache Software Foundation, 2014), which has already spatiotemporal processing and analysis features off-the-box (Karim, Soomro, & Burney, 2018), or Spatial Spark (Karim, Soomro, & Burney, 2018), a spatial extension for Apache Spark (The Apache Software Foundation, 2018).

The Data Analytics layer is responsible for data analytics and advanced processing activities, namely data mining, knowledge discovery, machine learning and deep learning tasks, with the aim of providing insights into the data. Specifically regarding MobiTrafficBD, the Data Analytics layer, along with the Data Management and Data Visualisation layers, is where the emphasis on the spatiotemporality of data is crucial. There is a need to bring Big Data analytics and spatiotemporal data analysis closer together, by implementing already proven spatiotemporal data mining and analysis methods within Big Data analytics tools, such as Apache Spark MLLib (The Apache Software Foundation, 2018), or distributed algorithm development frameworks, such as, for instance, Apache Mahout (The Apache Software Foundation, 2014).

Finally, the Data Visualization and User Interaction layer is responsible for delivering explorable and understandable insights by interacting with users through visual analytics methods. Regarding big spatiotemporal data, the main challenges are to develop innovative ways to visualise data in the geospatial domain, such as geo-locations, distances and space/time correlations (i.e. sensor data, event data) and to tackle the issue of multiple scale/granularity spatiotemporal data, facilitating the empirical search for acceptable scales of analysis and the verification of results by modifying the scale and the means of any aggregation (Big Data Value Association, 2020).

From the vertical concerns presented in the adapted BDVA-RM (Figure 3.1), the only one which needs the addition of spatiotemporality is the concern for the use of standards, since Data Sharing Platforms are not dependent of any spatiotemporal characteristics of data. As expressed by the BDVA, "the ‘variety’ of Big Data makes it very difficult to standardise. Nevertheless, there is a great deal of potential for data standardisation in the areas of data exchange and data interoperability" (Big Data Value Association, 2020). Specifically, for MobiTrafficBD, the use of data exchange and modelling standards is important due to the heterogeneity of spatiotemporal data, not only in terms of their sources but also in terms of data collection methods and technologies, formats and schemas, scales and granularities and spatial reference systems, just to name a few. Furthermore, different standards for different data types, such as traffic sensor data and public transportation data, may be combined in order to enrich the portfolio of possible data sources and types that can be used within MobiTrafficBD frameworks.

As a final side note, the Data Protection layer of the BDVA-RM, although not an active part of this work as previously stated, is responsible for data privacy, anonymization and security processes. Hence, there is an increasing necessity for the employment of privacy-protection and secure data exchange mechanisms that offer guarantees of truly secure, formal data privacy.

3.2 Logical Components and Data Flows

The logical components included in the proposed approach are defined according to the components present in the BDVA-RM, since it aims to be compliant with current standards and trends in the Big Data community, namely in Europe. Noticeably, the proposed model of logical components and data flows presents some significant modifications, omits some of the layers that are not subjects of this thesis work, such as the case for Data Protection, and extends the BDVA-RM with new components.

The proposed approach also considers relevant principles and guidelines provided by previous published works, such as the main Big Data Reference Architectures (e.g., NIST (NBD-PWG, 2015), BDVA (Big Data Value Association, 2020)), the Big Data Processing Flow proposed by (Krishnan, 2013), the survey named Spatiotemporal Aspects of Big Data, proposed in (Karim, Soomro, & Burney, 2018), the guidelines for quality Advanced Traveller Information Systems data (America's Advanced Traveller Information Systems Committee, 2000) or the Big Data Warehousing guidelines from (Costa C. F., 2019), just to name a few. Furthermore, the proposed logical components and data flows model also encourages conformity with three of the main guidelines proposed within the Lambda Architecture (Marz & Warren, 2015): first, data should be stored at the highest level of detail possible (i.e. raw data) since it may serve future analytical purposes not previously planned and minimizes the threat of losing data in the processing and analytics processes; second, whenever possible, data structures should be modelled and used to store a set of immutable events, avoiding updates to existing data; finally, data at different speeds certainly has different requirements and, therefore, different logical components for batch and streaming data must be taken into consideration. Furthermore, this work strives to house all the desired properties of a Big Data system, as proposed in the Lambda Architecture, and already highlighted as non-functional requirements: Robustness and fault tolerance, low latency, scalability, extensibility, ad hoc querying and minimal maintenance.

The logical components and data flows model is presented in Figure 3.2 and is a generic, conceptual model of how a MobiTrafficBD framework should be designed and built. The model is divided into three distinct main components: Data Providers, such as for instance, physical sensors, law enforcement authorities and road infrastructure operators' databases or traffic and mobility data aggregator and provision companies, the MobiTrafficBD framework itself and End Users, which comprise RITMOs or any other stakeholders that have access to the framework or that use the guidelines to tackle new use cases related to Mobility and traffic, as for instance public transportation operators or traffic law enforcement authorities. The arrows represent the flow of data between the framework and the Data Providers and End Users components. Each colour represents the type of data that flows through the architecture: blue represents batch, historical data, green represents real-time data streams and red represents the interactive data accessed through queries to databases containing both streaming and batch data. Furthermore, two types of arrows are presented: full lines are mandatory flows of data, while dashed lines are optional paths for the data to flow within and across the framework's layers. The distinction between data flows is an important characterization since data gathered at different speeds has different requirements, and should be handled differently, even if the overall data handling process is the same (e.g., different data storage mechanisms and technologies should be used when storing batch or streaming data).

3.2.1 Data Providers

The bottom component of Figure 3.2 marks the beginning of the flow of data through a MobiTrafficBD framework. Data providers are physical entities that capture, aggregate and introduce new data into a MobiTrafficBD framework and can take the form of people, companies, sensors, computer systems or Web sources, just to refer a few possibilities.

Hence, the Data Providers component represents the set of available data sources, whether internal or external to the framework, online or offline, and with automatic or manual data capture or aggregation. The data may be represented through GRTS (e.g., sensor readings), ST events (e.g., traffic events captured by experts or by automated event processing systems), graph data (e.g., cartography and map data) or other data types, such as in the case of Web services for location awareness and enrichment (e.g., Foursquare [291]).

Although some of these data types may be more often characterized as data streams or batches, it is considered that all data types can be represented by both batches and streams of data. Some of the responsibilities of a data provider are to enable data access through suitable interfaces, to provide adequate metadata, to enforce access rights and to assure data privacy and security throughout the data capturing and transmission processes.

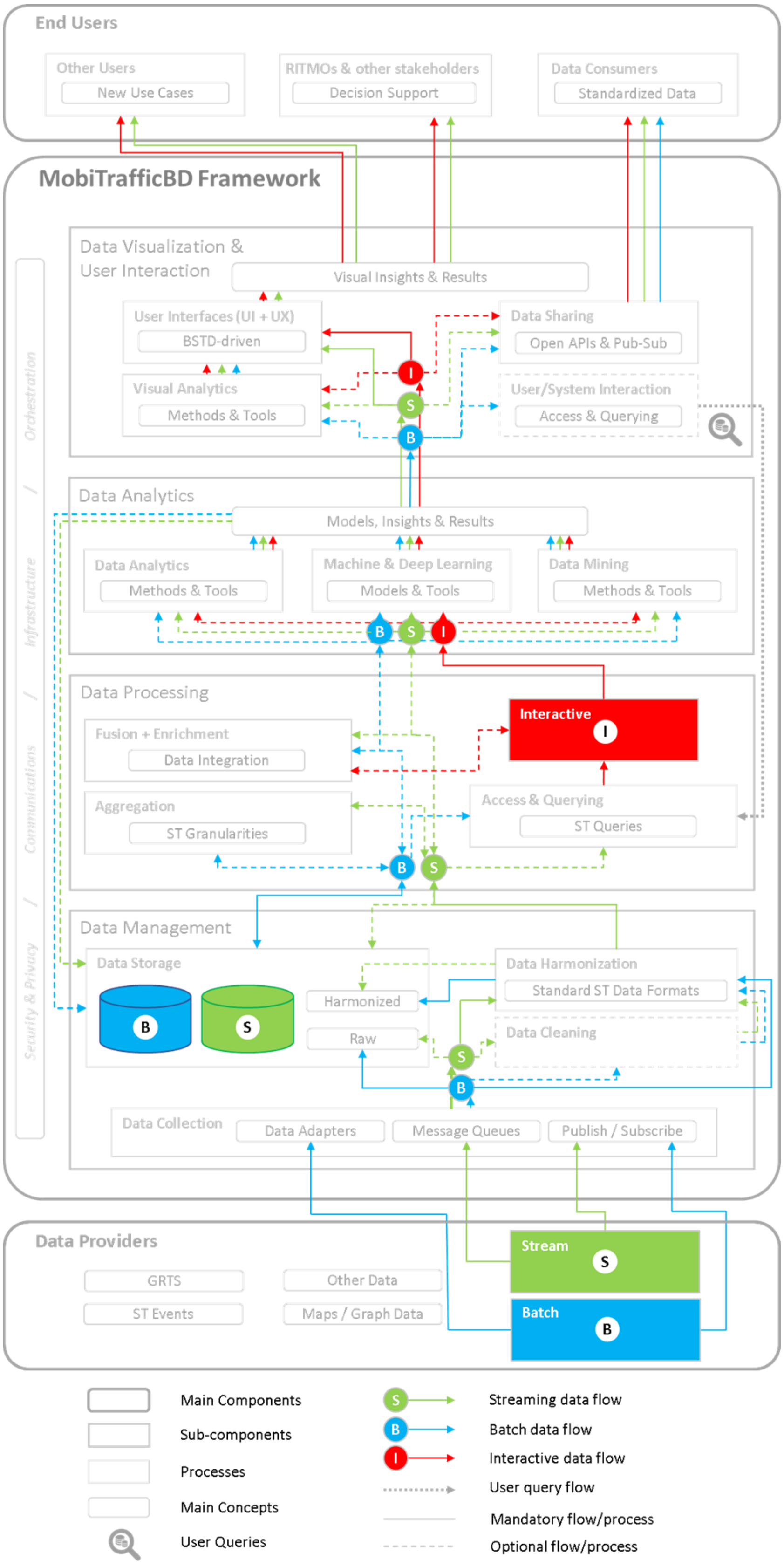


Figure 3.2 — Logical Components and Data Flows Model

3.2.2 MobiTrafficBD Framework

A generic MobiTrafficBD framework is comprised of four horizontal sub-components that mimic the horizontal layers of the BDVA-RM, with the exception of the Data Protection layer, which is represented in the logical components and data flows model as a vertical concern, in the form of Security and Privacy, because it is crucial to guarantee data privacy and security throughout the flow of data within the framework. The remaining vertical concerns are Communications, Infrastructure and Orchestration. These vertical concerns must be considered in every horizontal layer of the framework since they provide the direct support for Big Data collection, storage, processing and analysis processes.

Communications refers to the communication mechanisms used to enable communication between different layers in the framework and between physical and virtual infrastructural resources used by the framework. These communication mechanisms must be able to cope with the high-volume, high-speed characteristics of Big Data, while presenting fault tolerance and resilience against data loss during data transmission between the different entities, components and layers of the framework. Infrastructure represents the hardware and software infrastructure that grants the processing capabilities needed for MobiTrafficBD framework to perform Big Data management, processing and analysis procedures, and may be categorized as on-premises, when the servers are proprietary and are deployed on the premises of the company responsible for the MobiTrafficBD framework, or cloud-based, if the framework is deployed on a cloud environment.

Finally, Orchestration symbolizes the "glue" that brings all the vertical concerns and horizontal layers together. It is responsible for managing the framework in terms of workload distribution across physical and virtual infrastructural resources, easy configuration and deployment, and development operations (DevOps) in general. It is directly linked with several non-functional requirements, such as Scalability, Extensibility, Minimal Maintenance, and Low Latency since it controls and optimizes all the processes within the layers of the framework.

Looking at the horizontal sub-components, the main data-driven processes are represented by squared boxes and their internal concepts and characteristics are represented by rounded boxes. The suggested horizontal sub-components comprise all phases of the Big Data lifecycle, already introduced in the previous chapter, and shown in Figure 2.4 (Big Data Lifecycle). Each sub-component will be described in the following sections. Data flows through the different sub-components and their processes, on both mandatory (full) and optional (dashed) paths.

The proposed data-driven logical components and data flows model follows two complementary data-handling approaches: bottom-up and top-down. A bottom-up approach is primarily focused on streaming (green arrows) and batch (blue arrows) data. In the bottom-up approach, data is captured through data collection processes and flows upstream, undergoing processing and analytics procedures that are defined and configured beforehand, and results are normally composed by detected anomalies and patterns, through the application of anomaly and pattern detection, clustering or classification processes, or by predicted outcomes, by utilizing pre-trained prediction models. This entails that end users do not have the option to directly choose the raw data they want to analyse and visualize. On the other hand, in a top-down approach, the data to be processed and analysed is selected by end users, through and data querying and access mechanisms. This approach is reserved for interactive data, since end users must select the spatiotemporal scope and range of the data to be analysed, as well as the attributes of interest for the analysis.

3.2.2.1 Data Management

The Data Management sub-component (Figure 3.3) is responsible for all the data processes from data collection to data storage, formally known as Extract-Transform-Load (ETL) or Extract-Load-Transform (ELT) processes (Bala, Boussaid, & Alimazighi, 2016).The choice between ETL or ELT resides in the strategy undergone for storage of raw data. For instance, in the case of streaming data, an ETL approach may be used, entailing that data is extracted, transformed and stored via background processes. Nevertheless, a hybrid approach is recommended, meaning that both ETL and ELT should be aggregated so that data is stored before (raw data) and after (harmonized, cleaned data) the transformation step. The logical components and data flows model of Figure 3.2 only presents harmonization and cleaning as intermediary processes, but in fact there can be other processes in between data collection and storage (e.g., data fusion between different data sources). Data is collected from data providers and sources through different mechanisms.

In the case of big volumes of batch data, the most common data collection mechanism is the use of data adapters, i.e., adapters specifically developed for the type of data source at hand. There is a wide range of possible data adapters, such as database adapters, which are specifically designed to collect data from relational and other databases, frequently using ODBC (Open Database Connectivity) (Signore, Stegman, & Creamer, 1995) connectors, Web service adapters, which are often based on the HTTP (Hypertext Transfer Protocol) (Fielding, et al., 1985) or FTP (File Transfer Protocol) (Postel & Reynolds, 1985) protocols, or file adapters, which read specific file types, such as CSV (Comma Separated Values) (Shafranovich, 2005), JSON (JavaScript Object Notation) (ECMA Technical Committee 39, 2017) or XML (eXtended Markup Language) (Murata, St. Laurent, & Kohn, 2001), just to name a few. The main requirement for these adapters is the ability to access and collect big volumes of batch data in a fast and effective way. Examples for batch data collection will be presented in Chapters 5 and 6, namely the collection of large volumes of GRTS in the form of traffic sensor data (sub-sections 5.2.2.1 and 6.1.1) and public transportation-related ticketing transactions (sub-section 6.1.1).

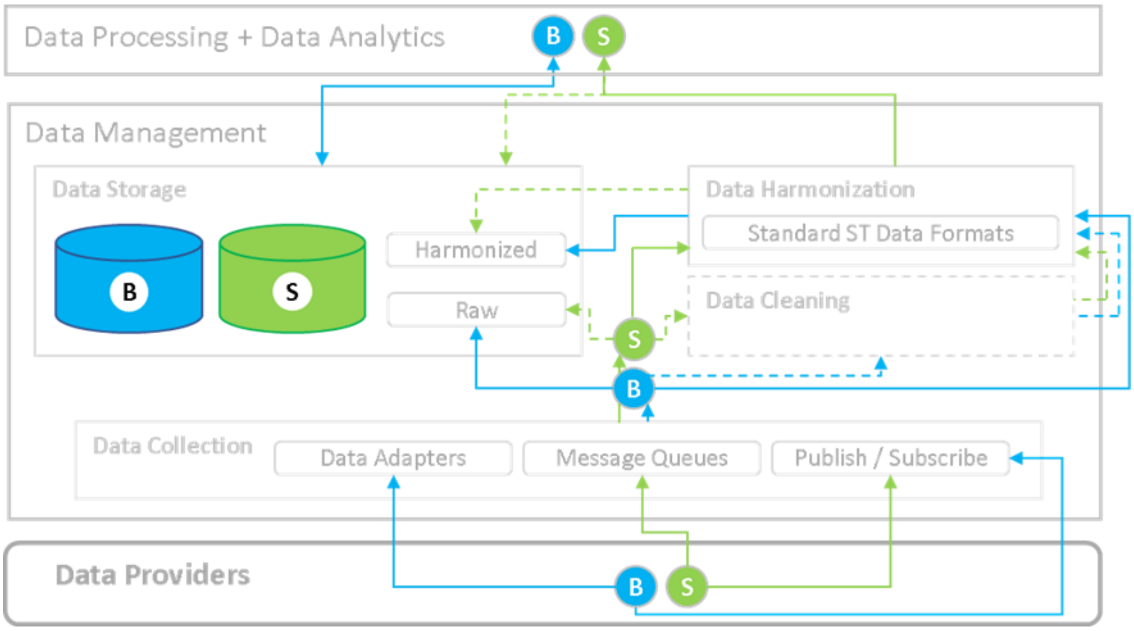


Figure 3.3 — The Data Management Sub-component

For streaming data, the most common data collection mechanism is based on the message queuing paradigm. Message queuing is a form of asynchronous communication between systems or services in which incoming messages are stored in a queue until they are extracted and processed. The basic architecture of a message queue is simple; there are client applications called producers that create messages and deliver them to the message queue. Another application, called a consumer, connects to the queue and gets the messages to be processed. Message queues may fall into the publish/subscribe mechanisms category, although publish/subscribe mechanisms can also be used to transmit batch data. The publish/subscribe paradigm is based on the subscription of a data source by a system, and the posterior broadcast of messages to all systems that subscribed that data source. In both message queueing and publish/subscribe paradigms, data may be transmitted in several data formats, such as the ones already presented above, JSON, XML, CSV, etc. Examples for streaming data collection will be presented in Chapters 5 and 6, namely the collection of GRTS streams in the form of real-time traffic sensor data (sub-sections 5.2.2.1 and 6.1.2) and social media-based ST Events (sub-section 6.1.2).

After its collection, it is advisable to store raw data before undergoing further data management processes. In the case of batch data, it is mandatory to store raw data to prevent data loss errors that may occur when cleaning and harmonizing data, or in other processes further down the framework's pipeline. Even in the case of streaming data, it is strongly encouraged for raw data to be stored, directly upon arrival of each message, or through a background job that accesses a buffer of individual records to store them in their raw format. A reason for storing streaming data is to store it as an historical data set that can be used for creating and optimizing streaming analytics models, such as predictive models or anomaly detection rules, for instance.

In the case of GRTS, in which the spatial dimension is static, spatial and time-static attributes (attributes that do not change over time) to be stored in separately of time-variable attributes (attributes that change in time). One example is traffic sensor data, in which the location of the sensor and time-static attributes, such as road name, road direction, or other metadata can be stored in one database table and individual sensor readings, with different time-variable attributes along with the reading's timestamp in another table. Practical examples on how to select a suitable storage mechanism or technology will be presented in sub-section 5.2.3.

Following the process of storing it, raw data may need to be cleaned, although it is not a mandatory step (marked as a dashed box in Figure 3.3); it was already stated in this chapter that the presented work will not delve into the specifics of data cleaning. It suffices to refer that data cleaning is an important procedure, especially in the case of spatiotemporal data, like ST Events and GRTS, although it should be realized in the data source and not after, meaning that data providers should enable data cleaning processes associated to their data gathering and provision processes. One new trend regarding the provision of data cleaning procedures at the time of data gathering is referred to as data cleaning on the "edge", i.e., directly on the data gathering hardware, such as sensor platforms or data gateways (Wang, et al., 2019). An example on the use of a standard Data Mining methodology for MobiTrafficBD cleaning will be the subject of sub-section 5.1.1.2.

After storage and optional cleaning, raw data must be harmonized. Data harmonization is a critical step because it enables full data interoperability while providing the necessary compliance with proven data standards and interoperable models. The harmonization process is a procedure in which the attributes present in the raw data format are mapped to the attributes in the harmonized data model. This mapping procedure is not only based on aligning the attributes' names and types (e.g., String, Number, Date, etc.) but also to transform their characteristics into the harmonized model's. For instance, considering an attribute representing date and time in a specific format, such as "2020-10-26 14:57:00", when harmonizing such attribute, the date and time format must be provided to the harmonization procedure, so that it can recognize the format and transform it to the harmonized one. Another example is the harmonization of location points, in the form of coordinates. There are many coordinate formats and geographic reference systems, and location attributes must be harmonized to be compliant with the chosen harmonized format and reference system. As the subject of MobiTrafficBD harmonization has been one of this thesis work's main contributors, several example use cases for data harmonization will be explored throughout Chapter 5.

After harmonization, it is again recommended that the resulting data is stored. For batch data this step is mandatory, while for streaming data its strongly advised, since it will form the basis for future historical data, as previously highlighted. Although, when working with real-time streaming data, there is a need to process such data in real-time and present results and insights in an almost immediate fashion, there is also a growing need to store real-time streaming data in a swift and efficient way. This subject will be revisited in the following chapters, but it is worth stating that there are several strategies to store streaming data while maintaining low latencies.

One of the most popular strategies is to store the incoming streams in a fast writing database, such as in the case of in-memory databases (e.g. Redis (RedisLabs, 2015)), time-series databases (e.g., InfluxDB (InfluxData, Inc., 2013)) or real-time analytical databases (e.g. Apache Druid (The Apache Software Foundation, 2019)), often through the use of background services that work in parallel with the main analytics processes to access data streams in order to store streaming data without interfering with the fast flow of real-time data analytics. These topics will be revisited thoroughly in Chapter 5.

3.2.2.2 Data Processing

The next sub-component in the proposed logical components and data flows model is Data Processing (Figure 3.4). This component is responsible for all the processing tasks supporting the actual application of both Data Management and Data Analytics processes, with the goal of supporting Big Data processing tasks. Data processing tasks include data transformation, such as data enrichment, data aggregation and data fusion, and data selection, querying and access. For both the Data Processing and Data Analytics sub-components, three different types of data processing are considered, as overviewed in sub-section 2.3.5, depending on the data type at hand (Costa C. F., 2019): Batch processing, stream processing and interactive processing.

As will be evident in Chapters 5 and 6, data processing can be considered as an underlying process used by the Data Management and Data Analytics sub-components to enhance the Big Data capabilities of their own processes. This means that, often, data processing concerns the Big Data processing engines used as a basis for the Data Analytics and Data Management processes, such as in the case of Apache Hadoop (The Apache Software Foundation, 2018) and Spark (The Apache Software Foundation, 2018) for batch processing or Apache Storm (The Apache Software Foundation, 2018) and Flink (The Apache Software Foundation,

2014) for stream processing. Examples for the use of such Big Data processing engines for Data Management and Data Analytics will be presented, respectively, in sub-sections 5.2.1, 6.1.1 and 6.1.2. Hence, the Data Processing sub-component will be approached in Chapters 5 and 6 as a necessary dependency of the Data Management and Data Analytics sub-components, when it comes to handle, process and analyse MobiTrafficBD.

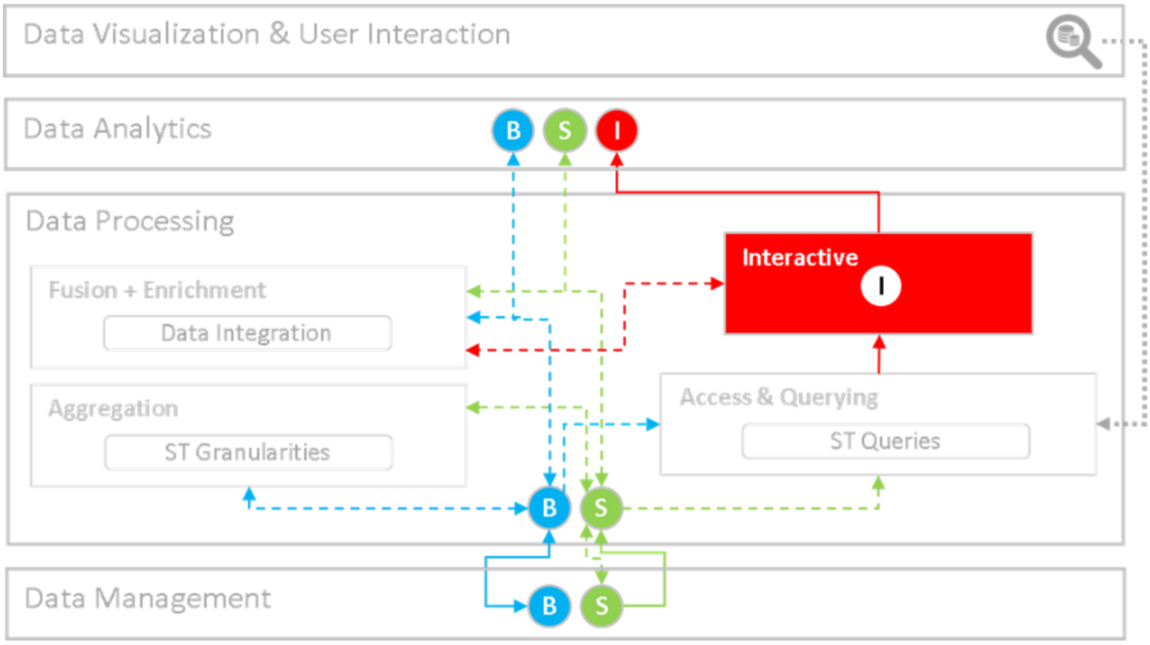


Figure 3.4 — The Data Processing Sub-component

Nevertheless, there are processes which are normally carried out by the Data Processing sub-component, such as data aggregation, data fusion, data enrichment and data access and querying, to point out a few. Within the context of the proposed methodology, data aggregation primarily focuses on combining spatiotemporal data through the adjustment of their spatiotemporal granularities. Different data sources often present different spatiotemporal granularities upon their collection as, for instance, in the case of sensor readings, which may be captured at different time intervals (e.g., a reading per each five minutes or each few seconds), or in the case of meteorological stations, which present different spatial granularities (e.g., weather data for a wider region or for a specific city). Hence, it is imperative that, when performing analytical processes over spatiotemporal data, these granularities are adjusted and conform to defined spatial and temporal ranges. Therefore, when analysing data from a particular spatial area at a particular time interval, all data sources available within such spatiotemporal range should be represented with similar granularities.

Data fusion refers to the combination of MobiTrafficBD with other data sets, to combine the original data sets to produce a more complete data set with more information, containing all the data attributes or associated metadata of the original sets. One example is the fusion of traffic sensor data with traffic event data and weather station data for a known location, or the fusion of traffic event data with the type of traffic event or the degree of severity of such event. Finally, data enrichment entails the creation of new attributes, derived from raw data as well as the extraction of patterns in data, resulting from complex event processing or other pattern mining methods. For instance, traffic sensor metadata may be fused with more information, such as the type of road in which the sensor is placed, the maximum speed allowed, the number of lanes, or other pertinent information. Data enrichment is useful for further contextualization of MobiTrafficBD, in the sense that it provides semantics and context to MobiTrafficBD. For instance, the simple process of reverse geocoding, which consists of getting a physical address from geographical coordinates, is considered a kind of data enrichment.

Finally, data access and querying enable users and external systems to query and select data according to their needs. This process is key for the top-down data-handling approach, already explained above. The data access and querying process starts with a direct interaction with an user or system that wants to access specific data (symbolized by the dashed grey arrow in Figure 3.4) in which the user/system resorts to queries and other data access mechanisms to limit the data attributes, scopes and ranges of the accessed data so that it better conforms with analyses' objectives. Data queries are the main data access mechanisms and enable data extraction from databases or other storage technologies using a query language. Queries can filter data through their spatiotemporality-defining attributes, filter out unwanted or irrelevant attributes and join/aggregate distinct datasets within the same database. Apart from direct queries, there are also visual-aided data access mechanisms, such as interactive filtering user interfaces, which use queries in their background processes, but present an intuitive interface for user interaction for data access and filtering. This subject will be revisited thoroughly in Chapter 6.

As a final note, all arrows in Figure 3.4 that come from the Data Management sub-component to the Data Processing sub-component are bidirectional because it may be necessary in some cases to store any data sets resulting from each of the Data Processing processes, such as fusion and aggregation. For instance, it may be necessary to store data in a harmonized temporal granularity, as in the case of traffic sensors, or to store data that was enriched or fused with other data sets, such as ST event data that was enriched with type of event, severity, type of road in which the event occurred, etc. Furthermore, the bidirectionality of some of the arrows, such as the bidirectional arrow between interactive data and the enrichment process, entail that data can go through several of the Data Processing processes.

3.2.2.3 Data Analytics, Data Visualization and User Interaction

The two last sub-components within the framework are Data Analytics and Data Visualization and User Interaction (Figure 3.5). The Data Analytics sub-component is responsible for applying Data Analytics, Data Mining, Machine Learning, Deep Learning or any other data analysis techniques available in the framework. In this component, the emphasis goes to data analysis models, methods and tools focused on MobiTrafficBD, such as the ones already overviewed in Chapter 2. Some examples are spatiotemporal clustering and classification, mobility- and traffic-related prediction models or pattern and anomaly detection and complex spatiotemporal event processing. Data may flow through and between processes or it may go directly to the next sub-component (Data Visualization and User Interaction), without undergoing any Data Analytics process, such as in the case of query results, in the form of interactive data, which may be directly presented via user interfaces and data sharing mechanisms.

General outcomes from the application of the necessary data analysis processes are represented as data analytics and data mining methods' results, which can also produce insights that will be used in decision-making support tasks, and trained models for classification, prediction or pattern and anomaly detection. The latter may also be stored in the Data Management sub-component for future reuse as basis for prediction, classification and other data analysis processes that are based on model training. These outcomes are then passed to the Data Visualization and User Interaction sub-component.

The Data Visualization & User Interaction sub-component is responsible for all methods of data delivery to end users. The User Interaction process is based on user queries. A user query is represented by the looking glass icon and the dashed grey arrow connected to the Access & Querying process in the Data Processing sub-component. To present information about the data available for the query, the User Interaction process receives batch data, in the form of metadata and of contextual boundaries for the different data attributes, for instance the temporal range, the maximum and minimum spatial coordinates or the numerical boundaries of attributes of the whole data stored in the database. The user can then use this information to know where to find the necessary data, to better construct the query and to limit the query's scope to the contextual boundaries. Users may receive queries' results through visual interfaces, which present the queried data in meaningful ways, such as using charts, maps or any kind of visual aids to better support data interpretation and understanding, or through data sharing platforms, which provide data in specific text formats, such as JSON or XML, but with no visual aid to support data interpretation.

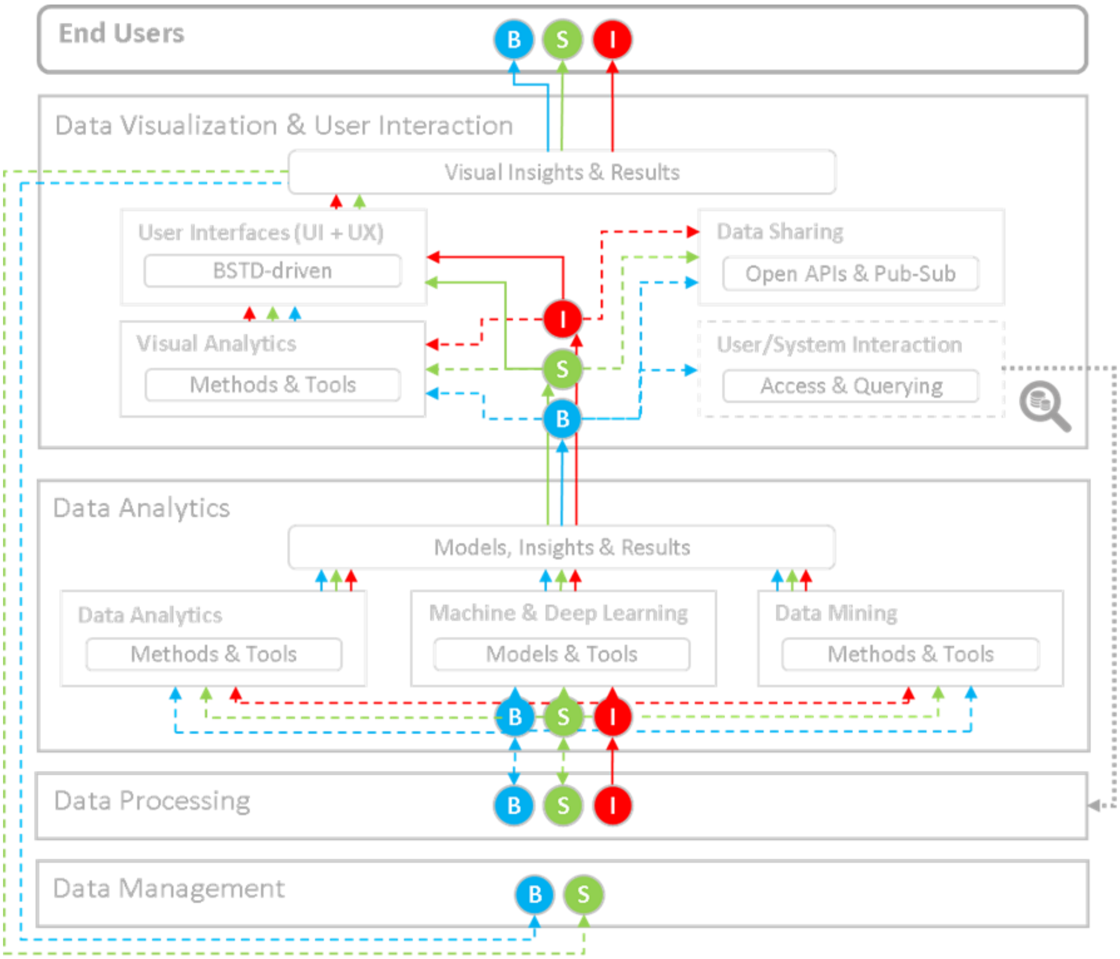


Figure 3.5 — The Data Analytics and Data Visualization and User Interaction Sub-components

The Data Sharing process is based on the already mentioned data sharing platforms, which use data transmission mechanisms, such as Application Programming Interfaces (APIs) and publish-subscribe mechanisms to share data with data consumers that are external to the framework. These mechanisms enable the delivery of batch, interactive and streaming data, not only as results and insights from the Data Analytics sub-component, but also as data retrieved directly from the Data Storage process. These mechanisms are of extreme importance because they provide a direct pipeline for data sharing and, working in conjunction with the Data Harmonization processes in the Data Management sub-component, enables data interoperability across the data sharing process. This is important due to the need to harmonize and make data more interoperable, both in terms of data model standards that can be reused across platforms, but also, particularly in the case of MobiTrafficBD, in terms of spatiotemporal attributes' characteristics, formats and reference systems.

The Visual Analytics process works directly with the User Interfaces process, providing custom visual aids and tools that better present the necessary data and that extract value and insights from the data, such as patterns and outliers, just through the application of visual methods, helping the users in their decision-making processes. Examples of visual aids are charts (bar charts, line charts, etc.), graphs (direct, networks, etc.), maps (geographical, heatmaps, choropleth, etc.) or any other data presenting method available. These visual aids are then integrated in user interfaces, both Mobile- or Web-based, to allow users to interact with and navigate through the data. Interaction often occurs through filtering, highlighting, clustering and aggregating data within the visual aid, and navigation is often associated to zooming or panning through the visual aid. These user-interface interactions are often translated to data queries or requests to the framework's components, but the user has no direct interaction with the query or request per se.

The User Interfaces process is responsible for providing the user interfaces and the visual aids to users in such a way that users are able not only to better interpret and understand the results and information presented to them, but also to provide insights that will guide them to better decisions and to achieve their goals. Quoting (Cook & Thomas, 2005), "Visual representations and interaction techniques take advantage of the human eye's broad bandwidth pathway into the mind to allow users to see, explore, and understand large amounts of information at once". Hence, it is important that these visual representations and interaction techniques are applied depending on the specific data at hand and that user experience (UX) is considered in order to better customize user interfaces (UI) to users' needs.

Particularly in MobiTrafficBD frameworks, both Visual Analytics methods and tools and User Interfaces must be MobiTrafficBD-driven, in the sense that visual aids, representations, interaction techniques or any other method should be applied depending on the nature and characteristics of the Big Spatiotemporal Data at hand and the context of Mobility and Traffic expert domains. For instance, user interaction through interfaces should be done with visual aids that enable spatiotemporal data selection (e.g., bounding boxes), navigation (e.g., zoom & pan, timelines) and interaction (e.g., spatiotemporal filtering and aggregation). Furthermore, users should be able to easily locate and contextualize Big Spatiotemporal Data, whether through the representation of real-world marks and infrastructure in the visual aids (e.g., roads and buildings represented in an interactive map) or using specific symbology associated to the Mobility and Traffic domains (e.g., use traffic accident, roadworks or traffic jam icons on an interactive map).

3.2.3 End Users

An end user is a person or system, external to the framework, that can execute one or more of the following actions: search and download data; analyse data (e.g., execute ad hoc queries, train/test data science models and apply Data Mining methods); consume reports, dashboards and other data visualization mechanisms; and include data, insights and results in business processes. These interactions more than often follow a demand-based interaction, in which end users initiate interactions and then wait for response from the framework (Costa C. F., 2019).

As already explained in Chapter 1, the main end users of MobiTrafficBD frameworks are RITMOs, the road infrastructure and traffic monitoring operators, since these operators are the main recipients of the results and insights provided by these frameworks. But they are not the only end users that can capitalize on insights and results coming from these frameworks. If the use case remains the same, i.e., mobility- and traffic-related data management and analysis, two other types of end user are considered.

First, if RITMOs provide the results to their main clients, or in other words, the everyday commuters and drivers, these results and insights may be used by these end users to better plan their daily trips (e.g., through traffic sensor data, commuters may know if there are traffic jams on their daily path) or to avoid traffic or mobility events (e.g., commuters may have access to traffic-related ST event information that show there is an accident or a public demonstration on their way, and may choose for a different route). Second, there is a growing need to share and integrate data across frameworks of the same of different domains (e.g., a public transportation data framework may use traffic information and insights to present delays or service abnormalities their clients). Hence, other type of end users are data consumers that access the data within the framework through APIs or publish-subscribe mechanisms.

Finally, other end user scenarios may be created within the Mobility and Traffic domains, such as in the case of Urban Planning, Emergency Management and Environmental Action, to name a few, or by changing the frameworks' scope from the Mobility and Traffic domains to other domain or area of interest in which Big Spatiotemporal Data is at the centre, such as in the case of Geographic Information Systems. This means that the prescriptive approach, guidelines and models presented in this work can be followed for use cases other than the ones related to Mobility and Traffic.

3.3 Technological Infrastructure Model

The model of logical components and data flows presented in Section 3.2 represents the starting point for the design of MobiTrafficBD frameworks, whereas the model of technological infrastructure presented in this section represents the starting point for their implementation. The technological infrastructure model, presented in Figure 3.6, focuses on technologies that can be the basis for instantiation of the different logical components and their associated processes, while also focusing on the physical infrastructure (hardware) that can be used to deploy MobiTrafficBD frameworks.

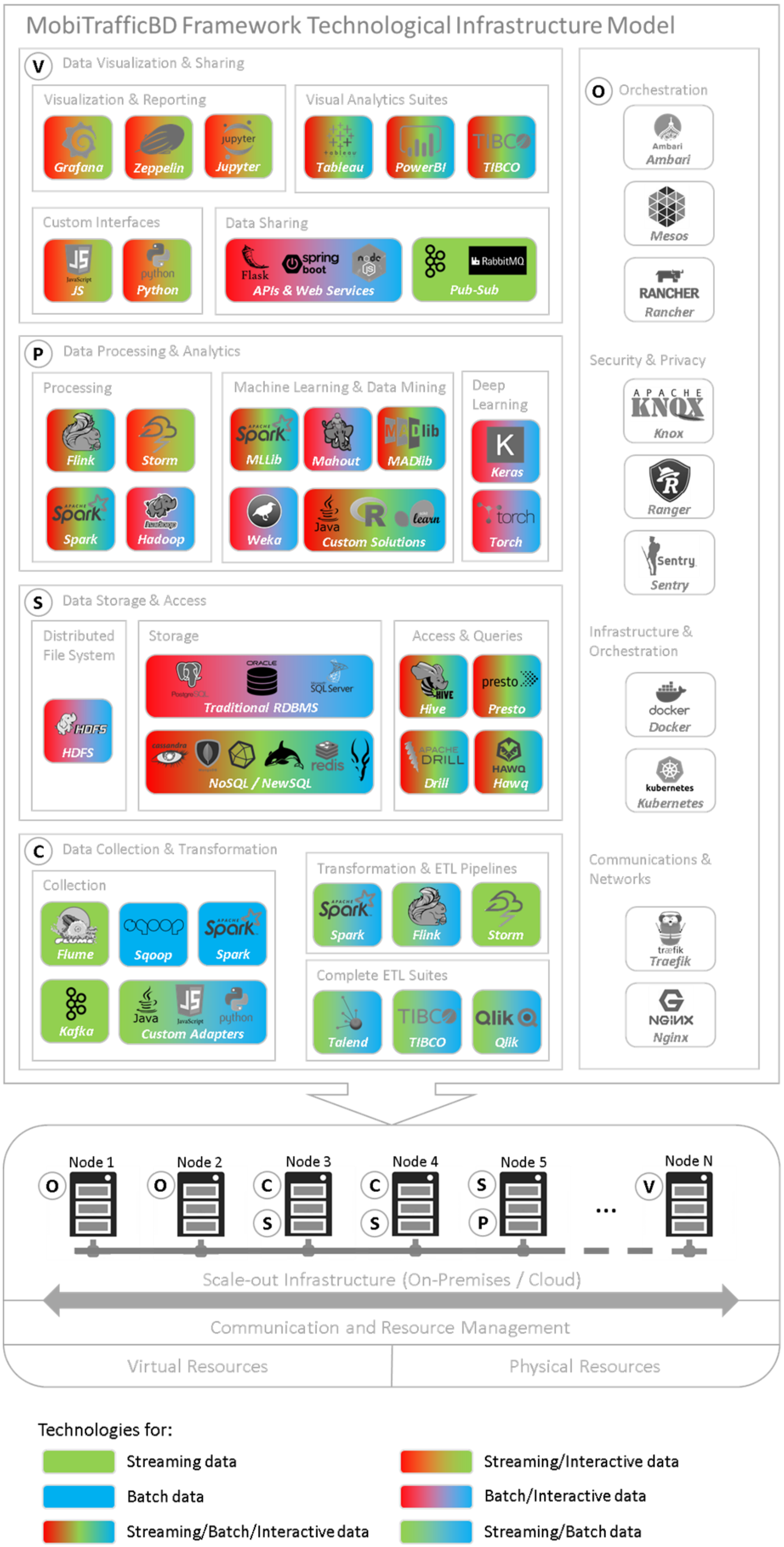


Figure 3.6 — Technological Infrastructure Model for MobiTrafficBD frameworks

Thus, the model of technological infrastructure, including several examples of state-of-the-art technologies for every logical component of the logical components and data flows model presented in Figure 3.2, enabling the direct association between both pictures, which in turn provides a consolidated, simple and coherent perspective of the design and implementation phases of MobiTrafficBD frameworks. This association is achieved not only by the presence of similar components across both pictures, but also by the application of the colour scheme of Figure 3.2 in Figure 3.6. Hence, the colours (blue, green, red) are used to represent the different data flows in Figure 3.2 (batch, stream, interactive data, respectively) that each technology in the model tackles.

The bottom part of Figure 3.6 depicts how a scale-out infrastructure, composed of physical and/or virtual resources and deployed on the cloud or on-premises, can support the application of the represented technologies, for each logical component, represented by its respective circled letter (O: Orchestration, C: Data Collection & Transformation, S: Data Storage & Access, P: Data Processing & Analytics, V: Data Visualization & Sharing). It is important to state that the technologies that are comprised in the technological infrastructure model of Figure 3.6 must be seen as suggestions and not preferential choices, because on one hand, the technological ecosystem for Big Data, although in rapid expansion, is already big enough to house a panoply of technologies that present the same characteristics and tackle the same challenges and issues.

On the other hand, the suggested technologies were chosen due to the projects and use cases in which this work is based upon and cannot be understood as preferred technologies over any other that may be suitable for the same data-driven challenges. Therefore, each logical component comprises several suitable technologies that must be considered as alternatives or complementary to other existing technologies, and not as mandatory or the most suitable for every implementation.

Starting from Data Collection & Transformation (Figure 3.7), technologies are divided as supporting data streams, data batches or both. On the streaming data side, Flume (The Apache Software Foundation, 2009) and Kafka (The Apache Software Foundation, 2017) are suitable technologies that can be applied to collect data streams, while on the batch data side, Sqoop (The Apache Software Foundation, 2011) can be used to collect data batches from relational databases and other sources and store them into the Hadoop Distributed File System (HDFS).

Another option to handle the collection of data batches is Apache Spark (The Apache Software Foundation, 2018). Although Spark is a Big Data processing engine, it allows for fast data collection, transformation and storage in several databases and other data storage paradigms, i.e., Big ETL, as presented in (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018). Moreover, it might be necessary to implement custom adapters for specific data collection scenarios, using well-established programming languages, such as Java, Javascript or Python. As in the case of Spark, other tools can also be used to build transformation and ETL pipelines. This is the case of known Big Data processing engines that can be applied to ETL tasks, such as Apache Flink (The Apache Software Foundation, 2014) and Spark itself, for both streaming and batch data pipelines, and Apache Storm (The Apache Software Foundation, 2018) exclusively for streaming data pipelines.

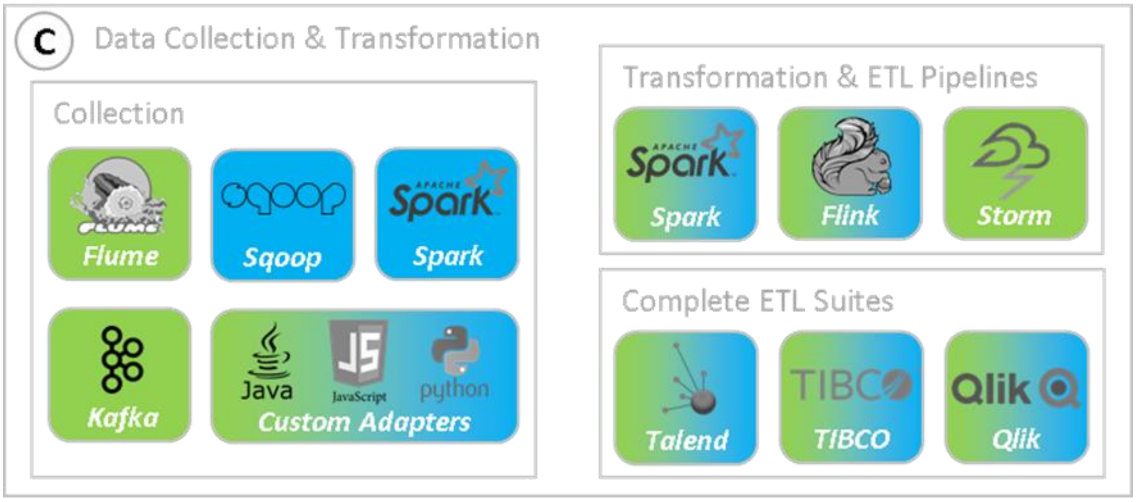


Figure 3.7 — Example Technologies for Data Collection & Transformation

Finally, there are also fully fledged ETL suites oriented towards Big Data scenarios, such as, for instance, Talend Big Data (Talend, 2006), TIBCO Jaspersoft ETL (TIBCO Software Inc., 2020) and Qlik ETL Solution (QlikTech International AB, 1993) suites, which come packed with tools and components for both streaming and batch data ETL. Some suites come with specific tools to handle spatiotemporal data, while in other cases, there might be the need to add a spatial or spatiotemporal extension to the suite, such as in the case of Talend, which has an extension for spatial data (Prunayre, et al., 2007). On the downside, this kind of Big ETL suites have paid subscriptions, and their open-source, free or community versions often just provide integrated user interfaces to build pipelines and submit tasks, and typically make use of other technologies to assure adequate distributed processing, such as the ones already presented (Apache Spark, Storm, etc.), since its native tools may not be scalable.

In the case of Data Storage & Access (Figure 3.8), the example technologies range from distributed file systems to data access mechanisms, going through relational database management systems (RDBMS) and NewSQL and NoSQL database systems. The most common distributed file system is the Hadoop Distributed File System (HDFS), as it is the most widely used and enables storage of all kinds of data, structured and unstructured, in a file-driven approach. HDFS is used by other data access technologies to store tabular data, as in RDBMS, such as in the case of Apache Hive (The Apache Software Foundation, 2011), which uses HDFS to store data in a tabular form, using Hive tables. Likewise, HDFS enables storage of data in analytics-oriented file formats, such as Parquet (The Apache Software Foundation, 2018) or ORC (The Apache Software Foundation, 2020) (both these file formats enable spatial data storage via binary formats (Vonk, 2015) or via spatial extensions (Roche, 2019)).

When the Big Data paradigm first emerged, RDBMS were the traditional database systems, widely used across the industrial and academic sectors. But this new paradigm threatened to kill RDBMS, since these systems did not meet the necessary requirements to handle large volumes (in the order of Terabytes, instead of the traditional Gigabytes) of often unstructured data with high variety and, sometimes, generated at unprecedented speeds. RDBMS were built to store and access data in tabular form and is based on a centralized architecture, meaning that it is vertically scalable (scales by adding more machines) and not distributed. These and other characteristics of RDBMS do not work well with Big Data.

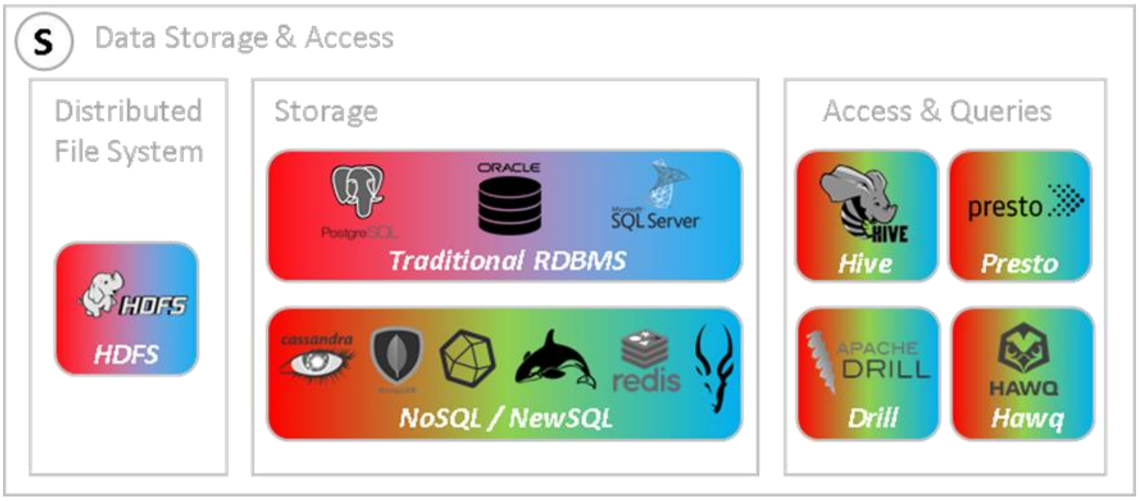


Figure 3.8 — Example Technologies for Data Storage & Access

First, a generous part of Big Data is comprised by unstructured data, and RDBMS are more suitable for structured data; Second, vertical scalability is not adequate for distributed architectures, as adding more and more computing power to a single machine is not possible, whereas Big data takes a "scale out" approach in which new machines can be added to the distributed cluster, providing more storage space, processing power and fault tolerance, through replication of information across multiple nodes; Third, RDBMS performance degrades rapidly when storing increasing data volumes, in terms of throughput and query response times.

Hence, new data storage paradigms emerged to cope with the problems of RDBMS towards Big Data, namely handling ever-growing volumes of batch data or fast-running streaming data. These new paradigms, NoSQL and NewSQL, already introduced in the previous chapter, brought new ways of storing and retrieving data from databases. From document-oriented (e.g. MongoDB (MongoDB, Inc., 2015)) and in-memory databases (e.g. Redis

(RedisLabs, 2015)) to big table (e.g. Apache Hbase (The Apache Software Foundation, 2007) and Cassandra (The Apache Software Foundation, 2016)) and SQL-on-Hadoop (e.g. Apache Impala (The Apache Software Foundation, 2015)) stores, NoSQL presented several approaches for distributed database systems that could "scale out" horizontally, instead of the vertical scaling of RDBMS, meaning that NoSQL databases may be replicated across nodes and machines. Moreover, NewSQL databases tried to bring the best from both worlds, seeking to provide the scalability of NoSQL systems for online transaction processing (OLTP) workloads while maintaining the characteristics of a traditional RDBMS (Pavlo & Aslett, 2016). Besides the examples already presented in the previous chapter, a good example of NewSQL databases are distributed timeseries databases, such as InfluxDB (InfluxData, Inc., 2013).

Nevertheless, in the last years, RDBMS are evolving to present new Big Data capabilities. For instance, PostgreSQL added several new features such as BRIN indexing, which is based on small but very effective indexes for very large, naturally ordered tables, faster sorts and query results' summarization techniques, among others (The PostgreSQL Global Development Group, 2016). More lately, other enhancements and extensions were created to bring PostgreSQL closer to Big Data, such as enabling JSON documents storage or key-value data storage, to cope with unstructured and semi-structured data storage, or the creation of GreenPlum (VMware, Inc, 2020), a massively parallel processing (MPP) database system, based on PostgreSQL, specifically designed for fast analytics. On the other hand, cloud service providers, such as Oracle and Microsoft enabled vertical scalability in the cloud for their proprietary RDBMS (Oracle Database (Oracle Corporation, 1979) and Microsoft SQL Server (Microsoft, 2019), respectively). Even so, RDBMS still have a long way to go, when it comes to handling fast streams of data, making RDBMS better suited for batch data storage and interactive data retrieval.

The modularity of the proposed prescriptive approach and technological infrastructure model allows for flexible technological choices when implementing MobiTrafficBD frameworks, while maintaining the architectural construct and data management guidelines. Ultimately, the choice of storage technologies, or any other for that matter, is solely dependent on the use case at hand and it is advised that practitioners perform preliminary analyses when making technological choices, whether for storage or any other component, due to the fast evolution of the Big Data technological ecosystem. This subject will be further explored in the next chapter.

Lastly, data access and querying technologies provide interactive SQL interfaces to query both batch- and streaming-based storage systems, focusing on NoSQL and Hadoop-supported storage technologies. That is why these systems are frequently coined as SQL-on-Hadoop systems, although they also support other NoSQL and NewSQL systems. From the panoply of existing alternatives, Apache's Hive (The Apache Software Foundation, 2011), Drill (The Apache Software Foundation, 2012) and HAWQ (The Apache Software Foundation, 2020), and Presto (The Presto Software Foundation , 2013) are highlighted in Figure 3.6. All of the above have some kind of spatiotemporal querying support, whether through built-in procedures or through extensions, and evaluating their performance benchmarks, their connectivity with storage technologies and the overall efficiency of spatiotemporal and non-spatiotemporal queries is of major relevance to implement an adequate data access and querying component in MobiTrafficBD frameworks.

The Data Processing & Analytics component (Figure 3.9) corresponds to the Data Processing and Data Analytics components in the logical components and data flows model of Figure 3.2. For data processing, the already mentioned technologies for ETL pipelines are, in fact, originally built to implement data processing pipelines for enrichment, aggregation, summarization and other processes. These technologies are Apache Hadoop for batch and interactive data processing, Apache Storm for streaming and interactive (using sliding windows) data processing and Apache Spark and Flink, for both cases. The above data processing technologies allow for spatiotemporal data processing, whether out-of-the-box or through extensions. For a thorough analysis on the spatiotemporal data processing capabilities of Apache Hadoop, Spark, Flink and others, please refer to (Karim, Soomro, & Burney, 2018), while for a comprehensive usage example of Apache Storm as an engine for real-time spatial queries, please refer to (Zhang F. , et al., 2016).

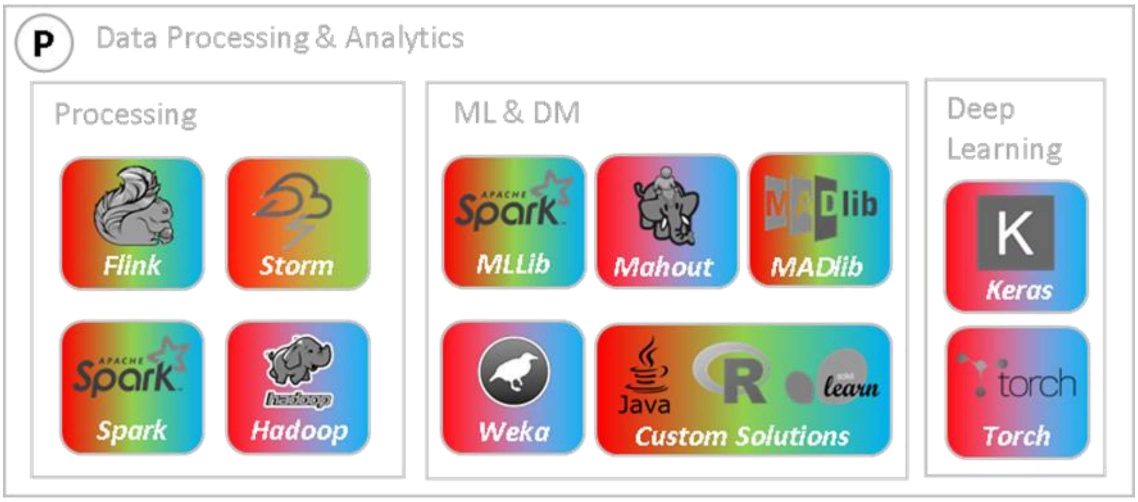


Figure 3.9 — Example Technologies for Data Processing & Analytics

Data Analytics technologies differ mainly in terms of their suitability for batch and streaming data analytics and the algorithms and methods that each technology comprises. There are several data analytics solutions, each of which has its own purpose and specificities. The Apache Spark Machine Learning Library (Spark MLlib) (The Apache Software Foundation, 2018) is a machine learning library that makes use of the distributed processing capabilities of Spark to run Machine Learning (ML) methods on large volumes of data, but it also provides ML methods for data streams. Other example of a distributed analytics library is Apache MADlib (The Apache Software Foundation, 2020), which is built to run directly on PostgreSQL or GreenPlum engines, enabling local, in-database ML procedures directly on the data side, for RDBMS or MPP. Finally, another example of a Data Mining (DM) and ML library is Weka (The University of Waikato, 2005), which allows for distributed DM and ML, using Apache Spark as its base processing engine.

But what happens when the libraries do not comprise the necessary algorithm or method for the use case at hand? This issue occurs a lot for spatiotemporal data, since most of the libraries and technologies already mentioned are not purposely built for spatiotemporal DM and ML. In this case, a custom implementation, using Java, the R language or one of the many Python-based DM and ML libraries, such as Scikit Learn (Pedregosa, et al., 2011), may be the best solution. Another option is the Apache Mahout (The Apache Software Foundation, 2014), which is a distributed linear algebra framework that has a mathematically expressive domain-specific language to enable the development of distributed DM and ML algorithms, running in both Apache Hadoop and Apache Spark.

Deep Learning is a relatively new concept that has been brought to light by academia and industry in the past few years, mainly due to the evolution of computing power, data storage and the sheer amounts of data produced nowadays. Today, there is no generic data-driven framework that does not take into consideration this new class of Machine Learning methods (Schmidhuber, 2015). Deep Learning methods are especially relevant for unsupervised learning and pattern discovery from unstructured data, through the use of complex and multi-layered implementations of neural networks (e.g., convolutional, recurrent, etc.), which mimic the way the human brain is organized into layers upon layers of neurons. Such methods need great processing power and huge amounts of batch data to deliver results. The two examples in Figure 3.6, Keras (Chollet, 2015) and Torch (Collobert, Bengio, & Mariéthoz, 2017), are the most used Deep Learning tools by both industry and academia. Keras is a Python API for TensorFlow (Google Brain Team, 2015), a Deep learning library developed by Google, whereas Torch, and its Python counterpart, PyTorch (Paszke, Gross, Chintala, & Chanan, 2016), is a scientific computing framework built to run on top of clusters of graphical processing units (GPU). Both Keras and Torch can be deployed in clusters or cloud platforms, whether they are based on central processing units (CPU) or GPU. For reference, there are already several works, many of which use the above example technologies, that demonstrate scenarios of deep learning for spatiotemporal analytics (Tan, Liu, & Liu, 2020).

The Data Visualization & Sharing (Figure 3.10) is the end component of the data pipeline that is the basis of MobiTrafficBD frameworks, since it is responsible for delivering to the end user not only the results and insights coming from the Data Processing and Data Analytics sub-components of Figure 3.2, through rich visual analytics methods and visualization techniques, but also to deliver data directly to end users and external systems, through data sharing tools, such as APIs or publish-subscribe mechanisms.

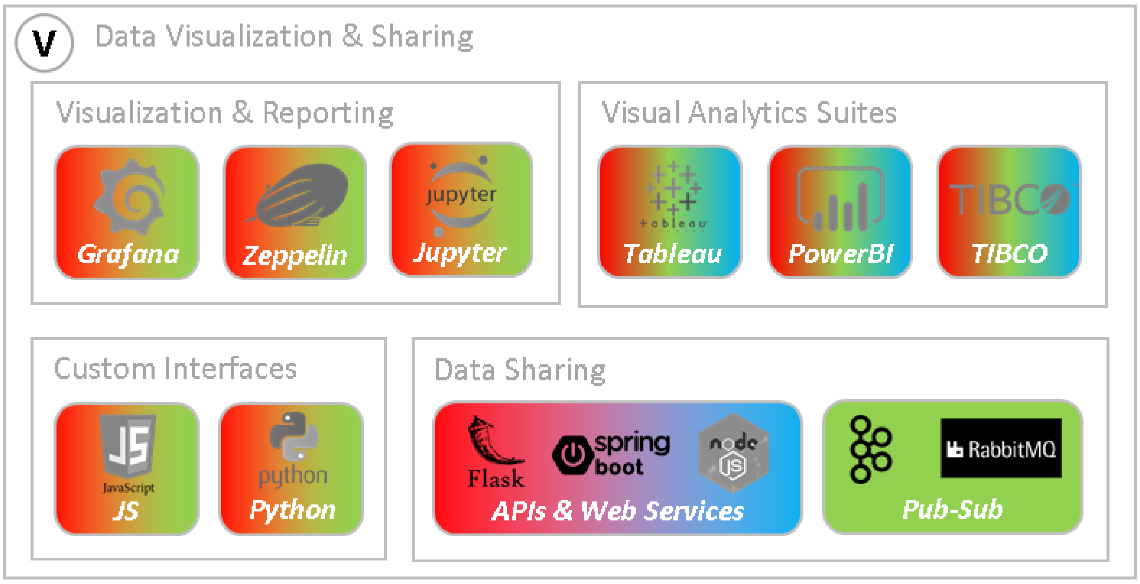


Figure 3.10 — Example Technologies for Data Visualization & Sharing

Hence, besides the visual analytics and visualization technologies, this component also comprises the data sharing technologies, such as any Web Service framework built in any programming language (primarily for batch and interactive data) or publish-subscribe mechanisms (specifically for streaming data) available in the market. Some examples are the Spring Boot framework for Java (VMware, Inc., 2020), NodeJS for Javascript (OpenJS Foundation, 2009) or the Flask framework for Python (Ronacher, 2010). These frameworks enable fast Web Service delivery for data sharing processes. Furthermore, data may also be shared through publish-subscribe mechanisms, as in the case of Apache Kafka or other message queuing tools, such as RabbitMQ (VMware, Inc., 2007).

Visual analytics is based on the conjunction of insights originated in the data analytics tools and enhanced by data-specific visualization techniques, in such a way that the end user recognizes better the patterns and insights in the data, opposed to when these visualization techniques are not used. Thus, two main options are possible. The first option is to use complete data analytics suites that comprise all the tools for data collection, transformation, processing, analytics and visualization, although these services may be used together or as independent services. Examples of data analytics suites are Tableau (Tableau Software, 2003), Microsoft's PowerBI (Microsoft, 2011) and TIBCO's Spotfire Analytics (TIBCO Software Inc., 2007). These suites are often proprietary and paid, although some of them have free editions, whether for the whole community or for academic purposes, such as in the case of Tableau.

The second option is to take advantage of the data analytics technologies chosen in the previous Data Processing & Analytics component, and couple them directly with visualization tools, developing and applying custom visual analytics solutions. In this case, practitioners have full control over customization and implementation of the visual analytics processes, but are more prone to less effective visual analytics results, since the choice of visualization and analytics methods and tools is not trivial and requires practitioners to be experienced data scientists, with know-how in both data analytics and visual analytics. Hence, practitioners and researchers may opt to use available, out-of-the-box visualization and reporting solutions, or to develop their own custom visualization services, using libraries and tools specific for programming languages.

Examples of the former are the Grafana visual analytics and interactive visualization platform (Grafana Labs, 2018), which provides several types of visual tools to build interactive data dashboards and supports adaptors and query builders for a wide range of data sources, Apache Zeppelin (The Apache Software Foundation, 2016), a Web-based notebook for visual discovery and analytics that serves as visualization and reporting suite for a number of data storage and processing technologies in the Apache Hadoop ecosystem and other solutions, and Jupyter (Project Jupyter, 2014), a Web-based interactive development environment for reporting notebooks containing live code, equations and visualizations that can be used for data cleaning, transformation, simulation, modelling and visualization. Regarding the latter, visualizations can be custom built using libraries and tools for the most common programming languages, as in the case of Javascript and Python.

The complexity of the technological infrastructure model entails the necessity of deploying technologies that support communication and networking between the infrastructural resources, optimize setup times and ease deployment, manage and monitor the whole infrastructure and guarantee data security and privacy. Some examples for these technologies are presented on the right side of Figure 3.6. Regarding networking and communications, reverse proxies such as Traefik (TraefikLabs, 2016) and Nginx (F5, Inc., 2004), are responsible for user requests' load balancing, data compression, provision of access and authentication mechanisms as well as other security features and network management between physical and virtual resources.

For easily configurable and deployable infrastructures, containerization technologies are recommended, such as Docker (Docker, Inc., 2013) and Kubernetes (The Kubernetes Authors , 2014) as the most popular solutions, since they bundle application code together with the related configuration files, libraries, and dependencies required for it to run in any environment, eliminating the problem of deploying the same system in different environments and resources (on-premises vs. cloud environments, physical vs. virtual resources). Containers are software packages that are abstracted away from any host operating system, and hence, stand alone and become portable, being able to run across any platform or cloud, free of configuration issues. Containerization technologies enable automated configuration, deployment, scaling and management of distributed applications, allowing for custom distribution of technologies across physical and virtual resources.

There are several technologies that enable security and privacy across all layers of Big Data-based frameworks. Besides specific encryption and access control mechanisms, security technologies' examples are Apache Knox (The Apache Software Foundation, 2018), Ranger (The Apache Software Foundation, 2011) and Sentry (The Apache Software Foundation, 2011). Knox enables infrastructure perimeter security, by hiding clusters' access point and blocking service details, Ranger is a framework to enable, monitor and manage comprehensive data security across the Hadoop ecosystem, whereas Sentry is centralized platform for policy administration, authorization, auditing, and data protection. Finally, there is also the need for technologies that orchestrate and manage all technological and infrastructural aspects of MobiTrafficBD frameworks. Examples of such technologies are Apache Ambari (The Apache Software Foundation, 2019), which enables provisioning, management and monitoring of Hadoop ecosystem-based clusters and integration of technologies on the Hadoop ecosystem with the existing enterprise infrastructure, Apache Mesos (The Apache Software Foundation, 2012), which is distributed systems kernel that enables abstraction of physical and virtual resources, enabling fault-tolerant and elastic distributed systems to easily be built and deployed, and Rancher (Rancher Labs, 2014), a complete software stack for Kubernetes-based infrastructures' management.

As a final note, the example scale-out infrastructure presented in the bottom part of Figure 3.6 may be optimized by having the nodes deployed as single containers that can house one or more instances of the example technologies, allowing for easy configuration, deployment and scaling of the whole infrastructure. Communication and networking between containers should be managed by reverse proxies, and security and privacy technologies should also be deployed across the whole containerized infrastructure.

3.4 General Guidelines and Best Practices

Concluding, this section presents the most relevant guidelines that should be taken into consideration when deploying an adequate infrastructure for MobiTrafficBD frameworks:

1. Follow the given requirements throughout the project's lifecycle.

2. Choose a Reference Architecture, even if it is only for Big Data, considering that there is no reference architecture that consolidates both Big Data and spatiotemporal data aspects.
3. Add spatiotemporality aspects to a Reference Architecture through the use of technologies, tools and methods that support Big spatiotemporal data lifecycle management and analysis.
4. Spatiotemporality must be represented in all the layers of the developed architecture.
5. The design of the infrastructure should aim at horizontal scaling ("scale out"), in order to reduce costs, optimize performance and to capitalize on the full capacity of existing and emergent technologies within the Big Data ecosystem.
6. Storage of both raw and harmonized data is strongly recommended for both batch and streaming data, introducing an ELTL approach (hybridization between ETL and ELT).
7. The developed architectural model, although based on generic Big spatiotemporal data-driven guidelines, should be also driven by the use case at hand, even when the use case is not comprised in the Mobility and Traffic domain. Noting the fact that it is focused on the Mobility and Traffic domain, the proposed architecture is generic enough to cope with other use cases in different, Big spatiotemporal data-based domains.
8. The technologies comprised by the technological infrastructure model must be seen as non-mandatory examples but can serve as initial clues for practitioners and researchers when choosing the technology stack, in the design phase of any MobiTrafficBD framework implementation. These clues are the starting point to choose the right technology to the specific data-driven use case at hand.
9. Despite the completely free choice of technologies, it is recommended that, depending on the data characteristics and the use case at hand, practitioners should opt for technologies that can be reused, "recycled" and repurposed to different goals, shortening the framework's technology stack. This will enable not only the reduction of complexity and time for technological infrastructure configuration processes, but also will minimize the risk of poor compatibility between technologies, models and data formats, for instance.
10. Easy deployment and dynamic configuration supported by containerization strategies and technologies.
11. The logical components and data flows model (Figure 3.2) can be seen as a modular architecture, which enables the selection of individual modules (components) depending on the data flows to be managed and explored, the use case at hand and according to the selected data analysis and exploration methodology, as will be evident in

Chapters 5 and 6. This modularity of the prescriptive methodology is further enabled by the application of technology distribution and containerization guidelines provided in the technological infrastructure model (Figure 3.6).

12. Follow the best practices for optimized technology distribution across containers.

The way technologies are distributed across containers should be optimized by taking into account the following best practices:

- All technologies should be replicated on two or more containers to enable fault-tolerance and better computing performance. Emphasis is given to orchestration, storage and processing technologies.
- Certain technologies should be co-located in the same container to avoid unnecessary data movement across nodes. The main example is the co-location of storage and processing technologies, preventing network bottlenecks.

4

MobiTrafficBD Frameworks in Practice

Now that the model for logical components and data flows (Figure 3.2) and for the technological infrastructure model (Figure 3.6) for MobiTrafficBD frameworks are presented, this chapter explores several MobiTrafficBD frameworks' contexts and scenarios, as it may be necessary to use more practical examples for researchers and practitioners to master the proposed general guidelines, presented in the previous chapter. But the examples presented in this chapter will primarily serve as real-world demonstrators for the applicability of the proposed prescriptive methodology in real scenarios. Furthermore, the examples will be used to clarify some of the guidelines provided previously, and to evaluate their suitability in a broader scope of Mobility- and Traffic-related applications focused on Mobility and Traffic Monitoring Services for Intra-city, Inter-city and Country-wide scenarios, Proactive Improvement of Transport Systems Quality and Efficiency, Proactive Charging Schemes for Freight Transport and Public Transportation Network Monitoring and Optimization.

Hence, this chapter introduces examples of research-driven, real-world use cases for the design and development of MobiTrafficBD frameworks, following the proposed models, guidelines and best practices of the previous chapter. The technical details of these use cases will then be further explored in the next chapters.

4.1 FP7 MobiS Project

As already mentioned in Chapter 1, the main goal of the EC's Framework Programme 7 (FP7) MobiS project (European Commission, 2012) was to create a new concept and solution of a federated, customized and intelligent mobility platform by applying novel Future Internet technologies and Artificial Intelligence methods that monitor, model and manage the urban mobility complex network of people, objects, natural, social and business environments in real-time. MobiS federation and intelligence was based on the symbiotic relation between these stakeholders, innovative prediction and reasoning methods that use learned multi-criteria function to provide more efficient, energy-aware and environmentally friendly citizen mobility. MobiS aimed to federate novel artificial intelligence services and traditional information platform services coming from (i) existing transport private or public service providers, (ii) ambient data, based on sensor infrastructures, and (iii) social networking data.

To achieve its objectives, the project developed the MobiS federated platform, prediction/planning/reasoning services, multi-criteria decision function and federated mobility-based services that correspond to the above-mentioned information sources. Solutions were tested in three pilots:

- An inter-city mobility scenario in Sweden (Stockholm-Hudiksvall-Sundsvall).
- An intra-city scenario in Greece (Thessaloniki).
- A country-wide (inter-city) mobility scenario in Slovenia.

Since only a specific use case on the detection of traffic events through the application of Data Mining over Twitter data is going to be described as an example for the presented methodological approach, the author will not go deeper into the specifics of each pilot. For more information about the MobiS project, please refer to (European Commission, 2012). The use case will be presented in Section 4.4.

4.2 Horizon 2020 OPTIMUM Project

The Horizon 2020 OPTIMUM project (European Commission, 2015) is the major contributor to this thesis work. It was a European Commission-funded project that started in May 2015 and ended in August 2018 and had the grand objective of establishing a largely scalable, distributed architecture for the management and processing of multisource Big Data-enabling continuous monitoring of the transportation systems needs and providing data-driven mobility services based on proactive decisions and actions in an (semi-) automatic way, following a cognitive approach based on the Observe, Orient, Decide, Act (OODA) loop of the big data supply chain for continuous situational awareness (Galinec & Steingartner, 2013; Chan, Gawlick, Ghoneimy, & Liu, 2014).

The overall OPTIMUM architecture and the OODA loop for Big Data situational awareness may be directly mapped to the model of logical components and data flows proposed in Figure 3.2. The Observe phase can be mapped to the Data Management component of Figure 3.3 and accounts for the capture and ETL tasks over Big Data sets produced from a panoply of data sources, such as novel types of sensors and communication capabilities in vehicle and the traffic infrastructure, social networks or public transportation and mobility in general, just to name a few. The Orient phase corresponds to the Data Processing component of Figure 3.4, providing data fusion, aggregation and contextualization services, supported by online stream analytics and the Big Data framework that allow fusion of historical and real-time information from multiple sources, but this phase has already certain aspects of the Data Analytics component, since it is responsible for the traffic forecasting engine, complex event processing methods and a suite of statistical and stochastic techniques that support the predictive functionality of the forecasting engine. The Decide phase represents the Data Analytics component of Figure 3.5, comprising advanced analytics methods for system aware optimization, such as system-optimal multi-modal routing algorithms and innovative charging models, and contextualization, through on analytics methods for ecological footprint calculation and dynamic toll charging schemes and models. Finally, the Act phase may be mapped to both the Data Analytics and the Visualization and User Interaction components of Figure 3.5 with a strong focus on personalization, application of persuasive and user profiling strategies and recommendation services.

The project comprised a consortium of eighteen partners from eight European countries and was responsible for deploying and validating the proposed OPTIMUM architecture in three distinct pilot studies, each of which with specific use cases. The first pilot study was coined as proactive improvement of transport systems' quality and efficiency and was deployed in three different countries, namely in the city of Vienna, Austria, in the city of Ljubljana, Slovenia and in the city of Birmingham, United Kingdom. The second pilot study was deployed in Portugal and had the objective of creating a dynamic highway tolling system for freight transport, to reduce congestion by shifting some traffic to alternative times, routes or modes, or by eliminating trips. Finally, the last pilot study was held in Slovenia and had the objective of building an interactive Car2X (car to car, car to infrastructure) communication platform, using next-generation campervans. Since the author was directly involved in the first two pilots, these will be further explored in sub-sections 4.2.1and 4.2.2.

4.2.1 Pilot Study 1: Proactive improvement of transport systems quality and efficiency

As described in the OPTIMUM project's Web site (OPTIMUM Consortium, 2015), "complex urban transportation networks already offer a multitude of modalities and options, including public means such as trains, metros, buses, taxis, shared bicycles, shared cars and electric vehicles. New types of modalities are also expected to emerge within a diverse ecosystem of public, private and non-profit entities. Ideally, an integrated transportation network allows citizens to move easily from point “A” to point “B” regardless of mode or service provider, while sustaining overall user well-being and keeping greenhouse emissions to a minimum. Nonetheless, current systems are fragmented, and attaining a high quality of service while providing a safe, dependable, convenient and comfortable experience for all individuals — while also taking into account ITS optimisations — is not an easy task.

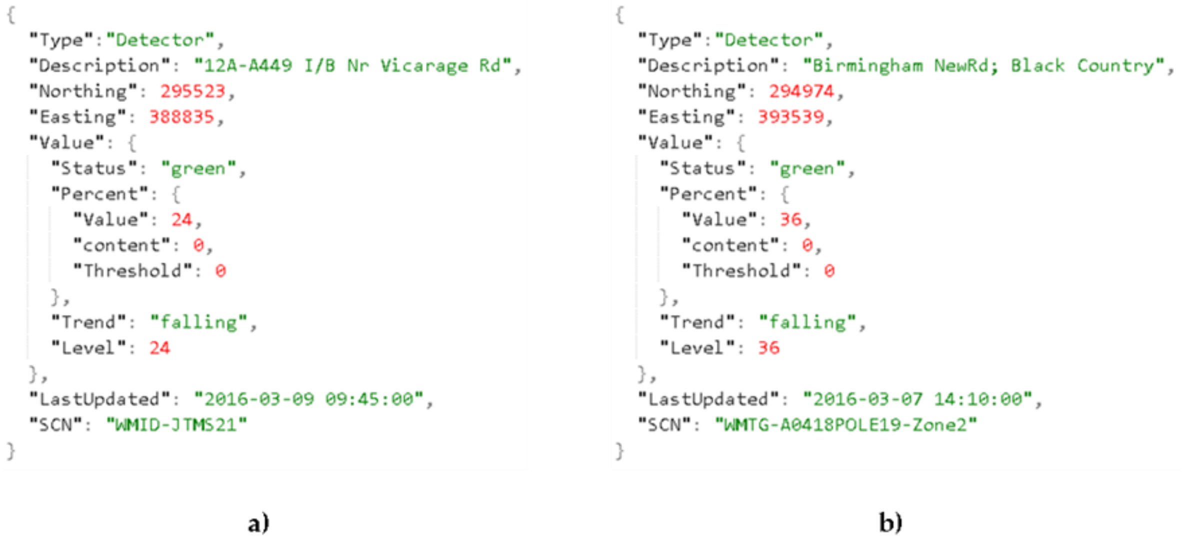
The OPTIMUM platform's instantiation for Pilot Study 1 was deployed in three major EU cities, each one with different characteristics in terms of data sources and transportation system particularities. The main aim of the study was to improve the quality and efficiency of multimodal trips, and Intelligent Transportation Systems as a whole, by supporting proactive decisions driven by transportation network and crowdsourcing information data. A multitude of information can be retrieved from the mobility data of people using multimodal and interoperable transport systems. Recent data collection technologies and analysis methods, in combination with modern transport telematics systems, allow for the identification of transport modes and the study of user habits. This provides the basis for transport information systems and models, which in turn increases the efficiency of an entire transport system.

The main aim of the three urban pilot studies was to proactively facilitate decision making for efficient integration of transport modes. This was achieved by implementing a smart multimodal transit concept, which lead to improved quality, accessibility and utilisation of interconnected transport systems. Thus, a complex model of the current traffic conditions, and a short-term prediction of these conditions, was realised on top of advanced real-time predictive analytics and a multitude of transport information".

This pilot study was supported by several data sources for every use case (city), most of them being characterized as MobiTrafficBD, and with records collected throughout the period of the project. The data sources relevant for the examples of the next chapters are described in the following sections.

4.2.1.1 Data Sources: Birmingham City, UK

The Birmingham City Council provides various open data sets to the public, sharing data through Representational State Transfer (REST) Web services. Some examples of data types shared by the council are vehicle flows, average speeds, road occupancy, travel times and congestions', incidents' and roadworks' information, coming from circa 3500 traffic sensors. Figure 4.1 presents two sample sensor readings from the Birmingham City Council REST Web services, represented in JSON.



```
{
  "Type":"Detector",
  "Description": "12A-A449 I/B Nr Vicarage Rd",
  "Northing": 295523,
  "Easting": 388835,
  "Value": {
    "Status": "green",
    "Percent": {
      "Value": 24,
      "content": 0,
      "Threshold": 0
    },
    "Trend": "falling",
    "Level": 24
  },
  "LastUpdated": "2016-03-09 09:45:00",
  "SCN": "WMID-JTMS21"
}
```

```
{
  "Type":"Detector",
  "Description": "Birmingham NewRd; Black Country",
  "Northing": 294974,
  "Easting": 393539,
  "Value": {
    "Status": "green",
    "Percent": {
      "Value": 36,
      "content": 0,
      "Threshold": 0
    },
    "Trend": "falling",
    "Level": 36
  },
  "LastUpdated": "2016-03-07 14:10:00",
  "SCN": "WMTG-A0418POLE19-Zone2"
}
```

Figure 4.1 — Birmingham City Council sensor data source examples. a) flow sensor reading; b) average speed sensor reading

Both examples are similar in their structure, but they are retrieved from different Web services, each of which is specific to one of the available data types. In terms of data availability, Table 4.1 presents the average daily availability for each of the data types.

Table 4.1 — Birmingham City Council Web service data types, record volumes and size

Data Type	Average Daily Records	Average Hourly Records	Daily Storage Size (MB)
Flows	99.948	4.164	134,80
Average Speeds	2.117.476	9.062	293,32
Travel Times	103.812	4.326	140,02
Congestions	11.881	495	16,02
Occupancies	5.088	212	6,86
Total	438.205	18.259	591,02

Besides the data sources provided by the Birmingham City Council, nation-wide data sources were also provided, mainly through UK's National Traffic Information System (NTIS) (Highways England, 2015). NTIS provides Web services to push both real-time and historic data to subscribers. These services publish data from a panoply of sensor types, such as Motorway Incident and Automatic Signalling (MIDAS), Traffic Management Unit (TMU), Automatic Number Plate Recognition (ANPR) and fused sensors, totalling 2928 sensors scattered across UK's road infrastructure, with a temporal range starting at April 2016 and ending in

October 2016. Table 4.2 presents the average hourly and daily number of records for each NTIS sensor type.

Table 4.2 — Average daily and hourly record number per NTIS sensor type

Sensor Type	Average Daily Records	Average Hourly Records
MIDAS	720.000	30.000
TMU	144.000	6.000
ANPR	432.000	18.000
Total	1.296.000	54.000

Data records are collected at different temporal granularities, with MIDAS and fused sensor data being collected every minute, and TMU and ANPR sensor data being captured once every five minutes, totalling more than 55 million records. Detailed specifications can be found in (Highways England, 2008) and examples of these data sources, provided in the DATEX II standard format, are presented, respectively, in Appendixes A.1 (page 235) to A.4 (page 240).

4.2.1.2 Data Sources: Ljubljana, Slovenia

As in the previous case, Slovenia provides nation-wide information about traffic and mobility, from traffic sensor and event data to wind conditions and public transport information, through public APIs (Žejn, et al., 2015). Traffic sensor, event and wind conditions data are provided in JSON format and example data representations are depicted in Appendixes A.5 (page 241), A.6 (page 243) and A.7 (page 245), respectively, whereas public transportation data is shared in the General Transit Feed Specification (GTFS) (Google, Inc., 2006) format. Table 4.3 presents the average hourly and daily record number per data type (traffic and wind) for the Slovenian use case.

Table 4.3 — Average hourly and daily record number per data type for the Slovenian use case

Data Type	Average Daily Records	Average Hourly Records
Traffic Sensors	85.200	3.550
Wind Sensors	5.040	210
Total	86.240	3.760

The temporal range for traffic sensor and wind information data types span from January 2017 to November 2017. Traffic sensor records are collected every five minutes from a total of 355 sensors, with information for both directions (i.e., 710 data collection points) corresponding to more than ten million sensor readings, and wind conditions are captured every two minutes in seven major cities and highways, for a total of 30000 records. In the case of traffic events, the information is collected only when an event occurs, and the temporal range spans from January 2011 to November 2017, with a total of 6265 event records.

4.2.2 Pilot Study 2: Proactive Charging Schemes for Freight Transport

The second pilot study proposes a dynamic toll charging system for shadow-toll highways in Portugal (Figueiras, et al., 2019), supported by Big Data technologies, to induce changes in heavy freight vehicles' behaviour by diverting heavy vehicle traffic from urban and national roads to underused tolled highways. This is accomplished by attracting or discouraging the use of specific highways through toll prices' variability, according to the quality-of-service prediction on those highways and adjacent alternative roads. The system is fed by traffic flow conditions of both tolled highways and their national road alternatives, combining historical and real-time data collected from traffic sensors scattered throughout highways and alternative national roads, to calculate the toll pricing of highways in advance, depending on traffic congestion conditions on both road types.

The design and development of the dynamic toll pricing model considers the traffic flow now-casts (including traffic events, maintenance, accidents and weather-related situations) and traffic flow forecasts, resulting in more accurate predictions for highways and national roads. Since traffic data quantity and quality are crucial to the prediction of road networks' statuses, real-time and predictive Big Data analytics methods are used.

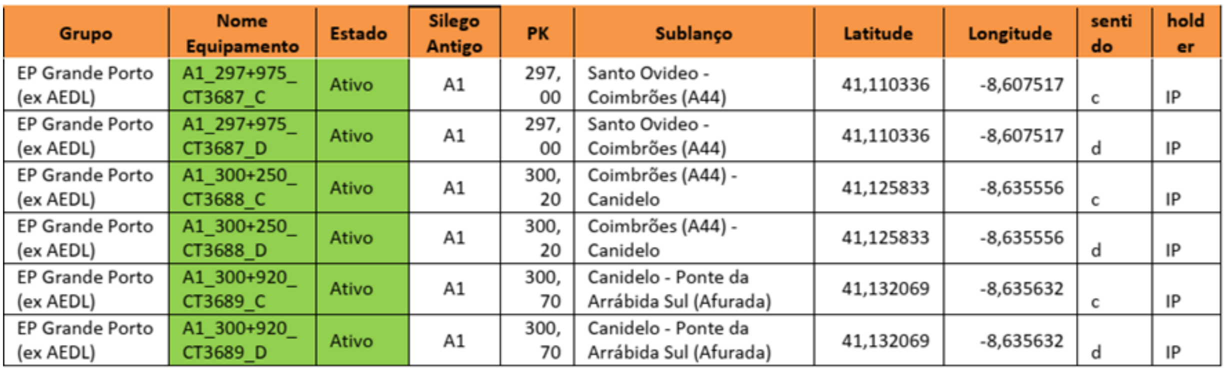
Hence, the dynamic toll charging system needs to be supported by the latest Big Data technologies to efficiently collect and process big amounts of traffic data, and swiftly perform traffic now-casts and forecasts to feed the dynamic charging model. Therefore, the main contributions of the work presented here can be highlighted as follows:

- Development of a Big Data infrastructure capable of collecting and processing large volumes of traffic data in real-time, and swiftly perform forecasting analytics to produce traffic predictions.
- A mathematical model for dynamic toll price calculation, which takes into account both the traffic in the tolled highway and in its toll-free, alternative roads.
- Integration of real-time toll pricing results of the dynamic toll charging system with logistics operator fleet management systems through dynamic toll information user interfaces.
- Test and validation of the system in a real-world scenario, targeting heavy freight vehicles.

The pilot focused on the highways and national road alternatives in the north of Portugal. For a more detailed view on this pilot, please refer to (Figueiras, et al., 2019).

4.2.2.1 Data Sources

Infraestruturas de Portugal (IP), the public road infrastructure operator, installed vehicle-counting sensors throughout their road network. For the Portuguese use case, which covers only the northern portion of Portugal's highways, IP selected 1127 active vehicle counting both in highways and their national road alternatives. The temporal range for counter data spans from January 2014 to December 2014. The metadata for the sensors is provided in an Excel spreadsheet, and a sample is presented in Figure 4.2. The metadata parameters are "Grupo", defining the highway group, "*Nome Equipamento*", the unique ID for the sensor, "*Estado*", the status of the equipment (e.g. active, not active), "*Silego Antigo*", the name of the road, "*PK*", the kilometre point, "*Sublanço*" the highway section name, "*Latitude*" and "*Longitude*", the latitude and longitude, expressed in the World Geodetic System (WGS) coordinate system, "*sentido*", the bearing or direction ("c": from lowest to highest road kilometre; "d": from highest to lowest kilometre) and "*holder*", the highway operator.



Grupo	Nome Equipamento	Estado	Silego Antigo	PK	Sublanço	Latitude	Longitude	sentido	holder
EP Grande Porto (ex AEDL)	A1_297+975_CT3687_C	Ativo	A1	297,00	Santo Ovideo - Coimbrões (A44)	41,110336	-8,607517	c	IP
EP Grande Porto (ex AEDL)	A1_297+975_CT3687_D	Ativo	A1	297,00	Santo Ovideo - Coimbrões (A44)	41,110336	-8,607517	d	IP
EP Grande Porto (ex AEDL)	A1_300+250_CT3688_C	Ativo	A1	300,20	Coimbrões (A44) - Canidelo	41,125833	-8,635556	c	IP
EP Grande Porto (ex AEDL)	A1_300+250_CT3688_D	Ativo	A1	300,20	Coimbrões (A44) - Canidelo	41,125833	-8,635556	d	IP
EP Grande Porto (ex AEDL)	A1_300+920_CT3689_C	Ativo	A1	300,70	Canidelo - Ponte da Arrábida Sul (Afurada)	41,132069	-8,635632	c	IP
EP Grande Porto (ex AEDL)	A1_300+920_CT3689_D	Ativo	A1	300,70	Canidelo - Ponte da Arrábida Sul (Afurada)	41,132069	-8,635632	d	IP

Figure 4.2 — Sample IP sensor metadata records

Counter data is delivered in two main ways: by Secure File Transfer Protocol (FTP), in CSV format (example and description in Appendix A.8, page 247) and via SQL dumps (example and description in Appendix A.9, page 249) Sensor data in CSV format only possess vehicle counts, while sensor SQL dumps contain vehicle counts, average speeds and highway occupancy percentages. Furthermore, IP granted access to proprietary electronic toll sensor data, spanning from October 2010 to February 2017, from 204 toll sensors for the tolled highways, which can record vehicle passages. The electronic toll data is property from the concession holders for each highway selected for the pilot, namely Via Livre and Ascendi. Examples for these data sources are presented in Appendixes A.10 (page251) and A.11 (page 252), respectively. All sensor readings have a sample rate of five minutes. These vehicle-counting and electronic toll sensors aggregate vehicle counts by vehicle class, since there are five classes of vehicles in Portugal:

- Class 1 vehicles are all motorcycles and vehicles with two axes that have the distance between the front axis and the road surface of less than 1,10 meters.
- Class 2 vehicles are all vehicles with two axes that have the distance between the front axis and the road surface of greater or equal than 1,10 meters.
- Class 3 vehicles are all vehicles with three axes.
- Class 4 vehicles are all vehicles with four or more axes.
- Class 5 is a special class for motorcycles that possess electronic toll charging systems (exclusive for electronic toll data).

Moreover, IP also provided traffic event data spanning from September 2010 to July 2016, both in batches, through SQL dumps, and real-time streams, through XML-based Web services, totalling at 9416 traffic events. Examples of both data types are represented in Appendixes A.12 (page 254) and A.13 (page 256), correspondingly.

4.3 Portugal 2020 Mobile Security Ticketing Project

Mobile Security Ticketing is a Portugal 2020-funded project (Portugal 2020, European Union, 2020) aiming to achieve an implementation of an alternative support for contactless ticketing based on Host Card Emulation (HCE) technology available in the latest smartphones, this solution is independent of Mobile Network Operators (MNO), but perfectly integrated in the existing infrastructure operators. This development enables the following innovations, which are all high added value: replacement of the ticket card with a Mobile application; use the smartphone to make the purchase and display ticket availability rather than the use of traditional sales channels; consultation of helpful information on the smartphone instead of dedicated panels available in some places.

Mobile Security Ticketing involves technological developments that must be achieved in two distinct areas: development of a safe mobile app and creation of an automatic detection system and event handling. For transport operators in general, the solution to develop will justify its introduction as a complementary system to the existing ticketing systems and an easy introduction due to the low impact on the ticketing infrastructure already installed even if they are provided by other integrators. The mobile app, which is the visible project component to the user, intends to be an innovative model of interaction with the passenger, consolidating on the Smartphone the purchase, security storage and integrated publication of information to the passenger. These features are now scattered on multiple channels and without the omnipresence offered by the permanent network connection of the smartphone. The georeferenced information collected automatically from the user community, will, in addition to providing the passenger's state of the network in real time, provide operators with additional data on the use of transport (source / destination) that they currently have difficulty in obtaining by the traditional methods.

Therefore, the mobile app is supported by data consolidation services, such as data exploration and inference mechanisms, that have the objective of obtaining insights about the public transport infrastructure in two different flavours: commuters' paths within the public transportation network and the network's performance status through online monitoring of network's sections. These services must be able to cope with various data sources, both real-time and batch, and are supported by Big Data technologies to perform ETL, data analytics and visualization tasks, to provide public transport operators with patterns and insights about their networks.

4.3.1 Data Sources

The association of public transport operators for the city of Lisbon, Portugal (OTLIS), maintains a close connection with almost all state-owned and private public transport operators, and stores all the monthly validation data for monthly profiles and single-use tickets from these operators. Profiles are monthly paid public transport subscriptions, often charged based on the social profile of the commuter (e.g., children, elderly, student, military, etc.) whereas single-use tickets are anonymously bought for single public transport rides. In the case of single-use tickets, no data about the user is gathered, while in profile-based smart cards, several data parameters about the user are stored, such as gender and age, residency postal code, etc. An example for OTLIS validation data, along with the respective parameters' description, can be found in Appendix A.14 (page 258). The data corresponds to the month of May 2018, corresponding to more than 55 million validation records.

4.4 Examples’ Overview and Cross-reference Table

This section provides an overview of each of the example use cases for the different projects and pilots presented in this chapter. Table 4.4 presents the cross-reference between the different use cases that will serve as examples in the following chapters, the project and pilot that supported their development and validation (marked with the symbol "✓") and the chapter in which the example will be overviewed (marked with the symbol "★"). For the OPTIMUM project, the pilot scenarios' countries are represented by their abbreviations (UK - United

Kingdom, SL - Slovenia, PT- Portugal). It is worth to note that all the references for each example are research works performed by the author of this thesis.

Table 4.4 — Example-Project-Chapter Cross-reference Table

Example	**Project / Scenario**					**Chapters**	
	OPTIMUM			Mobile Secure Ticketing	MobiS	Chapter 5	Chapter 6
	Pilot 1 (UK)	Pilot 1 (SL)	Pilot 2 (PT)				
Big Data harmonization pipeline (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018; Figueiras, et al., 2016; Figueiras, et al., 2016)	✓	✓	✓	✓		★	★
CEP for traffic event detection (Figueiras, Antunes, Guerreiro, Costa, & Jardim-Gonçalves, 2018; Antunes H. A., 2017)		✓				★	★
Real-time traffic flow analysis (Figueiras, et al., 2018; Rosa, 2017)		✓	✓			★	★
Public transport network status analysis and visualization (Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019)				✓		★	★
Twitter mining for traffic event detection (Gutiérrez, Figueiras, Oliveira, Costa, & Jardim-Gonçalves, 2015)	✓				✓		★

The rationale behind Table 4.4 is to not only contextualize the examples that will be used to elucidate and instantiate the guidelines and best practices provided to practitioners and researchers in the following chapters, but also to broaden the spectrum of opportunities enabled by the usage of the prescriptive methodology, guidelines and best practices presented in this thesis work, by providing practitioners and researchers with concrete examples that have specific contexts, use cases and data-driven objectives and use different methods, techniques and technologies, but all fit into one or more of the components presented in the logical components and data flows model in Chapter 3 (Figure 3.2). The references for the examples are all taken from research works realized by the author of this thesis.

The "Big Data harmonization pipeline" example (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018; Figueiras, et al., 2016; Figueiras, et al., 2016) proposes an architecture for a Big Data harmonization pipeline to extract, harmonize to standard formats and store efficiently traffic- and mobility-related data. The proposed architecture is able to deal with raw data in many formats and sizes and to address interoperability at the data level, enabling the development of additional added-value services for highways users. It will be overviewed throughout Chapter 5, as it is the main example for the design and development of harmonization, standardization and interoperability processes for MobiTrafficBD. It will also serve as an example for data visualization towards an initial exploratory analysis for large volumes of data.

The objective for "CEP for traffic event detection" example (Figueiras, Antunes, Guerreiro, Costa, & Jardim-Gonçalves, 2018; Antunes H. A., 2017) is to categorize and detect complex events based on a repository of road traffic data. Data is collected by road flow sensors placed along major roads and motorways. As an output of this study a traffic event detection application prototype was developed, using CEP techniques. This example will be used to promote guidelines and best practices regarding the use of data exploration methodology standards to guide the design and development of data cleaning tasks in Chapter 5, and data analytics of MobiTrafficBD, in Chapter 6.

The purpose of the "Real-time traffic flow analysis" example (Figueiras, et al., 2018; Rosa, 2017) is to study the existing mechanisms for treatment and management of large volumes of data, techniques of real time processing and visualization of data, and to implement an application that reunite these techniques to analyse traffic flow and congestion in real-time. It will serve to exemplify the use of data exploration methodology standards to guide the design and development of data cleaning tasks in Chapter 5 and the methods for real-time data stream processing, analytics and visualization in Chapter 6.

The challenge addressed by the "Public transport network status analysis and visualization" example (Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019) is to analyse the supply-demand trends of the transportation network of Lisbon's metropolitan area as well as the ability of Big Data technologies to cope with data collected from transport operators, by inferring automatically and continuously complex mobility patterns in the form of insightful indicators (such as connections, transhipments or pendular movements). This example will be used in Chapter 5, as a second example in which the

Big Data Harmonization pipeline was used to perform data harmonization tasks over MobiTrafficBD, this time for public transportation data, and in Chapter 6, to demonstrate the application of the proposed prescriptive methodology in the batch processing, analytics and visualization of big volumes of MobiTrafficBD.

Finally, the main objective of the "Twitter mining for traffic event detection" example (Gutiérrez, Figueiras, Oliveira, Costa, & Jardim-Gonçalves, 2015) is to detect traffic events from Twitter messages, called "tweets", by combining a mix of Natural Language Processing, classification, clustering, name-entity recognition and geolocation methods to not only detect a traffic event, but also to detect the type of event (e.g., traffic jam, accident, etc.) and to locate the spatial position of the event, with some degree of uncertainty, based on extracted information about roads, road sections or place names. This example constitutes a good example of using a sequential architecture for streaming ST Events (tweets can be considered a ST event, since they possess, along with the tweet message, spatiotemporal information in the form of the location at which the tweet was published and the timestamp for the publishing). This sequential approach enables a data-driven pipeline in which data streams are further evaluated and transformed in each step, making it a suitable example for sequential processing and analysis of streaming data.

5

Modelling, Collection and Harmonization of Heterogeneous MobiTrafficBD

Now that the overall prescriptive methodology, guidelines and best practices were described and the use cases that will be used as examples in this and the forthcoming chapter are presented, it is time to drill down into the specifics of the Data Management component of Figure 3.3. This chapter will focus on interoperability issues, since it is in the Data Management stage that interoperability is achieved using standards, harmonization methods and storage technologies. Particularly, data interoperability, which addresses "the ability of systems and services that create, exchange and consume data to have clear, shared expectations for the contents, context and meaning of that data" (Data Interoperability Standards Consortium, 2021), must be achieved in the Data Management stage to provide already fully interoperable data to the upcoming Data Processing, Data Analytics and Visual Analytics stages. These components are more than often use case-driven, meaning that the Big Data processing, analytics and visualization processes have a short range of application in other use cases, since they are too specific towards solving a concrete problem or challenge. Moreover, data harmonization and interoperability enable practitioners and researchers to apply *the same* processing, analytics and visualization methods to data coming from *different* data sources, captured from *different* technologies with *different* hardware and software specifications, and spanning *different* geographic locations and temporal ranges.

5.1 The use of Standards in Modelling Strategies

As already discussed in sub-section 2.2.3, the selection and application of standards, specifically data modelling and formatting standards, enable data interoperability by supporting data harmonization of heterogeneous data sources towards unified and widely used data models and formats. But data modelling and formatting standards are not the only standards that are crucial to build any data processing framework, and MobiTrafficBD frameworks are not an exception.

Other standard types that are useful for MobiTrafficBD frameworks are, for instance, data exploration and mining standard methodologies or, for a more general case, traffic- and mobility-related standards that are not directly linked to data, such as in the case of traffic control standards (e.g., road types, signalling, vehicle categories, etc.) or commuting and mobility standards (e.g., crosswalk distribution across the road network, public transit standards, etc.). This section will focus on the data modelling and formatting standards available for MobiTrafficBD, specifically DATEX II, and on data-driven methodologies for data exploration and understanding, which the author considers vital for data cleaning and harmonization tasks' design and development.

5.1.1 The Importance of Data Exploration Methodologies in Data Modelling & Harmonization

Before going through the actual data modelling standards for MobiTrafficBD, it is worth to highlight the importance of following a standard methodology for data exploration when designing and developing data harmonization, cleaning and storage processes, i.e., the processes comprised in the Data Management component of Figure 3.3. The design stage for data management processes starts from the elicitation of available data sources and business scenario requirements and research challenges and involves finding solutions for a panoply of issues related not only to the data being collected but also to the domain, field of study and context for which the overall solution is built.

From a business scenario perspective, popular data-driven issues are "what data sources are relevant for answering the business scenario's research challenges?", "from the relevant data sources, which data parameters and characteristics can be used to solve the business scenario's challenges?" or "are the relevant data sources enough, in terms of quality and quantity, to ensure the proper analysis and insight creation to support decision making tasks and to provided definitive answers to the research challenges?", for instance. From a data perspective, common issues comprise "what data types are available?" (e.g., traffic sensor data, traffic event data, public transport data, etc.), "can the data sources be clustered by similar data types?" (e.g., two or more sources represent traffic sensor data), "what is the overall data quality?" (e.g., traffic sensor data is not complete due to sensors' down times and failures), "are there any data sources already represented in one or more standardized formats?”, "are there any data modelling standards that are suitable for representing the available data sources?" or "what are the main characteristics of the available data and how can they be leveraged in order to better answer to the research challenges at hand?", just to name a few.

More than often, these questions are difficult to answer without a proper methodology to explore and understand both the data and the data-driven business scenarios, due to a panoply of reasons. First, there is an inherent difficulty to ensure ongoing and effective engagement of both business and ICT professionals throughout the project's lifetime, since data management tasks are highly iterative, and the issue of data management practitioners and researchers to drift away from business and other ICT professionals is a harsh reality. Because these tasks are not broadly understood or accessible, the business partners cannot participate.

Furthermore, data-centred projects come in a wide variety of flavours, from non-inferential business intelligence and data warehousing solutions to real-time Big Data analytics solutions, which present more exploratory and interactive scenarios. Hence, no two projects are alike in their data management processes. Lastly, variability across data-driven projects poses challenges for project managers, who need to hire suitable people and make time and cost estimates. Exploratory activities require expert data scientists and increase time and cost uncertainty, whereas data management activities require more data engineers and are more easily contained within a fixed time interval and budget.

There is not a single methodology that can be applied in every scenario, but a data exploration methodology would help to mitigate the above issues by, on one hand, giving the data management development team a framework within which each iteration must fit—if the iteration is not moving toward a better decision, then it is not helping (Taylor, 2018) — and, on the other hand, support project planning by clearly separating the various data management and exploration activities (Martínez-Plumed, et al., 2019). Although not specific to data harmonization and management processes, there are several data exploration methodologies available in the literature. Some examples are the older Knowledge Discovery in Databases (KDD) methodology (Fayyad, Piatetsky-Shapiro, & Smyth, 1996), the Sample, Explore, Modify, Model, Assess (SEMMA) methodology (Azevedo & Santos, 2008) and the well-known Cross-Industry Standard Process for Data Mining (CRISP-DM) methodology (Wirth & Hipp, 2000).

These methodologies were initially developed to guide IT researchers, practitioners and professionals through all stages of Data Mining and Analytics projects, throughout their design and implementation, from business-related tasks, such as requirements' elicitation, business understanding and mapping of business objectives, to data-related tasks, such as data exploration and understanding, selection, cleaning and harmonization, analysis, modelling and mining (Azevedo & Santos, 2008). Since then, and due to the evolution of data science and engineering, these methodologies have been also evolving to cope with the new paradigms for these fields (Martínez-Plumed, et al., 2019).

In several of the examples presented in Chapter 4, CRISP-DM was used to guide the several stages of Data Management processes' design and development (Figure 3.3), as proposed by the author in (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018; Figueiras, et al., 2016; Figueiras, et al., 2016) ("Big Data harmonization pipeline" example), (Figueiras, Antunes, Guerreiro, Costa, & Jardim-Gonçalves, 2018) ("CEP for traffic event detection" example) , (Figueiras, et al., 2018) ("Real-time traffic flow analysis" example) and (Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019) ("Public transport network status analysis and visualization" example). CRISP-DM was chosen solely because it is still one of the most commonly used Data Mining methodologies by both industry and academia (Martínez-Plumed, et al., 2019; Azevedo & Santos, 2008), and the choice for a specific methodology, such as the ones presented as examples or any other available, depends on the specificities of the scenario and data at hand.

In (Azevedo & Santos, 2008), authors draw a detailed parallel overview of KDD, SEMMA and CRISP-DM. In the case of data cleaning and harmonization tasks of the Data Management component in Figure 3.3, the goal is clear: to provide the upcoming sub-components of the logical components and data flows model of Figure 3.2 (Data Processing and Data Analytics) with clean, harmonized and interoperable data. Hence, CRISP-DM can be used as a guide in the design and development of these tasks.

5.1.1.1 CRISP-DM: Brief Overview

CRISP-DM (Wirth & Hipp, 2000) is based on a cyclic process with six stages: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation and Deployment, as shown in Figure 5.1. The Business Understanding stage focuses on exploring the project Challenges and requirements, define the business goals, converting them to a clear data exploration and mining problem definition, and create a preliminary plan to solve the proposed objectives. The Data Understanding stage comprises data collection, sampling, quality analysis and exploration tasks to familiarize with the data, discover initial insights, detect data quality issues and identify relevant subsets that enable hypotheses formulation and unravel hidden knowledge. The Data Preparation stage, according to the Data Mining perspective, corresponds to the preparation of the final dataset from the initial raw data. Data preparation tasks are likely to be performed multiple times, and not in any prescribed order. Tasks include table, record, and attribute selection, data cleaning, and construction of new attributes. But, from a data harmonization perspective, it also corresponds to the preparation of the data access tools and adapters.

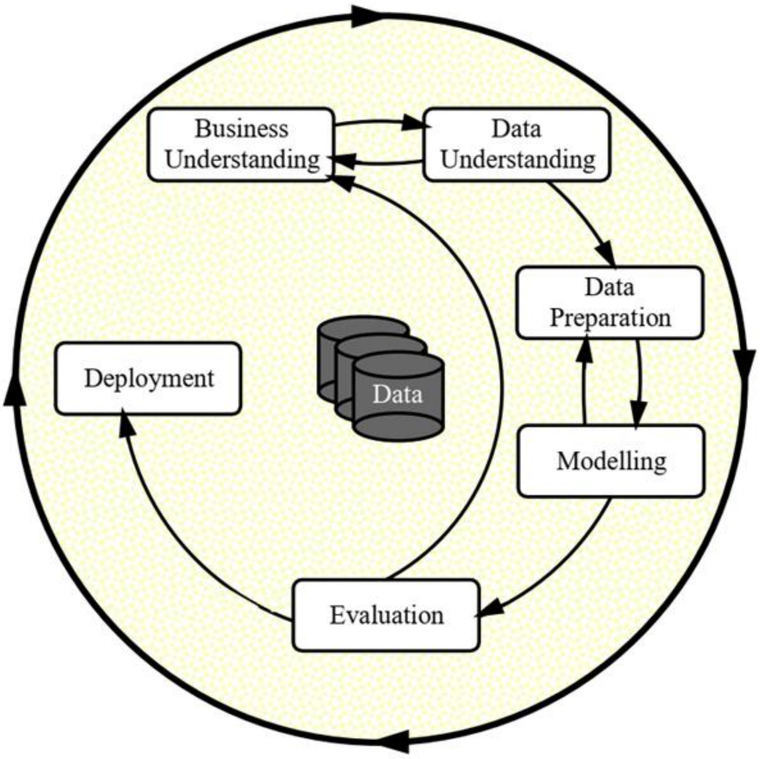


Figure 5.1 — The CRISP-DM cycle (Wirth & Hipp, 2000)

The Modelling stage corresponds to the Data Mining modelling tasks, in which analytics, mining and machine learning models are selected, applied and their parameters are calibrated to optimal values, to tackle the business challenges collected during the Business Understanding stage. This stage, from a data management perspective, defines the creation and transformation of data into the standardized, interoperable data models. The Evaluation stage, as the name entails, comprises all evaluation procedures to assess if the selected models are the most effective to tackle the aforementioned business challenges: if so, then the models will be deployed in the Deployment stage; otherwise, practitioners should return to the Business Understanding stage to understand what was wrong with the selected models. Finally, the Deployment stage is generally not the end of the project. More than often, the gained insights will need to be organized and presented in a way that final users can capitalize on. Depending on the requirements, the Deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process.

The following sections will present some examples of the application of the CRISP-DM methodology, or at least some of its steps, as support to the Data Management component (Figure 3.3) of the logical components and data flows model and the comprised harmonization and cleaning tasks.

5.1.1.2 CRISP-DM for Data Cleaning

The Business Understanding and Data Understanding stages of CRISP-DM can serve as the basis for data cleaning tasks, since the design and execution of these tasks need strong business and data knowledge, in the form of concrete business challenges, a clear picture of how the available data sources may provide solutions to such challenges, how the overall quality of the data at hand may pose issues to the envisaged solutions or which are the major data quality flaws in the data and how can they be mitigated. Further, the Data Preparation stage is also vital to the data cleaning phase, since it comprises the initial decisions around the data quality, namely which datasets have enough quality to be useful for the upcoming analysis tasks and which sets do not meet the required quality and must be discarded. Finally, the Modelling stage refers to the selection of data cleaning strategies and techniques that will mitigate the data quality issues, already explored in the Data Understanding stage, and their application to the data selected during the Data Preparation step.

The first instantiations of the above approach of using CRISP-DM to support data cleaning tasks are based on the "CEP for traffic event detection" example [284] and on the "Real-time traffic flow analysis" example (Figueiras, et al., 2018). In the first example, the business challenge was to develop a complex event processing system that could find anomalous events in traffic sensor data that could correspond to severe traffic events, such as accidents, traffic jams or other public events that would have an impact in the traffic, and correlate the detected events with known traffic events, already reported by RITMOs, and stored in a traffic event database. In the second example, the business challenge was to develop a real-time data collection and processing framework that could serve as a basis for the application of Data Analytics and Mining mechanisms over data streams to monitor and analyze traffic in real-time on Slovenian highways.

The data sources used in both examples were the Slovenian Traffic Sensor Data dataset (Appendix A.5, page 241), used in both examples, and the Slovenian Traffic Event Data dataset (Appendix A.6, page 243), used only in the CEP for traffic event detection, and the Data Understanding stage focused on the analysis of the data characteristics and quality of these datasets. The initial step was to do a preparatory data exploration process that described the data's structure, parameters and metadata of both data sources. Structurally, the traffic sensor data presents two different formats for different time intervals: from January 2016 to November 2016, a JSON schema that was based on an older XML schema was used, but the API's data format was updated to a more user-friendly JSON schema, which corresponds to the temporal period from December 2016 to May 2017. Despite the difference in formats, the data contents represented in both schemas are the same, although the data acquisition time intervals are also different for both schemas, with an interval between sensor readings of 5 minutes for the period between January and November of 2016, and of 10 minutes for the period between December 2016, and May 2017.

The second step was to perform a spatiotemporal coverage analysis. Firstly, a spatial dispersion analysis of both datasets was made, as presented in Figure 5.2. The spatial dispersion analysis shows that both datasets have a country-wide dispersion, although they are focused on the main road infrastructure arteries and cities of Slovenia, and that there are some good overlapping points between the two datasets, namely near the capital, Ljubljana (centre left cluster of points in the figure), the second biggest city, Maribor (upper right cluster of points in the figure) and some highway sections near the country's borders.

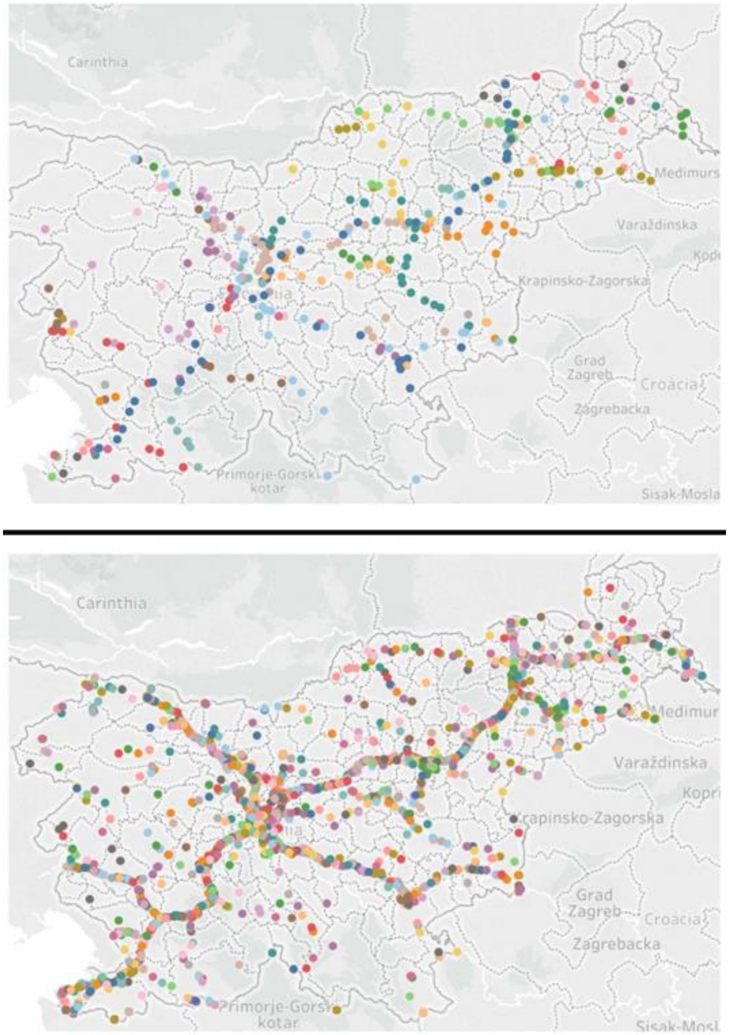


Figure 5.2 — Spatial dispersion analysis for the Slovenian Traffic Sensor dataset (top) and the Slovenian Traffic Event dataset (bottom)

Afterwards, a temporal coverage and availability analysis was performed. Temporal coverage represents the temporal range (i.e., minimum-maximum time limits) of the acquired data whereas temporal availability depicts a percentual relation between the available quantity of acquired data records and the expected number of acquired records, as represented in Equation 1.

$$Availability = \frac{\#\ of\ available\ data\ records}{\#\ of\ expected\ data\ records} * 100 \qquad (1)$$

Hence, for an interval between readings of 10 minutes, the expected number of readings for each sensor per hour is 6 readings, corresponds to 144 readings per day and 4.320 readings per month (in the case of the 5-minute interval, the values are doubled). If, for instance, a sensor has an average of 116 records per day for a given month, it means that its availability percentage equals 80%. Table 5.1 presents the monthly average data availability percentage for all 355 available sensors.

Table 5.1 — Monthly average data availability percentage for the Slovenian traffic sensor data

Month	Data Availability (%)	Time interval between sensor readings (minutes)
January 2016	96,2	5
February 2016	91,4	5
March 2016	0,25	5
April 2016	98,6	5
May 2016	99,2	5
June 2016	88,4	5
July 2016	88,8	5
August 2016	90,4	5
September 2016	92,7	5
October 2016	96,4	5
November 2016	26,4	5
December 2016	86,3	10
January 2017	97,2	10
February 2017	82,7	10
March 2017	86,4	10
April 2017	63,8	10
May 2017	18,4	10

In fact, in the Data Preparation stage, this percentage value was chosen as the quality threshold for sensor data: if the overall data availability percentage for one month is equal or higher than 80%, its data is cleaned and stored to be used by both examples; otherwise, that month's data is discarded. The same data selection exercise was then performed for individual sensors, as represented in Figure 5.3: the data availability percentage (y-axis) for a sample of nine sensors, represented by their unique IDs (x-axis), and for the first five months of the data's overall temporal range.

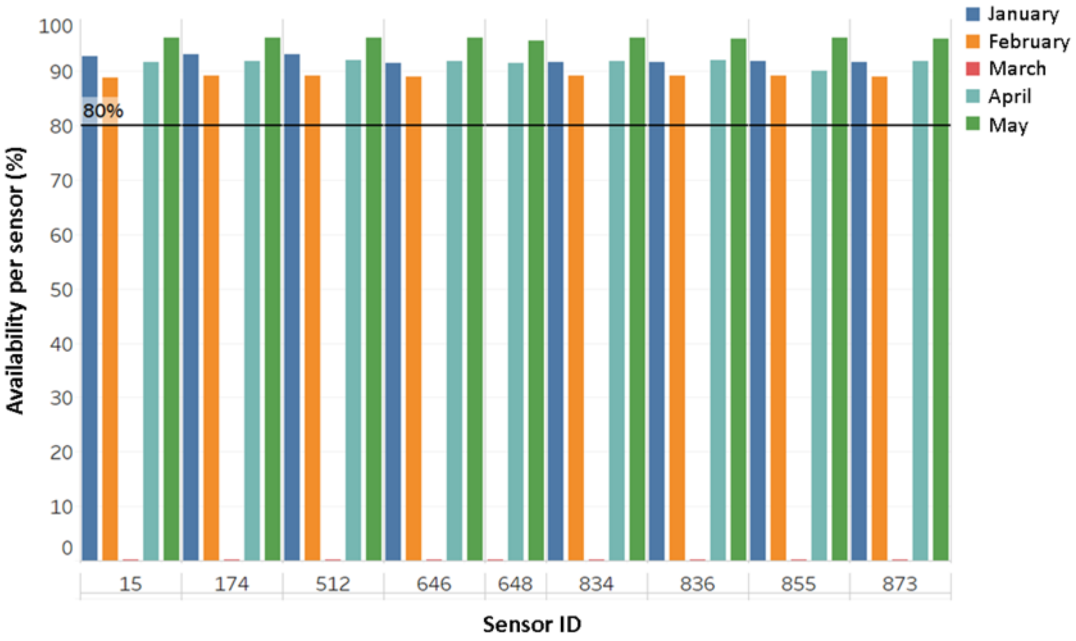


Figure 5.3 — Monthly availability for the first five months of 2016, for a random sample of nine sensors

There are clearly four months with availability percentages below 80% as represented in Table 5.1 and Figure 5.3 and were discarded: March and November of 2016 and April and May of 2017. In the case of March 2016, November 2016 and May 2017, the availability issue is that these months only have sensor readings for one (March 2016, as shown in Figure 5.3) or a few days only, which may mean that the sensor network was down due to maintenance or repairing tasks, energy shortage or communication problems between the network and the data acquisition server. For the case of April 2017, there are sensor readings for all sensors for every day of the month, but most of the days, there are only a few records acquired, usually between 21:00 and 23:00. Besides these availability issues, other issues were discovered during the Data Understanding stage and that were tackled through data selection in the Data Preparation stage and cleaning strategies in the Modelling stage:

- ***Duplicate readings***: There were some cases of duplicate readings, i.e., readings for the same sensor and the same date and time. Duplicate readings were removed.

- ***Mismatch between time gap between vehicles and road occupancy***: There are some cases in which the time gap between vehicles (*steci_gap*) and the road vehicle count (*stevci_stev*) do not match, as for instance, if the time gap between vehicles is 999 seconds and the vehicle count for the same reading is 24 vehicles, which is not possible. In this case, the adopted strategy was to divide the time interval between readings by the vehicle count, giving an equal average time gap between all vehicles for that reading.

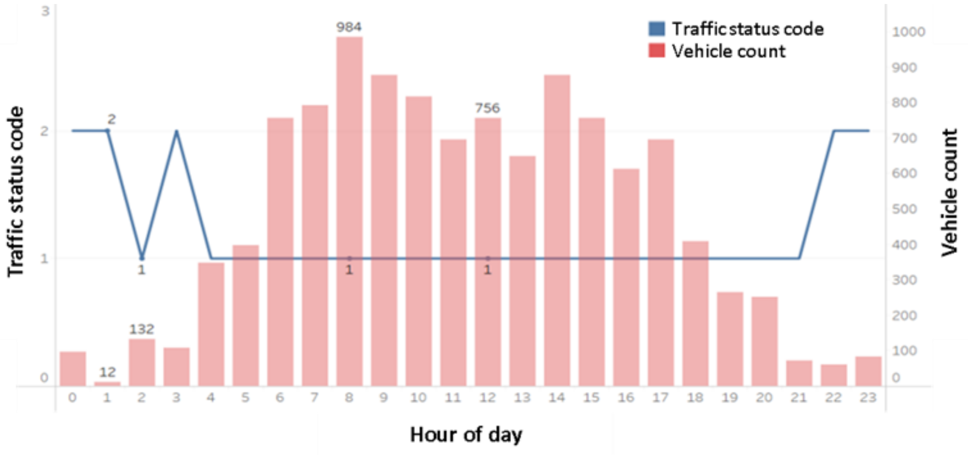


Figure 5.4 — Hourly correlation between the traffic status code and the actual vehicle count, for the 27th of April, 2017

- ***Mismatch between traffic status code and road occupancy:*** the traffic status code integer parameter (*stevci_stat*), which goes from 1 (lowest traffic concentration) to 6 (highest traffic concentration), should be directly correlated with the vehicle count parameter. However, some readings do not comply with this rule. For instance, Figure 5.4 represents the hourly data correlation between the status code and the vehicle count for a randomly chosen day and for a single sensor, in this case the 27th of April 2017. At 1:00 o'clock the status code equals 2 and the vehicle count equals 12, whereas at 2:00 o'clock the status code is 1 but the vehicle count equals 132, which is a breach of the above rule. The decision was not to rely on the status code parameter for the subsequent analyses, discarding it.
- M***ismatch between average speed and road occupancy:*** There are anomalous cases, such as an average speed of 255 kilometres per hour for one sensor reading i.e., for a period of 10 minutes, or average speeds of 2 kilometres per hour for a road occupancy of 50 vehicles in 10 minutes. These cases were treated as outliers.

Finally, the Data Selection stage comprised the selection of the data to be used in both the examples, which will be overviewed in Chapter 6, and what data to discard, and the

Modelling stage corresponded to the creation and application of data cleaning strategies, such as the ones already cited. As mentioned, the data cleaning processes are not the focus of this work and so, this subject will not be further discussed. For a thorough guide and best practices on data cleaning, the author recommends (Osborne, 2013).

This example illustrates the value of the application of a data exploration methodology, in this case CRISP-DM, on the design and implementation of data cleaning tasks, by providing a guiding framework with concise stages for business and data understanding, data selection and cleaning, and driven by both the available data and business challenges. Furthermore, the methodology stages are also well defined, enabling a better understanding of the role of each stakeholder in the process, such as RITMOs on the business side and data scientists and researchers on the data side, with clear bridges between both: data sources and business challenges.

5.1.1.3 CRISP-DM for Data Harmonization

As in the case of data cleaning tasks, a data exploration methodology like CRISP-DM can be particularly useful, especially when considering a wide set of data sources that must be harmonized. Harmonization processes for geographically scattered, multi-schema data sources are not easy to design and develop since, due to the difference between formats, data parameters and even spatial and temporal representations of these parameters, it is not trivial to find a common ground for the creation of a single harmonized schema that can encompass all the available data sources for the same data type (e.g., traffic sensor data).

Therefore, the application of the CRISP-DM stages to data harmonization may be important to better understand the data at hand, its commonalities and differences, whether structural or parametric, and to choose a data schema or model that suits these differences and commonalities in a single package. Moreover, the exercise of transforming data into a harmonized format is not straight-forward, as it entails several data comprehension, matching and filtering processes.

For the "Big Data harmonization pipeline" example (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018; Figueiras, et al., 2016; Figueiras, et al., 2016), there was the need to explore all the project's data sources and try to find the so said common ground between them. The example will focus on traffic sensor and event data, but other data sources were also submitted to the same process, such as car parking and bike sharing data sources. The first step of the Data Understanding stage was to perform a thorough report on the available data sources, catalogued in terms of pilot or provided by third parties (DTC: Dynamic Toll Charging pilot; MMR: Proactive Improvement of Transport systems pilot; TPP: Third party-provided) and country (POR: Portugal; UK: United Kingdom; SLO: Slovenia), by their name tags, as presented in Table 5.2.

Table 5.2 — Data sources list, catalogued by name tag

Data Source	ID	Description
DTC_POR_ConcessionToll-CrossingVolumes	1	Toll crossing volume of vehicles for each concession
DTC_POR_HighwayCounters	2	Highway counter data (5-minute granularity)
DTC_POR_RoadVolume	3	Counter Data on National Roads (5-minute granularity)
DTC_POR_TrafficEventsDB	5	Historical traffic events database
DTC_POR_TrafficEventsWS	43	Real-time Traffic events Web Service
MMR_UK_AverageSpeeds	9	Average vehicle speeds from approximately 963 locations within the West Midlands conurbation
MMR_UK_Congestion	10	Congestion levels at 98 locations within the West Midlands conurbation
MMR_UK_Flows	11	Traffic flows from 743 loop detectors within the West Midlands conurbation
MMR_UK_JourneyTimes	12	Travel times data from Automatic Number Plate Recognition (ANPR) cameras located at 237 locations within the West Midlands conurbation
MMR_UK_Occupancy	13	Occupancy levels from 82 loop detectors within the West Midlands conurbation
MMR_UK_TravelTime	14	Travel times data from 437 locations within the West Midlands conurbation
TPP_UK_TrafficEvents_INRIX	34	Feed that provides snapshot of all vehicles within a fleet –only available with key provided by British Traffic
TPP_UK_MIDASTrafficCounts	44	NTIS collects traffic data from Highways Agency's MIDAS Gold servers every minute. MIDAS Gold data including speed, flows, occupancy and headway is reported on a per lane basis where the site is configured for counting

TPP_UK_ANPRJourneyTimes	45	Travel data from Automatic Number Plate Recognition (ANPR) cameras located at strategic locations on the network
TPP_SLO_LoopSensorsFeed	38	Sensor feed from Slovenian roads –including average speed, number of vehicles, gap between vehicles, occupancy, etc.
TPP_SLO_TrafficEvents	41	Slovenia, coverage of current road traffic events (roadworks, accidents, traffic jams etc.)

Values in the "ID" columns are not sequential since the list had more data sources, and other data sources were added later to the project, but the example will only focus on the ones in the list, which correspond to the example data sources already presented in Chapter 4 and in the Appendixes. Two examples of the data source exploration reports are presented in Appendix A.15 (traffic sensor data example, page 260) and Appendix A.16 (traffic event data example, page 262). After the thorough exploration reports, the next step was to analyse the commonalities, structure- and parameter-wise, between the data sources described in Table 5.2. The result of this analysis in Appendix A.17 (page 264).

The table in Appendix A.17 (page 264) is divided into three data structures: road sensor metadata, road sensor values and traffic events. The different data sources of Table 5.2 were distributed between each of these structures according to their data types (road sensor data and traffic event data) and road sensor data was divided into two linked structures: road sensor metadata, containing all information of each traffic sensor and common for all data readings, such as the sensor's unique ID, location and bearing, and road sensor values, containing the sensors' captured values along with the capture timestamp. Additionally, the different data sources IDs were mapped to common parameters, shared by one or more data sources, corresponding to the "Data Sources" column of Appendix A.17.

This analysis, equivalent to the Data Preparation stage of CRISP-DM, was fundamental to not only select the appropriate harmonized schema but also to identify how raw data parameters could be mapped to a common, harmonized schema, as will be presented in the next section, which will cover the schema selection and the Modelling stage.

5.1.2 Data Modelling & Formatting Standards for MobiTrafficBD: DATEX II & Others

As already pointed out in Chapter 2, data modelling standards for ITS, such as DATEX II for traffic-related data or GTFS for public transportation-related data, are crucial to promote data interoperability on ITS. But why is data interoperability important for ITS? The main answer to this question is the fact that if all data were to be represented and exchanged in interoperable, standardized formats, there would be no need for the application of harmonization and cleaning processes, as these require a certain degree of processing power and time. Further, as in any digitalization venture, implementing data harmonization processes have associated costs. In an era in which real-time access to data is becoming more important, both for delivering quality data to users on time (e.g., public transport schedules, delays and disruptive events in the network) and for effective application in decision-making processes (e.g., taking actions to optimize the road infrastructure), the less time and computational power are consumed, the better.

This means that if the harmonization and cleaning processes would be bypassed, then data would flow directly to processing and analytics processes, which would involve shorter data analysis times and, consequently, quicker responses, in the form of new insights and knowledge, to the users/stakeholders and, ultimately, would translate in financial savings for RITMOs, whether direct, by saving in digitalization processes, or indirect, by enabling decisions that would lead to infrastructure and network cost reductions. Hence, the application of data modelling and formatting standards should be generalized and performed directly upon data collection, whether at the edge (i.e., at the devices that capture the data) or soon after (i.e., before the initial storage of raw data after its collection). Since this "interoperability-by-design" concept is still far from generalization, there is the need for cleaning and harmonization processes that drive interoperability of heterogeneous data sources forward.

Picking up on the "Big Data harmonization pipeline" example in the previous section and, particularly, the table of Appendix A.17 (page 264) that ensued the analysis of the data sources, the next step was the Modelling stage. In this stage, the parameters for each data type (traffic sensor metadata, traffic sensor value and traffic event), presented in Appendix A.17, were mapped to DATEX II standard model's parameters, whenever there was a direct match between both the raw data parameter and the DATEX II parameter, in terms of meaning (i.e., the parameters have the same meaning and context), representation (i.e., the raw parameter can be represented by the standard parameter as, for instance, in the case of common enumerations) and type (i.e., the data types of both parameters are the same, e.g. numeric, or the raw data parameter's type can be transformed to the standard parameter's type, e.g. string representing a number transformed to a numeric type). The result of this mapping is presented in Appendix A.18 (page 267).

For the cases in which no match was found for a specific raw data parameter, and depending on the relevance of the parameter, it was either discarded or proposed for a DATEX II "Level B" extension. For instance, when the parameter was an enumeration retrieved from an actual numeric measurement, as in the case of trends and status flags (e.g. "flowTrend" or "flowStatus"), the parameter was discarded, since it could be reproduced through the numeric measurement; In cases in which the parameter was relevant for future analyses, but did not matched any of the parameters in the standard model, such as in the case of the vehicle count for the five Portuguese vehicle classes, a "Level B" extension was proposed, as shown in Appendix A.18 (page 267).

The next step was to build the proper harmonized schema and harmonize the data according to it. In this example, a JSON document-based database system (MongoDB (MongoDB, Inc., 2015)) was used due to the characteristics of the data sources, as will be explained in sub-section 5.2.3. For now, it suffices to mention that the choice had primarily to do with data unevenness and heterogeneity between data sources corresponding to the same data type, as in the case of traffic sensors that capture different measurements (e.g., traffic flow, occupancy, average speed) in different spatial granularities (e.g., entire road, per lane), and the spatiotemporal support provided by the selected system. Document-based systems, in opposition to RDBMS, enable schema flexibility, meaning that some data records (documents) may have values for all the parameters in the schema while other records may only have values for some parameters.

Hence, the selected system, besides providing spatiotemporal support, should enable storage of a JSON schema that could encompass different parameters and granularities for the same data type and should be compliant and easily mapped to the DATEX II standard model. At the time of development of this example, the JSON-based DATEX II Light specification, already overviewed in Chapter 2, was not yet available and so, this intermediate mapping step between the harmonized JSON-based schema and the XML-based DATEX II model was a necessity.

Table 5.3 presents the most relevant DATEX II parameters, their description and the mapped JSON parameters, when applicable, to transform the JSON sample in the DATEX II model-based format of Appendix A.19 (page 271). If the mapping of parameters is not applicable, these values are often automatically added to the DATEX II standardized output data (e.g., in the case of the "confidentiality" parameter).

The JSON-based schemas for traffic sensor metadata and readings and for traffic events are presented in Appendix A.19 (traffic sensor metadata, page 271), Appendix A.20 (traffic sensor reading, page 274) and Appendix A.21 (traffic event, page 277), along with the corresponding DATEX II mapping for each data type. In Appendix A.19 (page 271), the JSON-based schema is the original MongoDB database record and is composed by all parameters that can be mapped (directly or via extensions) to the DATEX II model. XML parameters marked with

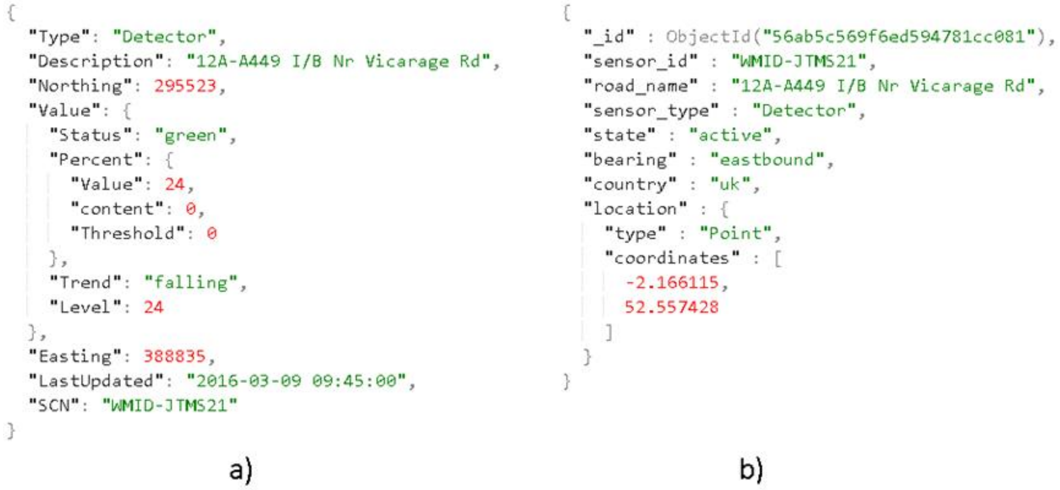
"*********" correspond to the vehicle classes in Portugal and are represented by the extensions to the "Level A" model.

Table 5.3 — DATEX II traffic sensor metadata's (Appendix A.19) most relevant parameters, descriptions and corresponding mapped JSON-based parameters

DATEX II XML Parameter	Description	Mapped JSON Parameter
country	Country of origin	country
nationalIdentifier	National authority identifier	N.A.
publicationTime	Time of publication	N.A.
confidentiality	Confidentiality level	N.A.
informationStatus	Information veracity status	N.A.
measurementSiteTable "id"	Unique ID for the measurement sites list (all sensors)	N.A.
measurementSiteTableReference	Unique name for the measurement sites list (all sensors)	N.A.
measurementSiteRecord "id"	Unique ID for the measurement site, comprised of country, concession holder and sensor ID	country, concession_holder, sensor_id
measurementEquipmentReference	Unique name for the site measurement	sensor_id
measurementSiteIdentification	Unique ID that serves as site identification	_id
measurementSiteName	Name for the measurement site	section
measurementEquipmentTypeUsed	Type of the measurement equipment	sensor_type
measurementSide	Side of the road (bearing) for the measurement site	bearing
period	Time period between measurements	N.A.

specificMeasurementValueType	Measurement type (TrafficFlow, Occupancy, Headway, Speed, Volume), extracted from the available measurements for the sensor	N.A.
vehicleType	Measured vehicle type (All, heavy, light, vehicle classes), extracted from the available measurements for the sensor	N.A.
latitude	Latitude expressed in WGS	coordinates [0]
longitude	Longitude expressed in WGS	coordinates [1]

Traffic sensor metadata sets for all pilots (UK, Slovenia and Portugal) were harmonized to the JSON-based MongoDB schema shown in Appendix A.19. As an example of the required transformations from raw sensor metadata to harmonized sensor metadata, Figure 5.5 presents the harmonized format for the raw data record in Figure 5.5 a).



```
{
  "Type": "Detector",
  "Description": "12A-A449 I/B Nr Vicarage Rd",
  "Northing": 295523,
  "Value": {
    "Status": "green",
    "Percent": {
      "Value": 24,
      "content": 0,
      "Threshold": 0
    },
    "Trend": "falling",
    "Level": 24
  },
  "Easting": 388835,
  "LastUpdated": "2016-03-09 09:45:00",
  "SCN": "WMID-JTMS21"
}
```

a)

```
{
  "_id" : ObjectId("56ab5c569f6ed594781cc081"),
  "sensor_id" : "WMID-JTMS21",
  "road_name" : "12A-A449 I/B Nr Vicarage Rd",
  "sensor_type" : "Detector",
  "state" : "active",
  "bearing" : "eastbound",
  "country" : "uk",
  "location" : {
    "type" : "Point",
    "coordinates" : [
      -2.166115,
      52.557428
    ]
  }
}
```

b)

Figure 5.5 — Traffic sensor metadata harmonization example from raw data (a) to harmonized data (b)

Table 5.4 presents the transformation rules applied to the raw data format (Figure 5.5 a)) to transform it into the harmonized format (Figure 5.5 b)). This short example is just a sample of the transformation rules for all data sources and data types. Some of the transformations are as simple as changing the name of the parameter and maintaining the original value, while other transformations are more complex, such as in the case of dates or locations. The more complex cases can be divided into five rules:

- For date parameters, the date format of the original raw parameter is required beforehand.

- For locations, the coordinates may need to be translated in terms of their reference system. Other option is to extract locations from a location list, which is independent of the raw data sources.
- In the case of DATEX II predefined enumerations, such as in the case of measurement types (flow, occupancy, average speed, etc.) or traffic event types (accident, traffic jam, roadworks, etc.), there must be a mapping between the raw data parameters' values to the corresponding enumeration.
- Other transformations are related to raw data parameters' types (character string, numeric, etc.). In these cases, a type-based transformation is required, as for instance from string-typed parameters representing numbers to numeric types.
- In some cases, further information may be necessary to complete the transformation. for instance, some information may be appended to one parameter, as in the case of road lane numbers being represented in the ID parameter of the measurement device, or in the case of needing other data sources to complement the raw data (refer to the "bearing" parameter in Table 5.4).

These transformation rules must consider the selected standard model to ease the process of outputting the harmonized data via the standard model, in this case DATEX II. This means that, if the harmonized parameters' value is already harmonized according to the model, whether is a date, a number, an enumeration or a location, the process of transforming the JSON-based harmonized format to the DATEX II model is significantly simplified. The major concern then is to have a complete map of JSON-based parameters' names to the corresponding DATEX II XML tags, as presented in Table 5.3. With both the transformation rules and the parameters map, it is easy to build a semi-automatic algorithm that performs the transformation process for all data sources, as will be presented in Section 5.2.

Thus, a thorough example on the use of data modelling and formatting standards is provided in this section, along with the description of the necessary processes to harmonize raw data sources and output them in standardized formats. For a complete guide of the harmonization processes for the examples in Appendixes A.20 (page 274) and A.21 (page 277) and other, please refer to the OPTIMUM project's deliverables (OPTIMUM Consortium, 2016; OPTIMUM Consortium, 2018) and to (Figueiras, et al., 2016).

Section 5.2 will focus on the design and development of a generic, semi-supervised system that performs the collection, harmonization, storage and standardized data sharing processes described in previous sections and following the guidelines provided in Chapter 3 and in the end of this chapter.

Table 5.4 — Applied transformations' list for the harmonization of raw traffic sensor metadata from the UK

Original parameter	Harmonized parameter	Transformation
SCN	**sensor_id**	The **SCN** parameter changes its name to **sensor_id**.
Description	**road_name**	The **Description** parameter changes its name to **road_name.**
Type	**sensor_type**	The **Type** parameter changes its name to **sensor_type.**
Status	**state**	The **Status** parameter changes its name to **state.** Mapping for the enumerations takes place as part of the transformation process. For instance, 'green' from the original parameter is mapped to 'active' for the harmonised parameter.
N/A	**bearing**	The **bearing** parameter is not available in the raw message. Further processing is required based on additional information (some sensors have been tagged in Open Street Map (OSM) files, while others required visual inspection) to define this. This parameter is set to 'undefined' and is later updated following additional processing.
Northing, Easting	**location**	The **Northing, Easting** values are to the **location** parameter (latitude and longitude). The original parameter is using the British National Grid reference system and therefore it is converted to WGS84.

5.2 Data Collection, Harmonization and Storage from Heterogeneous Sources

To bring all the aforementioned standards and examples into one system that is able to perform data collection, harmonization and storage of heterogeneous data sources, some requirements have to be met:

- Data Volume/Speed: The system must be able to cope with voluminous and fast data equally efficiently.
- Data Heterogeneity: The system must be able to tap into almost any data source, independent of the communication medium used.

- All-in-one: The system should enable easy reconfiguration and generic reuse across different scenarios, data sources and harmonized models and formats.
- Data Sharing: The system should provide mechanisms for harmonized data sharing, to promote data interoperability.
- Data Understanding: The system should promote data understanding by providing visual reports about the data being harmonized.

Data harmonization works in the literature are normally based on a few data types and data sources and on the harmonization of data into a single standard. This section proposes the design, development and deployment of a generic Big Data harmonization pipeline that can be easily repurposed and reused to cope with different data types and data sources, and harmonize data into any custom or standardized format, by enabling the insertion of new formats through their description in a meta data format. This harmonization pipeline is also able to support a better understanding about the data, by providing initial insights and statistics about the data being harmonized, in real-time. Further, it enables data sharing through the inclusion of custom APIs that allow for other researchers, practitioners or RITMOs to access fully harmonized data.

5.2.1 Big Data Harmonization Pipeline

The proposed Big Data harmonization pipeline's conceptual and technological architecture is represented in Figure 5.6.

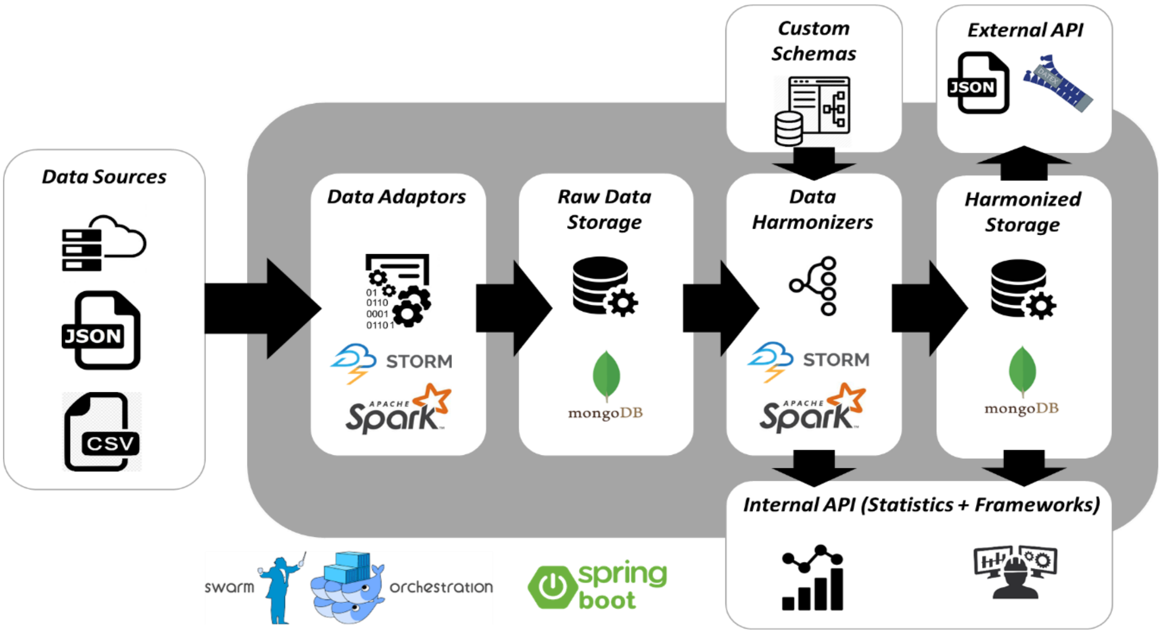


Figure 5.6 — Big Data harmonization pipeline's conceptual architecture

The pipeline comprises the data adaptors that will collect data from data sources, two databases, the raw data database and the harmonized database, the data harmonizers, algorithms that implement the required transformation rules to map raw data parameters to the harmonized model, and APIs to share the harmonized data, whether internally, if the pipeline is part of a bigger framework, as proposed in the logical components and data flows model of Figure 3.2, or is used, for instance, to produce visual reports about the harmonized data, or externally, through the provision of web services to share the harmonized data in standard or custom formats with external platforms, serving also as a harmonization pipeline that can be used by third-party entities to standardize their data sets.

Data adaptors collect data from a panoply of data sources, whether they are provided via files, web services, publish-subscribe mechanisms or database dumps, and store the raw data into the raw data repository. Data harmonizers get the raw data form the repository and transform it into harmonized formats, saving the newly harmonized data into the harmonized database. The Big Data harmonization pipeline relies on Apache Spark (The Apache Software Foundation, 2018) for batch processing, taking advantage of Spark's libraries for data acquisition from files and distributed processing power, and on Apache Storm (The Apache Software Foundation, 2018) for stream processing, relying on Storm's streaming capabilities and easy connection to publish-subscribe mechanisms and streaming databases. The data repositories are built with MongoDB (MongoDB, Inc., 2015), which adopts a NoSQL paradigm for non-structured and scalable storage. The main platform supporting the development of algorithms and APIs is Spring Boot (VMware, Inc., 2020) platform and the pipeline has a modular and distributed architecture, enabling parallel distribution of processing and storage tasks via Docker's containerization and orchestration tools (Docker, Inc., 2013).

The Big Data harmonization pipeline was reused in several projects, such as in the case of the "Public transport network status analysis and visualization" example (Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019), and some of these projects outside the Mobility and Traffic domain (e.g., Horizon 2020-funded project BOOST 4.0 (European Commission, 2018)). For user interaction, the pipeline was coupled with a Web application, that will be the subject of sub-section 5.2.2.

5.2.2 Big Data Harmonization Web Application

To ease the necessary human interaction and supervision on the collection and harmonization process of third-party data, a specific Web application was built, instantiating the conceptual architecture of Figure 5.6. The main objective of this application is to facilitate the upload, harmonization and export of MobiTrafficBD, supported by the methodological approach introduced in this work, and using the harmonized formats presented in previous sections.

In its first version the Web application handled batches of traffic sensor metadata and data, traffic events and weather data. The latter will not be overviewed in this work, but it is considered a GRTS and so, it is also harmonizable using the pipeline. Following versions added the ability to add new custom schemas and the ability to handle data streams. The application is composed by a collection of views, each of which is a direct interface to one of the processes in the Big Data harmonization pipeline. Hence, the Web application contains four views: Data Collection, Data Harmonization, Data Visualization and Data Export. These views are thoroughly overviewed in the following sections.

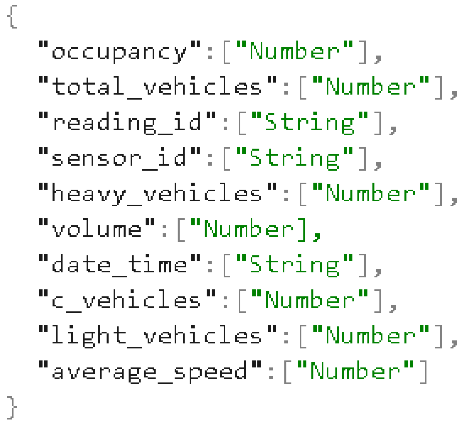
5.2.2.1 Data Collection

The data collection view enables users to upload files or configure the connection to Web service- and pub-sub mechanism-provided data, while choosing the type of data being uploaded. The permitted extensions for both cases are JSON, CSV or XML. In the case of Web services and pub-sub mechanisms, the interface provides input fields for the configuration of the services, such as the Web service method (e.g., GET, POST), URL for the services and any additional parameters required to access both Web services and pub-sub mechanisms. The view is presented in View 1(Appendix A.22, page 279) and View 2 (Appendix A.22, page 279), Appendix A.22, and is built to be intuitive: tabs to choose between File and Web service upload, limited number of input fields, usually with built-in options (without the need for textual input). In View 1, Appendix A.22 (Appendix A.22, page 279), the interface presents the File upload tab, which is composed by the "Select Data Type" and the "Browse Files" fields. The first field is a dropdown with the available data types (sensor metadata, sensor readings, traffic events, etc.), and the latter is a special file upload input, with capacity for four files with up to 4GB in size, each. Also, the validation of file sizes and extensions is handled by the input. The Web Service tab, shown in View 2, has four fields: the "Select Data Type" field, as in the File upload tab, the "Select Method" field, which lets the user choose between Web service methods, and the "Your URL" fields, which include the URL field and the additional parameters field, for both Web services and pub-sub mechanisms, such as Apache Kafka.

The backend collection process can be decomposed in two sub-processes: Upload and Storage. In the Upload phase, files are stored into a temporary server-side location, even in the case of web services, in which the input is converted into a file and stored. This enables modularity within the processing tasks, which is important in the case of adding new custom data parsers or data schemas, as explained later in this section. This process is bypassed in the case of streaming data to speed up the stream processing tasks. In the Storage phase, each file format (e.g., JSON, CSV) has its specific parser, each of which implements a single common interface, with a single procedure. Again, this ensures that new parsers may be added later, for more complex data formats (e.g., JSON arrays, multiple relational tables, etc.), adding modularity and flexibility to the solution.

After the storage process is done, another task extracts the schema from the stored raw data, with parameter name and type (e.g., number, string, date, etc.), which will be used in the harmonization step, for schema parameter matching between the newly retrieved schema and the chosen harmonized data schema. An example the retrieved raw data schema for a traffic sensor reading is shown in Figure 5.7.

The parameter type is represented as an array because sometimes the same parameter may be represented in different types depending on, for instance, the version and date of the files or Web services, since these data sources may change the format in which they share data, as shown in sub-section 5.1.1.3.



```
{
  "occupancy":["Number"],
  "total_vehicles":["Number"],
  "reading_id":["String"],
  "sensor_id":["String"],
  "heavy_vehicles":["Number"],
  "volume":["Number],
  "date_time":["String"],
  "c_vehicles":["Number"],
  "light_vehicles":["Number"],
  "average_speed":["Number"]
}
```

Figure 5.7 — Traffic sensor reading's raw data schema

5.2.2.2 Data Harmonization

The Data Harmonization view (View 3, Appendix A.22, page 280) is the next step of the Big Data harmonization pipeline instantiation, by enabling users to perform parameter and data type matches between the schemas for uploaded raw data and the harmonized schemas. Data sources coming from different pilots have different measurements or parameters. For instance, the Portuguese traffic sensor data has the occupancy percentage parameter, whereas the British sensor data has traffic headway parameter. These parameters only exist in their own country's sensor data. The following data structures are prepared to be dynamic in that sense, allowing for the presence of all the relevant parameters contained in all the data sources in the project related to the same data type.

The view enables users to map raw data's parameters to harmonized schemas' parameters, by creating connections between the tables on both sides. On the left side the Raw Format represents the raw data format, and on the right-side, Harmonized Format is the harmonized schema for the data type selected on the Collection View. Depending on the connected parameters' data types, the connection options and dependencies are shown in the box below the schemas. An example for the options of an Array object connection is shown in the view. In the example, the user connected CLASS1 (flow reading for category 1 vehicles) to the Array box, and so the following options in the Connection Settings box are presented:

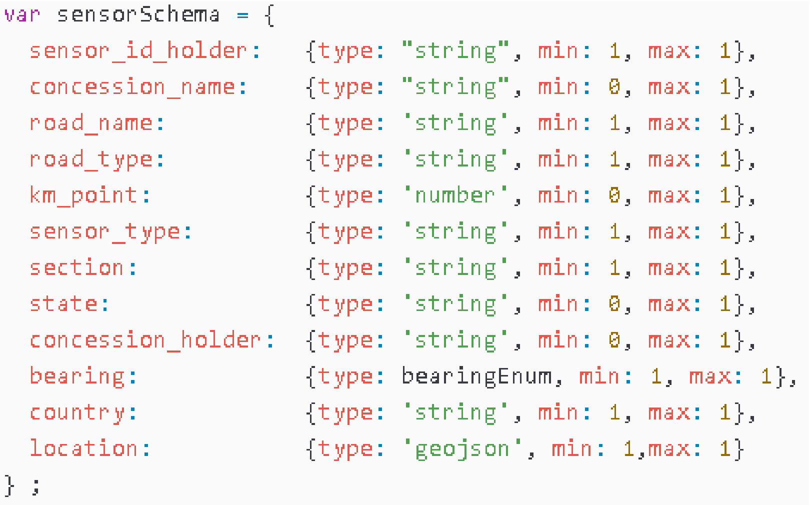
- Select Reading Type: select the sensor reading type (e.g., flow, speed, occupancy, etc.).
- Select Lane: select the road lane(s) from which the reading was extracted (optional).
- Select Vehicle Class: select the vehicle class, if any. Also valid for heavy and light vehicle categories.

The options are only shown for special data types (e.g., date, GeoJSON coordinates, arrays and enumerations). So, for instance, the connection to a date parameter would show the option Select Date Format, so that the user could input the ISO 8601 date format that would enable the conversion to date. For simple data types' conversions (e.g., number to number, string to number, etc.), there is no need for any additional information for the conversion. Table 5.5 shows an overview of these options.

Table 5.5 — Special transformations' input options

Schema	Key name	Input needed
Readings Schema	datetime	**Date Format**: ex. (ddMMyyyy'T'HHmmssZ).
	reading	**Reading Type**: flow, occupancy, volume, speed, headway. **Lane Number**: all or lane number. **Vehicle Class**: all, light, heavy, Portuguese Classes 1, 2, 3, 4 and 5.
Sensor Schema	location	**Latitude** and **Longitude;**
Event schema	start_date_time/end_date_time	**Date Format**
	location	**Latitude** and **Longitude**

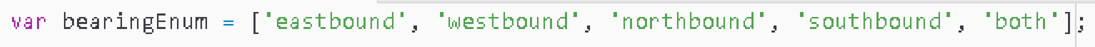
The available schemas are represented as meta-schemas in JSON format, within a Javascript file, as shown in the following figures. Each meta-schema is a Javascript object, with nested objects defining the properties of each parameter in the original schema. These properties represent the parameter's type, the minimum number of occurrences within one data record, defining if the parameter is required or not, and the maximum number of occurrences, which defines if the parameter may be present multiple times in one data record. Two examples of meta-schemas are provided below. Figure 5.8 represents the JSON meta-schema for sensor metadata.



```
var sensorSchema = {
  sensor_id_holder:   {type: "string", min: 1, max: 1},
  concession_name:    {type: "string", min: 0, max: 1},
  road_name:          {type: 'string', min: 1, max: 1},
  road_type:          {type: 'string', min: 1, max: 1},
  km_point:           {type: 'number', min: 0, max: 1},
  sensor_type:        {type: 'string', min: 1, max: 1},
  section:            {type: 'string', min: 1, max: 1},
  state:              {type: 'string', min: 0, max: 1},
  concession_holder:  {type: 'string', min: 0, max: 1},
  bearing:            {type: bearingEnum, min: 1, max: 1},
  country:            {type: 'string', min: 1, max: 1},
  location:           {type: 'geojson', min: 1,max: 1}
} ;
```

Figure 5.8 — Schema definition for traffic sensor metadata

Besides the regular types for each parameter (e.g., number, string), this schema presents two different types: the first is a variable, *bearingEnum*, which is an array that represents an enumeration for the road bearing, as shown in Figure 5.9, and *geojson*, which, along with *date* and *array* in the schema presented in Figure 5.10, is a special data type, which must be handled individually. So, in the case of *geojson*, it represents a GeoJSON point with two values, for latitude and longitude, which means that often two fields in the uploaded raw data's schema (latitude and longitude) must be mapped to the *geojson* field. In the case of the *date*, the ISO 8601 date format must be provided to parse the uploaded raw data's date parameter. The *min* parameter defines the required parameters in the schema: these are *sensor_id_holder*, *road_name*, *road_type*, *sensor_type*, *section*, *bearing*, *country* and *location*, which means that these fields must be present in the raw data.

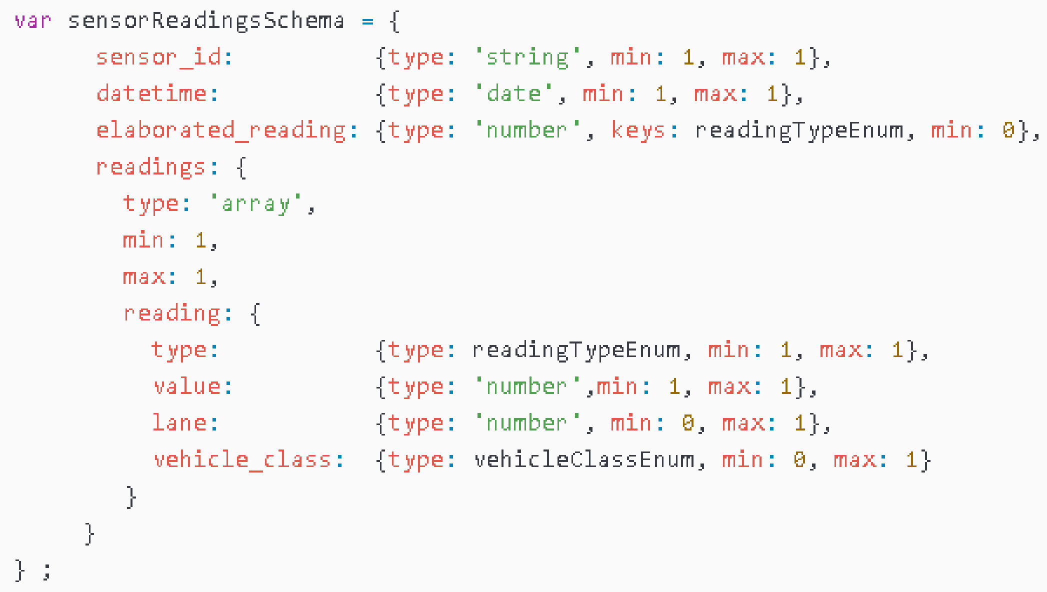


```
var bearingEnum = ['eastbound', 'westbound', 'northbound', 'southbound', 'both'];
```

Figure 5.9 — Enumeration definition for the bearing parameter

Figure 5.10 represents the schema for traffic sensor readings, and it is a bit more complex than the one in Figure 5.8. The complexity arises from the fact that this schema has an array of objects within it. This means that often multiple parameters in the raw data schema will be mapped to objects within this array. Furthermore, there is also several enumeration variables, a date type and a parameter property named keys. This last property defines that an elaborated_reading can have multiple entries, represented by the keys within the enumeration and with numerical values associated to them (e.g., total flow, total occupancy, etc.).

While the meta-schema of Figure 5.8 enables the harmonization process represented in Figure 5.5, the meta-schema of Figure 5.10 enables the harmonization process of the data example presented in Appendix A.11 (page 252), transforming it into the data format shown below, in Figure 5.11.

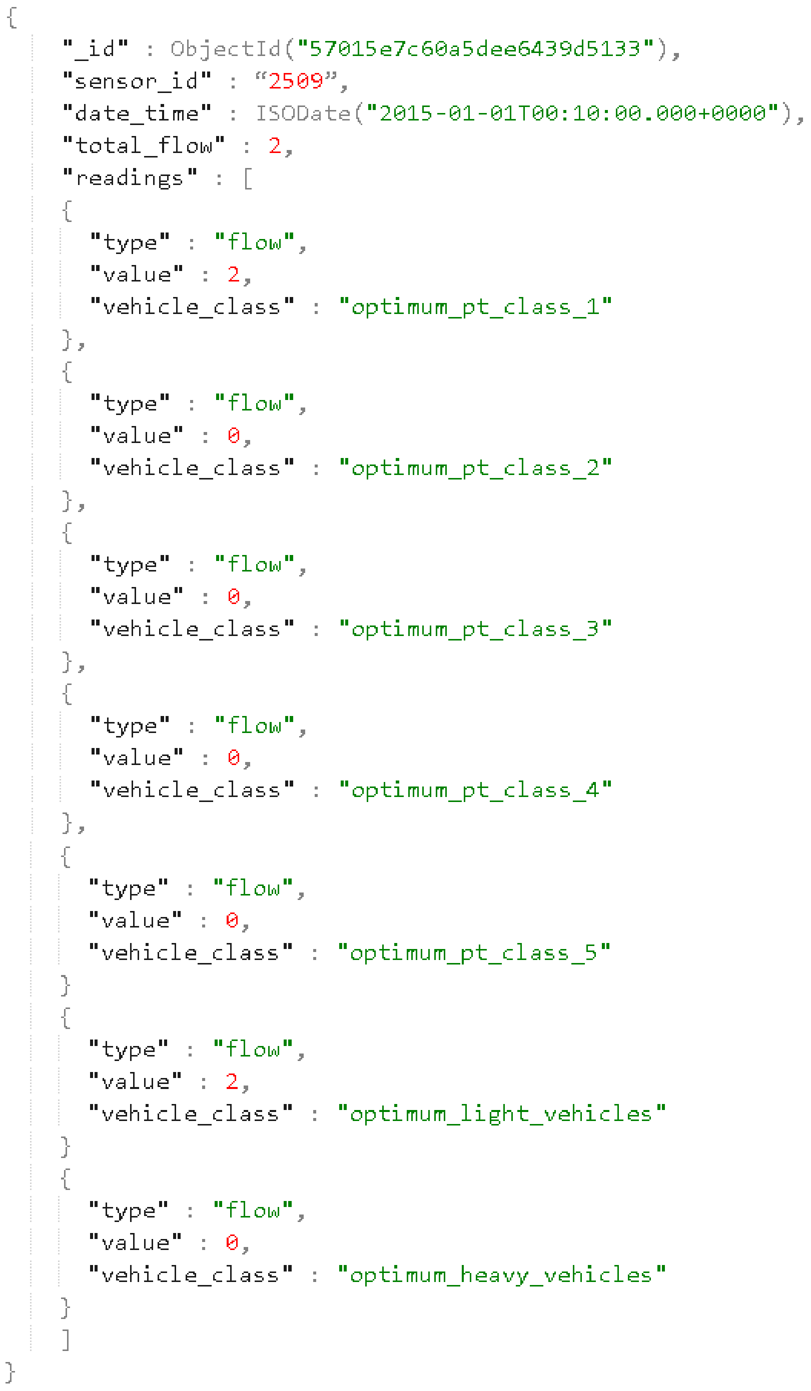


```
var sensorReadingsSchema = {
      sensor_id:          {type: 'string', min: 1, max: 1},
      datetime:           {type: 'date', min: 1, max: 1},
      elaborated_reading: {type: 'number', keys: readingTypeEnum, min: 0},
      readings: {
        type: 'array',
        min: 1,
        max: 1,
        reading: {
          type:           {type: readingTypeEnum, min: 1, max: 1},
          value:          {type: 'number',min: 1, max: 1},
          lane:           {type: 'number', min: 0, max: 1},
          vehicle_class:  {type: vehicleClassEnum, min: 0, max: 1}
        }
     }
} ;
```

Figure 5.10 — Schema definition for traffic sensor readings' data

The toll sensors, from which the data record of Figure 5.11 was collected, only measure the flow of vehicles for each vehicle type, already described in previous sections. Hence, the *elaborated_reading* parameter of Figure 5.10 is given the name *total_flow*, retrieved from the *readingTypeEnum* enumeration, while the *readings* array is used to represent individual flow readings for each vehicle type.

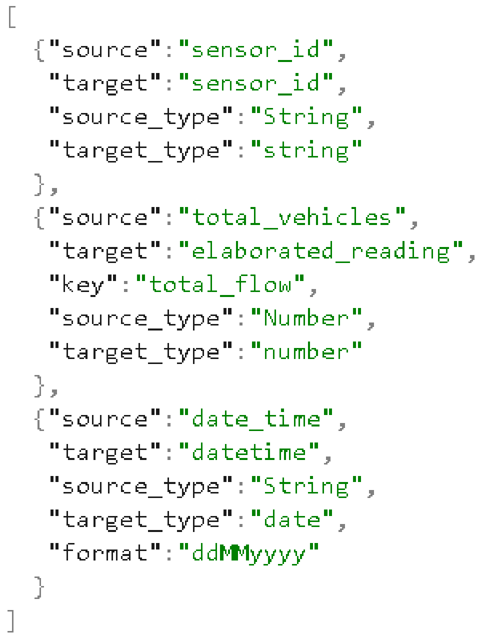
The *total_flow* parameter accounts for the aggregation of all flow readings in the array, whereas the flow readings in the *readings* array represent the flow for each Portuguese vehicle category and aggregated flows of heavy and light vehicles, which are themselves aggregations from the Portuguese vehicle categories (heavy vehicles correspond to categories 3 and 4; light vehicles correspond to categories 1, 2 and 3). Another consideration has to do with the *lane* parameter of Figure 5.10. Since the Portuguese toll sensors measure flows for the entire road, and not per lane, the *lane* parameter is not added to the data record.



```
{
    "_id" : ObjectId("57015e7c60a5dee6439d5133"),
    "sensor_id" : “2509”,
    "date_time" : ISODate("2015-01-01T00:10:00.000+0000"),
    "total_flow" : 2,
    "readings" : [
    {
      "type" : "flow",
      "value" : 2,
      "vehicle_class" : "optimum_pt_class_1"
    },
    {
      "type" : "flow",
      "value" : 0,
      "vehicle_class" : "optimum_pt_class_2"
    },
    {
      "type" : "flow",
      "value" : 0,
      "vehicle_class" : "optimum_pt_class_3"
    },
    {
      "type" : "flow",
      "value" : 0,
      "vehicle_class" : "optimum_pt_class_4"
    },
    {
      "type" : "flow",
      "value" : 0,
      "vehicle_class" : "optimum_pt_class_5"
    }
    {
      "type" : "flow",
      "value" : 2,
      "vehicle_class" : "optimum_light_vehicles"
    }
    {
      "type" : "flow",
      "value" : 0,
      "vehicle_class" : "optimum_heavy_vehicles"
    }
    ]
}
```

Figure 5.11 — Example data record for the Portuguese toll sensor readings

When all required parameters are connected via View 3 (Appendix A.22, page 280), a Submit button is activated on the bottom right of the view, to send the connections' information to the server and to start the harmonization process. For each connection, the required information for the conversion (e.g., parameter names and types, plus special input options) is stored in a JSON object and added in the conversion data array, which will be sent to the server, to support the harmonization process. An example of the conversion array is shown in Figure 5.12.



```
[
  {"source":"sensor_id",
   "target":"sensor_id",
   "source_type":"String",
   "target_type":"string"
  },
  {"source":"total_vehicles",
   "target":"elaborated_reading",
   "key":"total_flow",
   "source_type":"Number",
   "target_type":"number"
  },
  {"source":"date_time",
   "target":"datetime",
   "source_type":"String",
   "target_type":"date",
   "format":"ddMMyyyy"
  }
]
```

Figure 5.12 — JSON map between source raw data parameters and target harmonized data parameters

After these transformations, the harmonized data is stored into MongoDB harmonized repository. This process can be performed on data batches, through bulk insertions, or on data streams, through single record optimized insertions.

5.2.2.3 Adding New Custom Schemas

In addition to the harmonized schemas already provided within the pipeline, it is possible to add new data schemas (e.g., for a new data type). This is an especially important asset due to the flexibility it brings to the Big Data harmonization pipeline. To add a new schema, which will be added to the “Select Data Type” dropdowns of the Collection View’s tabs and used in the harmonization process, the user clicks the “+” button in the Collection View and selects "Add New Schema “. The "Add New Schema" view is presented in View 4 (Appendix A.22, page 280).

This view has two inputs: the “New Data Type” field, in which the user is prompted to insert the new schema’s name, which will be available in the “Select Data Type” dropdowns, and the “New Schema” text area, which inputs the new schema in JSON format. The main rule of the JSON format for a new schema is that the schema is a JSON object containing other objects with three parameters (except in the case of arrays of objects): *type*, *min*, *max*. As previously explained, *type* defines the parameter type (e.g., number, string, GeoJSON coordinates, array, enumeration, date), *min* the minimum occurrences of the parameter and *max* the maximum limit of occurrences of the parameter. An example of a custom schema being added is shown in View 4, named “lxParkingStatusSchema” for the Lisbon's parking data. By passing the verification process successfully, the new schema is added to the available schemas on the collection view, and can be used in the Harmonization view, just as any other default schema.

5.2.2.4 Data Visualization and Sharing

The final step of the pipeline consists of the Data Visualization and Sharing/Export views, presented respectively in View 5 (Appendix A.22, page 281) and View 6 (Appendix A.22, page 282). The Visualization view, which will be further discussed in sub-section 6.2.1, presents some data quality, availability and pattern statistics about the freshly harmonized data. This view is not a fully-fledged data visualization tool, but a static dashboard-like approach to give the user some feedback and initial insights about the harmonized data. The presented information is divided into:

- Statistics: Identified fields, number of records, maximum and minimum values, average and median, minimum and maximum record dates.
- Plots: Limits and average of records (box plot), number of records (line plot) and a sample plot, limited to the first ten thousand records.

The Data Export view enables users to export the harmonized data in JSON or DATEX II-compliant XML formats, once the harmonization pipeline's processes (collection, harmonization) are finished. This service is essential concerning data interoperability with external services/applications. In View 6 (Appendix A.22, page 282), the user provides the data type for the data to export and then, depending on the data type, some other inputs are required to export data from the server:

- Sensor Metadata: Returns the list of sensors. No further inputs needed.
- Sensor Readings: Returns the readings for one sensor and one day. Sensor ID and date inputs needed.
- Events: Returns the traffic events active for one day. Date input needed.

For JSON exports, and since MongoDB stores data using its custom JSON format, the export process is straight-forward. For DATEX II exports, special parsers were developed. Example conversions to DATEX II are presented in Appendixes A.19 (page 271), A.20 (page 274) and A.21 (page 277).

5.2.3 Which Storage Technology to Use? Data-driven Choices

Before wrapping up the chapter, it is worth discussing the choice of storage technology, depending on the characteristics of the data at hand. Data storage technologies are more differentiated nowadays than ever, as was already discussed in Chapter 2. A few years back, RDBMS were the *de facto* technology for data storage, but now, the panoply of types of database systems is growing, from document-based, big columnar and in-memory to specific time series-based and spatial databases. Hence, the choice between any of these data storage systems can be a great support to handle specific data characteristics and to deliver data to the processing and analytics tasks ahead.

In the Big Data harmonization pipeline example, the choice was MongoDB. MongoDB is a JSON document-based storage system, which brings some relevant advantages for the pipeline scenario, when compared with other solutions. The first advantage is MongoDB's ability to tackle Big Data's "variety" aspect, storing data without defining rigid schemas, which means schemas can change as data, application and business requirements evolve. This is especially important due to the volume, geographical spread and specific characteristics (e.g., national vehicle categories, measurements by lane or entire road, etc.) of the different data sources that can be mapped to the same data type. In column-based systems, such as RDBMS, the schema must be defined a priori and must comprise all necessary fields for all the parameters in the various data sources. Further, a strategy to fill fields in the case of missing or null parameters must be defined also before data ingestion.

Another advantage is the fact that MongoDB is a distributed storage system through replica sets, enabling better input/output performance for both batch and streaming data, contrasting with traditional RDBMS. As previously explained in Chapter 2, there are several new storage systems that are designed for distributed architectures, which enables better performance and input/output times. Although there are some cases of traditional RDBMS reinventing themselves to enable distributed storage, such as the PostgreSQL-based Greenplum (VMware, Inc, 2020), the norm for distributed storage systems was newly developed systems based on new storage paradigms, such as the document- and file-based or in-memory concepts.

Other data-driven choices depend still on the performance of the chosen system depending on the Big Data "volume" and "velocity" traits. For instance, when applied on high-velocity streaming data, the Big Data harmonization pipeline may use an in-memory database, such as Redis (RedisLabs, 2015), to optimize input/output processes when collecting data streams and passing them to the harmonization process. In this case, the in-memory database would serve as a buffer for parallel processes that, on one side, hand the data to the harmonization process and, on the flipside, store the data in an historical raw database that can be used in the future for other tasks, such a machine learning model training. In the case of large volumes of batch data, depending on the data characteristics, the chosen system could be a big-column system, such as Apache HBase (The Apache Software Foundation, 2007), when the schema is predefined and rigid, such as in the case of RDBMS, HDFS (The Apache Software Foundation, 2018), when the data is mainly unstructured (e.g., files, documents, etc.), or MongoDB, as already discussed.

Furthermore, the spatiotemporal characteristics of data may be relevant in the choice of database system, especially depending on the business goals and analytics challenges that will be tackled. If both space and time dimensions matter, then PostgreSQL (The PosgreSQL Global Development Group, 1996) or any equivalent or based system are recommended, due to the spatiotemporal functions offered by PostgreSQL and its spatial library, PostGIS (PostGIS Project Streering Committee, n.d.), which also works with Greenplum. If the processing and analytics tasks will focus on the time dimension of the data, then a time-series database, such as InfluxDB (InfluxData, Inc., 2013), should be considered. More complex choices may comprise a personalized approach in which several systems are chosen to perform different functions, such as in the case of an in-memory system as a buffer and other system as a main historical database, or a central, big column storage system that passes the appropriate data to other independent systems, such as time-series systems, to optimize the analytics processes over such data.

These are some examples of the importance of data when choosing the data storage system or systems for the framework. Researchers, practitioners or any stakeholder that has the goal of designing a MobiTrafficBD framework must have a clear understanding of the data at hand, besides their business and research goals, to be able to choose the correct storage system, as it will significantly affect the performance and the modus operandi of the framework. Regarding performance, certain systems and paradigms are better suited for certain tasks, as already discussed, while the way MobiTrafficBD frameworks organize data flows (Figure 3.2) is highly dependent on the type of storage system chosen, in terms of scalability, availability, performance and ability to tackle the business and research challenges that were at the basis of frameworks' creation.

5.3 Guidelines and Best Practices

In conclusion, this chapter presented the relevance of using standards throughout the design and creation of MobiTrafficBD collection, cleaning, harmonization and storage, as well as providing illustrative examples on how to apply these standards and the methodological approach presented in Chapter 3. Further, the example of the data analysis performed during the Data Understanding stage (or equivalent), presents strategies to mitigate the data interoperability issues discussed in the sub-section 2.2.3, while providing a complete example, from start to finish, on how to build a generic Big Data harmonization pipeline. As discussed in sub-section 2.4.1, regarding MobiTrafficBD, harmonization processes are often a neglected research topic, despite its importance in data pre-processing and interoperability.

Hence, this chapter aims at providing a methodological approach for MobiTrafficBD harmonization and storage, along with concrete examples of each of the phases that comprise the design and development of such processes, for researchers, practitioners and RITMOs to follow when building their own harmonization processes. Furthermore, the examples demonstrate the benefits of using the proposed methodological approach, discussed in Chapter 3, the appropriate standards for development support (CRISP-DM) and data modelling (DATEX II) and how to coordinate the choice of preliminary data analyses, the way of developing the harmonization processes from these preliminary analyses and the suitable data storage system. The fusion of all these guidelines, standards and approaches are, from the perspective of the author, crucial for the research panorama of MobiTrafficBD, since there are no similar works in the literature, as already discussed in Chapter 2.

The following guidelines and best practices serve as a summary of the main recommendations made in this chapter:

1. The use of standards should be pervasive in the design and creation of standards, from requirement elicitation modelling languages, Data Mining stage-based methodologies and other generic standard practices for building data-driven frameworks, to specific data-driven standards, such as data modelling standards (e.g., DATEX II, GTFS) to parameter formatting standards (e.g., coordinate reference systems, ISO 8601 for date formats, etc.).
2. The application of standard data exploration methodologies, such as, but not limited to, CRISP-DM, to guide the design and development of data-driven platforms is highly recommended, particularly when developing Data Management components (i.e., collection, harmonization and storage).
3. The chosen data exploration methodology is intended to serve as a conceptual guide and not a strict rule. Depending on the data and scenario characteristics, some stages of the methodology may be sidestepped, such as for instance if the data is already harmonized and provided in standardized formats, bypassing CRISP-DM's Data Preparation and/or Modelling stages.
4. The Data Understanding stage (or equivalent) should define the basis for data cleaning and harmonization processes, by enabling a clear analysis on the data quality and availability issues, such as spatiotemporal dispersion, availability percentages, or any other analysis that enables proper data cleansing, and on harmonization and interoperability challenges, either structural or parametric, by undergoing a thorough analysis

on the commonalities and differences between datasets representing the same data type, and designing or finding a schema that tackles these challenges.

5. The Data Preparation stage (or equivalent) allows researchers and practitioners to perform the necessary data transformations prior to the actual harmonization processes start. This stage comprises the application of data cleaning strategies and the preparation of data for the next steps, such as categorization of data sources into the corresponding data types, correlation between parameters across data sources or mapping of these parameters to standard models, in order to infer if the chosen model is adequate for the available data sources.
6. The Modelling stage (or equivalent) comprises the actual modelling tasks, which include the application of cleaning strategies over raw, unclean data and the execution of harmonization processes over cleaned data. This is the main stage for the chosen methodology, since it puts into practice all insights and strategies collected and designed in the previous stages.
7. When preparing for transformation tasks, such as cleaning and harmonization, thoroughly defining all the necessary transformation rules is a big support for the development of systems to perform these transformations semi-automatically.
8. The transformation process is often semi-supervised, since there are transformations that need specific human inputs to be accomplished.
9. In the logical components and data flows model of Figure 3.2, the Data Cleaning and Harmonization are separated purposely, since it is advisable that data cleaning processes are performed before actual harmonization processes take place. This strategy provides a way for cleaning tasks to be applied over raw data without the issue of corrupting harmonized data, enabling error-free data harmonization processes.
10. The Big Data harmonization pipeline is presented as an example of a generic harmonization platform that promotes standardized data interoperability. The presented example for the design and development of such a platform throughout its stages, from Business and Data Understanding (or equivalent) to Deployment (or equivalent), is suggested as a practical step-by-step guide for practitioners and researchers to apply the prescriptive approach presented in Chapter 3.
11. The choice of data storage systems depends heavily on the characteristics of the data at hand, as well as on the business and research goals. This means that the data drives the choice of storage technology, since it also drives the choice for the processing and analytics processes that should be applied in order to achieve the business and research challenges. Therefore, depending on the Big Data aspects (volume, variety, velocity) and on the way the data can present solutions to the business and research goals,

different storage systems may be more suitable than others. Some examples for this dependency were discussed in sub-section 5.2.3.

6

Bringing Together Big and Spatiotemporal Characteristics to Data Analytics

This chapter will not go as deep as the previous one, since Data Analytics and Visualization, although strongly data-driven, are closer to the business stakeholders, their challenges and goals, and often present a bigger focus on particular scenarios and problems, relegating data to second place. Even so, there are some generic guidelines and best-practices that are particularly relevant for MobiTrafficBD-driven analytics and visualization.

The massive volumes and unprecedented speeds of data being collected nowadays have relevant consequences on the ways Data Analytics and Visual Analytics techniques are applied. Especially in the case of MobiTrafficBD, in which the spatiotemporal components are of the utmost importance, when applying spatiotemporal-aware analytics processes and visualizations that must cope with datasets that present wide spatial coverages and encompassing large time intervals. Due to these implications, the traditional methods to process and visualize spatiotemporal data have been reinventing themselves and new strategies and methods have been created to handle Big Spatiotemporal Data. This chapter covers some strategies and methods to handle these implications on Big Spatiotemporal Data Analytics and Visualization processes.

6.1 MobiTrafficBD Analytics: Considerations about Tools and Methods

The growing availability of both large-volume and fast streaming datasets brought the end of traditional data analysis methods and algorithms and has enabled the emergence of computational models that capture various aspects of massive data computations. Some examples of such models are streaming and distributed algorithms, sublinear time and query time algorithms or complex event processing and deep learning methods. These computational models may be classified depending on their suitability for batch data or for streaming data. There are also some computational models that introduce hybrid approaches that use both batch and streaming data, such as continual learning models (Karim, Soomro, & Burney, 2018). This section will focus on examples and guidelines for these three categories.

In the case of batch data, distributed computational models, such as the ones from the Apache Hadoop ecosystem (and Hadoop itself), and deep learning techniques, such as TensorFlow (Google Brain Team, 2015) or Torch (Collobert, Bengio, & Mariéthoz, 2017), have been the most widely disseminated technological solutions. These models rely on linear time or higher order algorithms that run on top of high-performance, distributed clusters of processing machines, whether deployed on premises or on Cloud-based environments, to perform time-consuming and computation-intensive tasks over large volumes of historical data. For data streams, the goal is to perform processing and analysis tasks in sublinear time ranges, enabling real-time analytics results for time critical decision-making support. In this case, sublinear time, query time, property testing and complex event processing methods on one side, and stream processing models on the other, are the most common topics found in the literature. These methods are designed to produce results in near real-time independently of the size of the input data being analysed.

Besides temporal complexity, already overviewed in Chapter 2, there is also spatial complexity, which denotes the memory space usage proportional to the number of inputs. This type of complexity is tackled by the provision of extreme-scale processing hardware, which possess high memory capacity, along with linear or greater processing times for batch data, enabling better memory management, and "sliding window" strategies for streaming data, i.e., methods for traversing data sets by moving a fixed-size window (subset) over a sequence in fixed steps, which enable the optimization of memory space used by only processing the fixed window data subset. In the case of spatiotemporal data, such sliding windows could even traverse the data set within both dimensions, by dividing the whole temporal and spatial ranges into smaller, fixed time intervals and spatial bounding boxes, limiting the volume of data to be processed on each window.

Finally, there are hybrid approaches that take advantage of both batch-only and stream-only methods to be able to handle streaming and batch data at the same time. Some examples of these methods are continual learning methods that continuously learn and evolve based on the input of increasing amounts of data while retaining previously learned knowledge or hybrid processing engines, which can process both stream and batch data, such as Apache Spark

(The Apache Software Foundation, 2018) or Flink (The Apache Software Foundation, 2014), for instance. The following sections will take a closer look at each of these options.

6.1.1 Batch Data Analytics Strategies

When analysing batch data, the main goal is to process and analyse huge amounts of data in one sweep. This means that short execution times are not the main goal; rather, execution times need only to be short enough to produce timely insights from such amounts of data, depending on the use casa and data-driven goals at hand. Usually, batch processing and analysis tasks are scheduled for specific times of day, week or month. With traditional, centralized processing systems/frameworks, the time to process and analyse such amounts of data have become unfeasible; thus, distributed, parallel processing alternatives entered the game. The performance difference between centralized and distributed alternatives is a common performance benchmark of many research works and point to a better performance of distributed systems to process large sets of batch and historical data, as discussed in sub-section 2.3.5.

Moreover, the performance of these distributed systems is always dependent on the performance of the distributed cluster of machines that supports it, and its characteristics, such as available memory, number of nodes (machines), number of cores per machine, etc. So, depending on the temporal and spatial complexities of the algorithm and its execution time, the scale and performance of the distributed environment supporting processing and analysis tasks may or may not be enough for the processing task at hand. Therefore, even in the case of batch data, the choice of the data processing and analysis methods and algorithms must take into account the performance of the distributed environment.

Besides scalability of the distributed environment, there are other strategies to make batch data processing and analysis methods more efficient. One optimization strategy was already overviewed in Chapter 3 and relates to the distribution of the various technologies (databases, processing engines, visualization tools, etc.) in relation to each other in a distributed environment. One example was to have the database instances in the same node as the processing instances, in order to optimize the exchange of data between such instances.

This strategy is often based on Big Data processing engines' and Deep Learning platforms' new batch data processing paradigms. In the case of processing engines, such as Apache Hadoop and Apache Spark, the shift in paradigm occurred with the introduction of the MapReduce paradigm (Dean & Ghemawat, 2008), the basis for Hadoop. The term "MapReduce" refers to two separate and distinct tasks: the first is the map job, which takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs); the reduce job takes the output from a map as input and combines those data tuples into a smaller set of tuples.

The main drawback of MapReduce is that it takes the whole data set and performs the map and reduce tasks on the whole data set. This means that the system responsible for processing and analysis must have access to the whole data set at the time of performing such tasks. It also entails that, if new data is collected, the MapReduce job must be repeated for the whole data, i.e., the complete old data set plus the new data collected. Newer paradigms, based on or evolving from MapReduce also exist, some of them tackling the drawback above. One example is the Directed Acyclic Graph paradigm (Foraita, Spallek, & Zeeb, 2014), used by Spark, which will be revisited in this section.

Regarding Deep Learning platforms, such as TensorFlow (Google Brain Team, 2015) or Torch (Collobert, Bengio, & Mariéthoz, 2017), the idea is to use distributed environments to create neural networks with more than one processing layer, which is something that traditional artificial neural networks, running in centralized, single-server environments cannot do, since the processing layer, also called hidden layer, is tightly coupled with the server machine it is running on. In Deep Learning, the neural networks may have three or more layers, in their simplest form, to thousands of layers (Huang, Sun, Liu, Sedra, & Weinberger, 2016), due to the distributed environment's ability to parallelize multiple layers across machines or processing nodes.

Both these paradigms are now fully prepared to perform almost all classical Machine Learning and Data Mining processes, such as clustering, classification, prediction learning, anomaly and pattern detection, among others, in distributed environments, as already overviewed in Section 2.4. The extension of processing engines, such as Hadoop or Spark, with other libraries and platforms enables a vast range of Data Mining, Machine Learning and Data Analytics tools and methods to run on top of these processing engines to optimize their performance through parallelization of tasks via distributed environments. Further, specific spatiotemporal analysis methods and tools have also been used in conjunction with these processing engines, such as Apache Sedona (The Apache Software Foundation, 2015), as presented in Section 2.4. Some examples of these libraries and platforms are the Apache Spark Machine Learning Library (The Apache Software Foundation, 2018), which contains pre-implemented algorithms for a wide range of Machine Learning, Data Mining and Data Analytics methods, Apache Mahout (The Apache Software Foundation, 2014), which enables to researchers and practitioners to implement their own methods to run on distributed processing engines such as Spark, and Weka (The University of Waikato, 2005), which provides an extension for running methods on top of a Spark cluster, just to point out a few.

Another way to enable efficient access to data is data indexing techniques, already covered in Chapter 2. Data indexing in databases is a technique to optimize data access, which consists of a table containing the index column, which is often the primary key column for the original table to be indexed, and a column with pointers that hold the address of the memory block where each specific index is stored. Spatial, temporal and spatiotemporal indexing is also recurrently used by having the spatial data (e.g., longitude and latitude columns) and/or temporal (e.g., the timestamp column) columns indexed in the index column (Li, et al., 2017).

Data sampling, i.e., the extraction of a data subset, or sample, that statistically represents the complete data set but significantly smaller in size, is also a strategy to optimize execution times for batch data processing and analysis methods. In the case of MobiTrafficBD analysis, it is usual for RITMOs to provide a spatially or temporally bounded sample of the entire data set, which can span wide geographical areas (e.g., region, country, etc.) across large intervals of time (e.g., years). This sample can be used to extract underlying, preliminary insights, trends and relationships that can be extrapolated to the entire data set. Such extrapolation is mainly due to the cyclic and seasonal trends characteristic of MobiTrafficBD.

This means that, for instance, a sample of loop sensor data, in the form of a GRTS that contains vehicle counts, speeds and road occupancy information for a specific road segment, collected in short time intervals, can be used to extrapolate useful information about road usage that represent the whole data set. Depending on the size of the sample, different insights can be extrapolated: if the sample corresponds to a month of data, daily trends about the movement of vehicles, usage of the road during weekdays and weekends or peak hours can be extrapolated; if the sample corresponds to one year of data, then more comprehensive insights may be extracted, such as monthly and seasonal trends. Hence, the sample size choice is always dependent on the type and comprehensiveness of the analysis and, ultimately, of insights that the researcher or practitioner wants to achieve.

The choice of the right mix of strategies to apply, whether it is the data processing paradigm(s), the tools and technologies that implement such paradigm(s) or other strategies such as data sampling and indexing, is always dependent on a panoply of factors. The three most important ones, according to the author's view, are:

1. the data at hand: data characteristics are the most important element of data-driven frameworks. Data quantity, i.e., the volume of the data at hand, and data quality, i.e., the completeness, accuracy, error count and any other quality measures applied to data sets, are key factors to support the researchers' and practitioners' decision process in terms of the suitable strategies to handle the data at hand. Nevertheless, in the case of large-scale volumes of data, there are some strategies that are always recommended, such as in the case of data indexing or the optimized orchestration of the distributed environment and the allocation of technological instances across the environment.
2. the final goal or objective of the data analysis process: only second to the data at hand, the use case and final goal of the data processing and analysis tasks is always a crucial

factor for the choice of strategies to handle, process and analyse large volumes of data. If researchers point to a preliminary, exploratory data analysis, then data sampling may be a suitable strategy; for more complex analyses that take into account the whole data set, then the mix of strategies chosen must comprise the type of processing and analysis methods and their relative performance in terms of the time needed to achieve the required processing and analysis results. This execution time is crucial for timely support on decision making processes towards the final goal of the use case.

3. the overall characteristics of the distributed environment: last, but not least, and although this is the only customizable and scalable factor of the three (e.g., the system can be scaled with new processing nodes/machines, physical memory, etc.), the hardware and software specifications of the distributed environment are also an important factor for the selection of the strategies to adopt, in order to optimize the performance of batch data processing and analysis tasks. Depending on the quantity and quality of the data and the final goal of the use case at hand, several strategies may be chosen to optimize the distributed environment to cope with both these factors. The way technologies are distributed across nodes in a distributed environment was already discussed in this section, but the choice of processing and analysis methods is also dependent on the environment's specifications. There are different methods that may be used for the same processing and analysis goals (for instance, there are several density-based clustering algorithms), each of which with its own reliability, accuracy and performance characteristics and hardware and software requirements. More than often, methods that produce more accurate and reliable results have greater hardware and software requirements. Hence, the interdependence between the requirements of the analysis to perform, the methods to accomplish such analysis and the distributed environment's specifications in which the analysis is performed must all be considered for the optimization of the overall analysis.

Now that a general overview of the context and recommendations for batch data processing and analysis is given, it is time to go back to the examples presented in Chapter 4 and show the application of such recommendations in real-world scenarios. A general best-practice, also adopted on both the examples that will be presented in this section is to always perform data indexing when dealing with large volumes of batch MobiTrafficBD. Since access to batch data is usually performed via queries to the database, data indexing techniques are recommended so as to optimize data access through database queries. Indexes should be implemented for all data parameters that will be used in database queries, such as unique identifier or primary key parameters and spatial and temporal parameters. Both the examples used data indexing techniques to improve data access performance.

The first example is the "CEP for traffic event detection" example (Figueiras, Antunes, Guerreiro, Costa, & Jardim-Gonçalves, 2018; Antunes H. A., 2017). The conceptual architecture of the proposed complex event processing framework is presented in Figure 6.1. The framework's goal is to detect traffic events from a historical data set of road sensor data. In order to achieve its goal, the framework comprises a set of layers, or steps, that enable the collection of real-time data from traffic sensors, the analysis and understanding of the mobility patterns in the data, i.e., recurrent patterns in the traffic sensor data that denoted normal daily traffic and mobility processes, and the application of the analysis results in the detection of anomalies on these normal behaviours in everyday mobility, in the form of traffic events (e.g., accidents, abnormal traffic intensity, etc.).

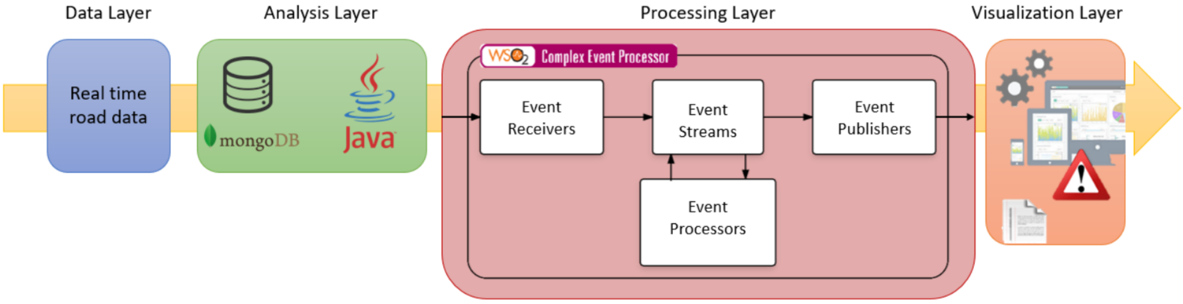


Figure 6.1 — Conceptual architecture of the "CEP for traffic event detection" example (Figueiras, Antunes, Guerreiro, Costa, & Jardim-Gonçalves, 2018)

The Data Layer comprises the real-time sensor data collection process. real-time data collected from traffic sensors were stored in a MongoDB database, building a historical data set of several months, spanning hundreds of sensors. In this example, two data sets were used: the Slovenian traffic sensor data set (Appendix A.5, page 241), in the form of GRTS data, containing data captured in 10-minute intervals for 355 sensors scattered throughout Slovenian roads and highways, and spanning from January 2016 to May 2017 (16 months); and the Slovenian traffic event data set (Appendix A.6, page 243), in the form of ST Event data, which was used to validate the overall performance of the framework by comparing complex event detection results with actual traffic events collected by authorities.

The Analysis Layer is responsible for a preliminary analysis on the data set to extract mobility patterns that can then be used to formulate the complex event processing rules to detect traffic events. A traffic event is an occurrence within the road network, which affects the normal behaviour of such network. Examples of events are the occurrence of an accident, traffic jams, repairs on the road, etc. An example of road use behaviour when an event happens is illustrated in Figure 6.2.

The vertical axis is divided into two metrics, average number of vehicles (blue line) and average speed (red line), and the horizontal axis represented a timeline in hours. The green ellipse represents the period in which an event occurred. It is possible to perceive the effects of an event, mainly marked by the decrease of the average speed and number of vehicles passing through.

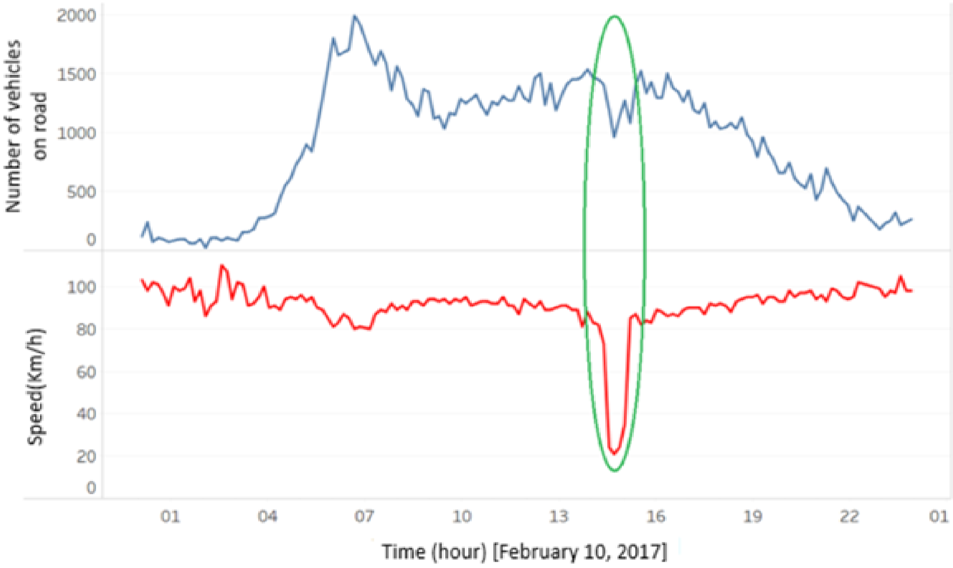


Figure 6.2 — Traffic event behaviour in traffic sensor data

The analysis process comprised the extraction of data samples to analyse the mobility- and traffic- behaviour and road use patterns on weekends, weekdays, holidays and seasonal periods. After this analysis, the hourly and daily patterns were created, and the mobility patterns, i.e., the patterns extracted in the analysis, were stored in the MongoDB database. All processes performed in the analysis layer were based operations of queries in MongoDB and Java.

The Processing Layer is supported by two steps: modelling rules for event identification and optimization of these rules. The rules modelling process, based on the analysis process, approached the creation of rules that allow to automatically identify the diverse types of events, through the influence they cause in the use of roads. As for the optimization process, it involves the evaluation and analysis of results, applying precision, recall and F-Measure algorithms for optimization of event identification rules. All the data processing is performed by WSO2 CEP tool (WSO2, 2005). The WSO2 CEP tool is not a Big Data tool per se, but it presents good performance benchmarks and, since recent releases, provides a distributed mode, in which it leverages Apache Storm (The Apache Software Foundation, 2018) as the support distributed engine, although this mode was not used in the example. Finally, a Visualisation Layer was developed to show the results of the complex event processing tool.

This example is a good example of how to tackle the development of a MobiTrafficBD Framework for traffic event detection without using so called Big Data technologies. Since performance and response time were not the main requirements in the use case, and although WSO2 presents good performance with large data sets, the choice for a traditional complex event processing tool was based on the simplicity of the tool itself and for being an open-source, free-to-use tool. Nevertheless, since the data set was large, some strategies were needed in order to at least validate the framework in a timely manner.

Hence, data indexing was used in MongoDB to index temporal and spatial data fields for easy access. Since, WSO2 relies on SQL-like language to build the CEP rules and directly queries the database to find events, data indexing was considered a good strategy to optimize the performance of the framework. Even so, the validation was performed with only one month of traffic sensor data (February 2017) for only five sensors. Thus, data sampling was used on order to assess the reliability of the framework in capturing traffic events, supporting not only the optimization of the rules to be applied over the whole data set, but also providing insights about the applicability of the framework on the whole data set.

Figure 6.3 presents the results of the validation. On the left side, (a), a comparison between the number of detected events (blue) and the number of traffic events comprised in the Traffic Events data set for the same sensor/road segment (orange), and corresponding matches between the detected events and the database events (grey). On the right side, (b), Precision, Recall and F-Measure validation results considering the different levels of event detection, given by the 10, 20 or 30% variation of the average speed of the vehicles in comparison with the average values of mobility patterns. The speed variation value which obtained better results in the events detection was for 20% of the speed variations.

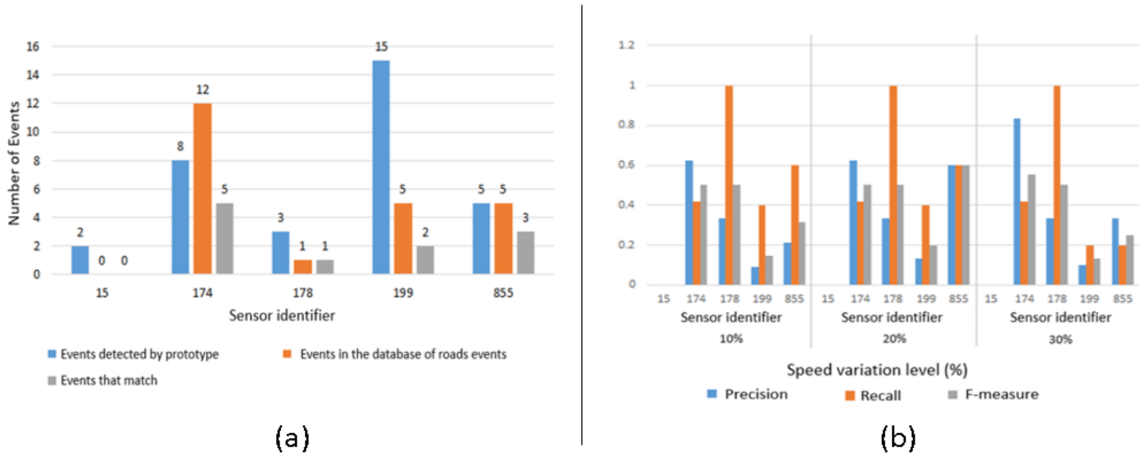


Figure 6.3 — CEP for traffic event detection results. (a) comparison between detected events and the traffic events data set, gathered by authorities; (b) Precision, Recall and F-Measure values for speed variations of 10%, 20% and 30%

Except for one sensor (174), the number of road events detected by the prototype is often higher than those entered in the database, which indicates that the framework detects events that were not gathered by the authorities or road concessionaires. Figure 6.3 also shows that not all events entered in the road events database correspond to those detected by the prototype. This occurrence is due to the fact that although the events occurred, the variations detected did not shift enough from normal behaviour, making it impossible to detect them by the framework.

The second example corresponds to the "Public transport network status analysis and visualisation " example (Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019). The main goal of this example was to design and develop an open-source Big Data framework for the analysis and visualisation of large volumes of public transportation data. Thus, this is a case in which the proposed prescriptive methodology was applied to a scenario other than the ones supporting RITMOs, although it maintains the intrinsic connection to Mobility and Traffic. Specifically, the public transportation data analysis focus on three main Mobility-related indicators:

- Connections: the commuter (passenger) changes to another route on the same mode of transport (e.g., changing lines at a subway station).
- Transhipments: the commuter changes to another mode of transport (e.g., after exiting the subway, the commuter catches a bus at the subway station's exit).
- Pendular movements: the commuter uses the same route and modes of transport from and to the origin location at different times of day (e.g., going to work and coming back home).

This example used two data sets: the first set contains the ticketing data of seven different public transport entities operating in Lisbon; the second dataset represents the General Transit Feed Specification (GTFS)-based (Google, Inc., 2006) data from Lisbon, Portugal. Both datasets cover a temporal dispersion of a month, May 2018. The ticketing data is the biggest data set, containing more than 55 million records, representing entry and exit validation in Lisbon's public transportation network. These records are acquired by different public transport operators daily, through the acquisition of data from smart cards. All the data records were gathered from more than 4500 different stop stations, combined with 1500 different types of tickets. The second dataset is represented in GTFS, a specification that defines a common format for public transportation schedules and associated geographic information. GTFS was firstly introduced by Google to handle Google Maps' public transportation information and contains information of urban public transportation schedules, stops and routes. In this example, the dataset corresponds to GTFS information about public transportation in Lisbon and is used to geographically pinpoint the validations and associate the validations to existing routes and stations.

Figure 6.4 presents the adopted architecture for the proposed framework that can be mapped to the logical components and data flows model in Chapter 3 (Figure 3.2). The Data Collection and Ingestion Layer is responsible for the ingestion of ticket validation data from different public transport operators and, since data comes from multiple sources, the first data harmonization procedures are performed.

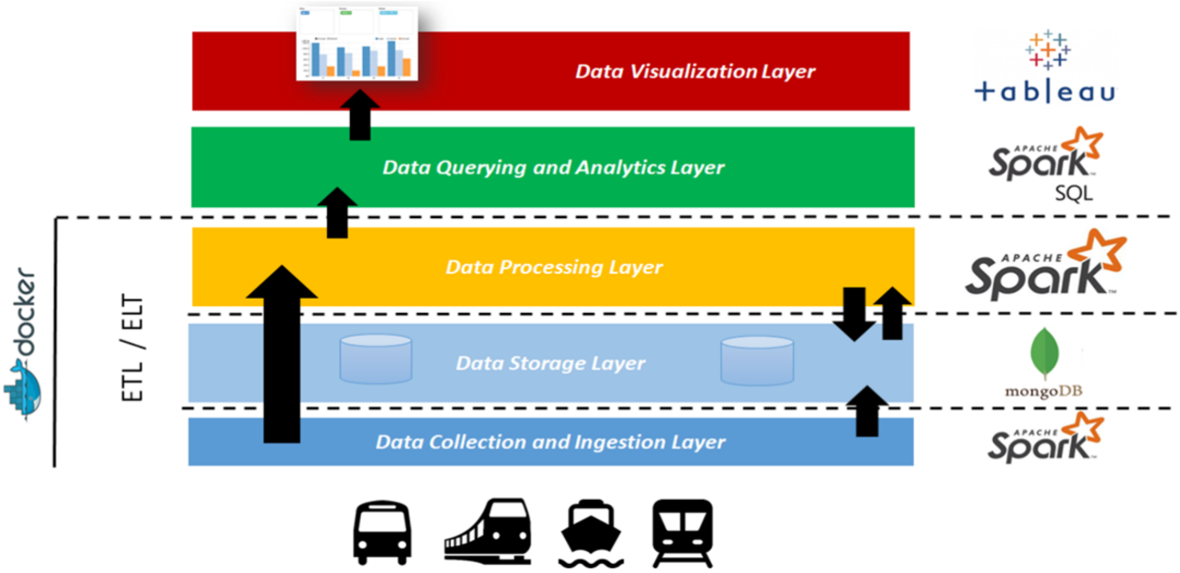


Figure 6.4 — Big Data architecture for the " Public transport network status analysis and visualisation " example (Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019)

The second layer is the Data Storage Layer, which is responsible for storing all data. In this layer, data presents different schemas and formats (e.g., validations, station locations, ticketing). The Data Processing Layer, based on Apache Spark, is responsible for heavy, demanding and intensive data processing tasks. Finally, the data querying and visualization layers expose the processed insights through query engines, such as Apache Spark SQL, and visualization tools, such as Tableau. For easy, scalable deployment and management, the entire Big Data architecture was built on top of a Docker Swarm environment, which is easily scalable to distributed clusters.

The focus will be given to the Data Processing layer. The first task for this layer deals with grouping different validations per trip. A trip is a set of one or more validation records gathered from the same user during a defined time frame. To define the optimal timeframe between validation s five different time intervals (30 minutes, 45 minutes, one hour, one hour and 15 minutes and one hour and 30 minutes) were analysed, concluding that the one-hour interval was the optimal time interval to gather validations corresponding to the same trip. The definition and organization of validation data into trips was the first step to create and automatically analyse complex mobility patterns in ticketing data. At this stage, all trips with only one validation were excluded because such trips did not contain any insightful indicators, such as connections or transhipments.

The second processing task is performed in parallel with the first. This task encompassed the complex process of mapping stops and stations in the ticketing data with stops and stations in GTFS data, to be associated with public transportation routes, in order to associate the station where the user validates the ticket with certain routes. Thus, the location in GTFS data is linked to the stops and stations, connecting the stations and stops with one or more routes. To create this mapping, a proximity algorithm was used to enable the matching between both data sets.

After these two tasks, it is now possible to combine the two resulting data sets and analyse them in order to identify complex indicators such as connections, transhipments and pendular movements. A custom algorithm that analyses individually each trip was developed for Apache Spark. This algorithm maps route and station data with validation records for each trip and identifies whether the user made a connection or a transhipment. To identify connections all trips that change route but keep the same public transportation mode were considered. To identify transhipments all trips that combined more than one public transportation mode were considered. Finally, the last step consisted in the identification of pendular movements, through the creation of an origin-destination matrix for each trip, based on the routes, and whenever a user made two trips with origin-destination pairs reverse to each other in the same day, these trips were identified as being a pendular movement.

The framework was tested on a single machine, but distributed across twelve CPU cores, achieving much better performance, producing useful insights in just four hours, comparing with traditional Data Warehousing processes, which performed similar analyses in a few days, using a Cloud-based proprietary environment. The framework used data indexing and sampling (only one month of data was used in the validation), but further optimizations could be done, shrinking the execution time to less than four hours for one month of data (e.g., the framework is reading/writing to MongoDB on each task, which is an unnecessary intermediate step for the final roll out of the framework).

6.1.2 Streaming and Hybrid Data Analytics Strategies

Most Data Analytics techniques assume a finite amount of data, persisted in a database or data storage tool, and perform their analysis in multiple steps by applying a batch method. Batch methods may pass multiple time through the same data instance along the analysis and entail that the whole data set is known and available before the analysis process begins (Galić, 2016). But these techniques are not applicable to spatiotemporal data streams due to constraints imposed by the characteristics and nature of the streams. These characteristics pose several challenges:

- Single pass: Since streaming data arrives continuously, analysis must be done in a single pass and in real-time over the data.
- Limited memory: Since data streams are possibly unbounded, i.e., may be infinite, storing each arriving data object is not possible.
- Limited time: The chosen methods must cope with the speed of the stream, meaning that the execution time should be less than the average arrival time between two consecutive objects in the stream.
- Varying time: Since the arrival time between two objects in the stream can vary greatly, the execution time will also vary according to the arrival time.
- Concept drift: When using model-based methods (e.g., clustering, classification, prediction), the underlying data model may change over time and the changes should be detected by streaming methods.
- Parametrization: Some methods need information about the data set a priori to be parametrized before execution. Some examples are the number of clusters in some clustering algorithms, or the training data sets for prediction methods. Streaming methods must be free of a priori parametrization since the whole data set is not known beforehand.

Although these challenges were initially proposed for data stream clustering methods in (Galić, 2016), they can be extended to all Data Analytics tasks, from clustering and classification to prediction or anomaly detection. In fact, the definition of data stream clustering used in (Galić, 2016) "*to maintain a continuously consistent good clustering of the sequence observed so far, using small amount of memory and time*", can be generalized to cope with all types of data stream analysis: "to maintain a continuously consistent good analysis of the sequence observed so far, using a small amount of memory and time". This definition can be broken down into several important considerations for processing and analysing stream data. Firstly, to maintain a continuously consistent good analysis entails that the analysis must consistently present good analysis results across the whole streaming data set, but this benchmark must be evaluated for all the observations (data records) registered until the present moment. Furthermore, the analysis process must consider the requirements of limited memory and limited time already overviewed.

There are several streaming analysis methods' types that have been used to cope with data streams, such as sublinear time, query time or property testing methods. Sublinear time methods produce results slower than the size of the problem, i.e., the output is produced in less time than the time between two consecutive data objects in a stream. Query time methods are used when a query is performed to get each data object in the stream. Queries take some time to deliver data inputs, and query time methods produce outputs at the same rate as the query delivers inputs. Finally, property testing methods classify individual data objects that arrive in the stream as conforming to a certain property or not.

The main advantage of these methods is that the execution time for these algorithms is equal or lower than the time to deliver data objects from the stream to the method, whereas the main drawback is that, due to the time and memory limits, these methods do not produce accurate results, delivering approximations or probabilities as output. But, although they are still valid strategies to tackle the challenges of streaming data, new strategies for analysing such data have been introduced more recently, with the advent of distributed processing engines. Stream processing engines, such as Apache Storm (The Apache Software Foundation, 2018) and Apache Flink (The Apache Software Foundation, 2014), have been broadly used in research works in the last years.

These stream processing engines enable not only the application of already existing sublinear time, query time and property testing methods but also the implementation of streaming versions of traditional batch data methods, through the use of programming abstractions, such as in the case of open-source Apache Samoa project (The Apache Software Foundation, 2015) or the research work presented in (Abeykoon, et al., 2019), in which authors extend two known methods, the Support Vector Machine classification method and K-Means clustering method, to produce online versions able to run on top of stream processing engines, such as Storm and Flink.

These otherwise batch data-driven methods are extended for streaming data by applying a specific strategy to data stream processing and analysis: the sliding window model. For sublinear time and similar methods, the main stream processing model is the stream model, which is based on processing and analysing each individual data object in the stream at a time. On the other hand, the sliding window model aggregates and processes a set, or micro-batch, of data objects that were received from the stream within a fixed time interval - the window. After the processing of this set is finished the fixed time window is "slided" to the next set of data objects received.

This strategy allows streaming methods to have bigger execution times since, instead of the execution time being limited to the time of arrival of each data object, it becomes limited to the time range of the sliding window. This allows for methods to process and analyse all data objects on the window's time frame and to synchronize the outputs into a data model that comprises the outputs for all observed data objects in the stream until the present window. Thus, it is possible to implement more accurate model-based methods, such as classification and prediction methods, that take into account not only data comprised in the present time window, but also the whole data observed up to the present.

Therefore, the decision between sublinear time and similar methods and online versions of batch data-driven algorithms or between a stream processing model or a sliding window one, must consider data stream characteristics, such as the rate of reception of each data object, and requirements for the analysis to be performed over the stream, as in the case of execution time and accuracy of the method.

Hybrid methods, i.e., methods that can cope with both batch and streaming data processing and analysis are often based on micro-batches and the sliding window model, with the sliding window size varying depending on the nature of the data and the use case at hand. In the case of data stream-based scenarios, the window size will be shorter whereas for data batch-based scenarios, the window size will be longer.

To exemplify the application of the proposed methodology on streaming or hybrid data analysis scenarios, two use cases will be presented: the "Real-time traffic flow analysis" (Figueiras, et al., 2018; Rosa, 2017) and "Twitter mining for traffic event detection" (Gutiérrez, Figueiras, Oliveira, Costa, & Jardim-Gonçalves, 2015) examples. In the "Real-time traffic flow analysis" (Figueiras, et al., 2018; Rosa, 2017) example, the main goal was to monitor and analyse traffic in real-time, by applying stream processing tools and methods over real-time traffic sensor data, in the form of GRTS. The data set used was the Slovenian Traffic Sensor data set (Appendix A.5, page 241), which provides data, collected in 10-minute intervals from 355 sensors throughout Slovenia, comprising information about average speed, average time gap between vehicles and occupation percentage per lane in each direction. Hence, each sensor reading for a three-lane road, such as a highway, will provide six different data objects, one per lane in each direction.

The "real-time traffic flow analysis" example had the main objective of swiftly perform different types of aggregation processes over data coming from real-time data streams to enable efficient analysis and visualisation of GRTS for RITMOs. The focus was on the delivery of added value insights that were simple and quick to process and aggregate but that would provide more useful insights to RITMOs, when compared with insights extracted from raw real-time data.

Two main aggregation tasks were addressed in this scenario: First, each sensor reading had the lane-specific data objects aggregated in terms of average values for speed, occupancy and gap between vehicles, creating an aggregated view of the entire traffic in each direction.

For instance, for a three-lane highway, six lane-specific data objects (three in each direction) were aggregated to create two direction-specific data objects. This aggregation process was performed each time a new sensor reading was received from the stream. Second, an hourly aggregation task was performed for each sensor, thus six sensor readings had to be aggregated per hour. In contrast with the previous aggregation task, this task had to maintain information each sensor in order to correctly aggregate readings belonging to the same sensor, hence requiring an hourly aggregation model for each sensor. Therefore, the first task represents a simple streaming data task and the second task, a hybrid data task.

The chosen data flow and technologies are presented in Figure 6.5. In this use case, the traffic sensor data was already available and stored in a MongoDB instance and the real-time stream had to be simulated. RabbitMQ (VMware, Inc., 2007) is the technology responsible for collecting data from MongoDB and creating the data stream, sending the data as stream messages to Apache Storm. Each message comprises readings for all sensors at a specific time. Apache Storm (The Apache Software Foundation, 2018) is the main streaming processing engine and is responsible for the data aggregation tasks, and also for sending the aggregated outputs to the visualisation component, which will be overviewed in Section 6.2.

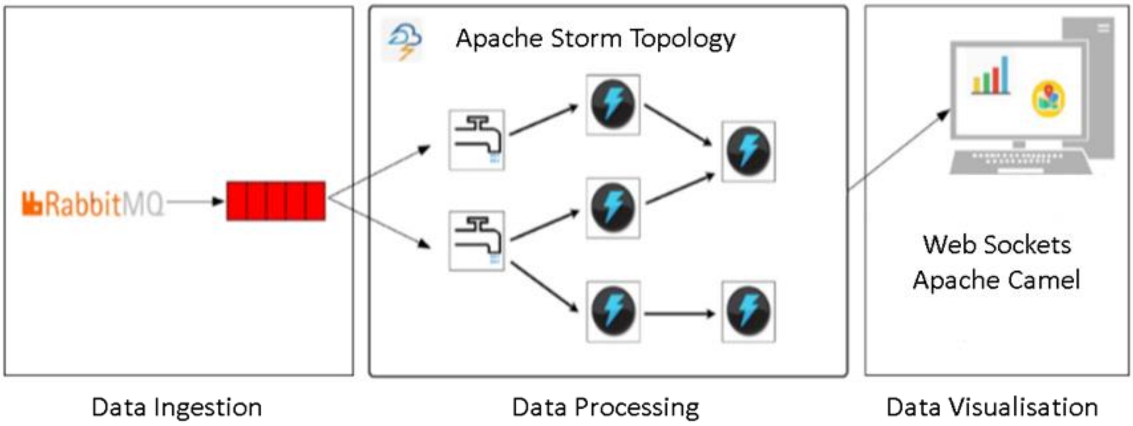


Figure 6.5 — Real-time Traffic Flow Analysis technical architecture

Each aggregation task was performed by a specially purposed Apache Storm topology. These topologies represent graphs representing the entire computation flow to be followed by Storm and comprise spouts, the data sinks that connect to message queues and other streaming data provision mechanisms, and bolts, the individual processing steps of the topology. The base topology graph for both topologies is shown in Figure 6.6.

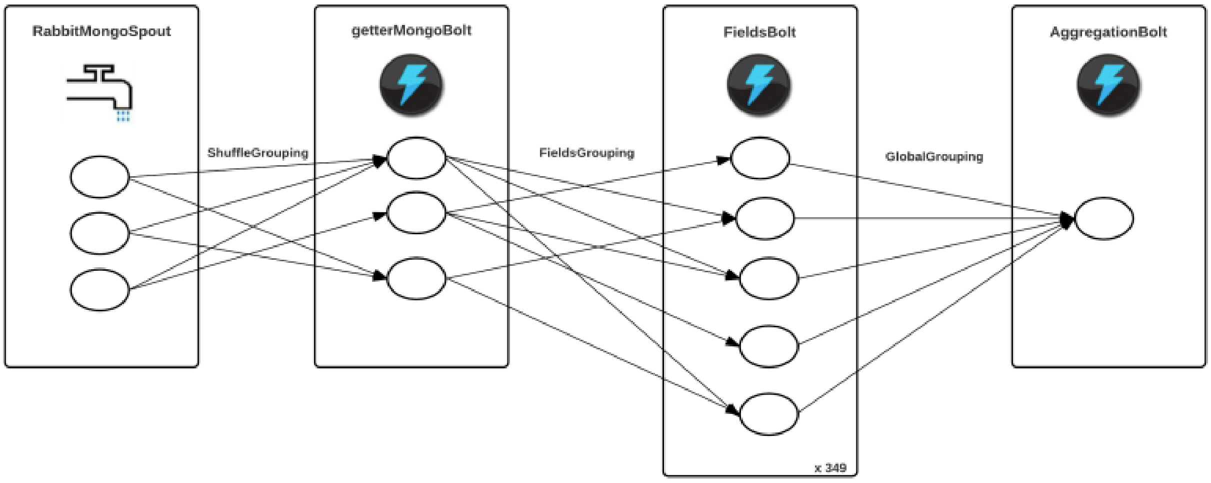


Figure 6.6 — Base topology graph for streaming data aggregation tasks

The RabbitMongoSpout verifies the arrival of messages provided by RabbitMQ every millisecond and serializes messages into data objects to be used by the bolts. GetterMongoBolts are a set of distributed processing tasks that randomly consume data objects provided by the spout and separate readings by sensor, through the sensor's unique identification number. FieldsBolts perform aggregation tasks on data objects for each individual sensor, meaning that there is one FieldBolt per sensor.

This is where the main difference between both topologies is noticeable: for the streaming data use case, in which there is no need to keep information from past readings and data objects, the grouping, i.e., the distribution of data objects to each bolt, was done through a FieldsGrouping strategy in which data from each sensor is divided by the fields in the data (timestamp, sensor ID, average speed per lane, occupancy percentage per lane and gap between vehicles per lane); for the hybrid processing task, in which there was the need to store past information to aggregate individual sensors' readings hourly, a custom grouping had to be developed to guarantee that each bolt would keep some sensor data for the same sensor, for the necessary period of time. Finally, the AggregationBolt aggregates all the output data from previous bolts for storage or visualisation purposes.

To test and validate the proposed architecture, seventeen months of traffic sensor data for 350 sensors were fed into the topologies in the fastest way possible, using a distributed environment on a single machine. Two setups for Storm were used, one with one worker node and another with four worker nodes. The average difference in performance between both setups is around 0.2 milliseconds. In each of the setups, 200 GetterMongoBolt instances and 350 (number of sensors) FieldsBolt instances were used.

The first topology was able to ingest and process 8.8 messages per second, which means that the aggregation task took 114 milliseconds per message, while the second topology was able to ingest 8.6 messages per second, which means that the second aggregation process took

116 milliseconds. These results could be further improved with the deployment of the proposed architecture on a more powerful distributed cluster environment. This example shows the relevance of using novel stream processing engines to efficiently process and analyse streams of MobiTrafficBD, in the form of GRTS, by providing a wide range of performance optimisation strategies, such as the distribution of tasks among worker nodes, processing threads and other resources available for the processing and analysis tasks.

The second example corresponds to the "Twitter mining for traffic event detection" (Gutiérrez, Figueiras, Oliveira, Costa, & Jardim-Gonçalves, 2015) use case. This example comprises a traffic event detection framework that processes and analyses Twitter messages to extract information about traffic events, such as their type, location and severity. The rationale behind the example is that Twitter messages, also known as tweets, may be seen as ST event data that contain useful information regarding traffic events which were created by people who witnessed the events. Social networks can be seen as a mechanism which allows the detection of very small events, such as a damaged car in a side street. In addition to that, the interval between the occurrence of a traffic incident and the publication of a tweet about such incident usually tends to be much less, when compared to the time required for a news agency to share information about such event.

Nevertheless, this source of information is often flawed in many ways. One of the most important limitations in extracting useful information from a tweet is its 14-character limit. Additionally, users can make spelling mistakes, and use varying conventions and abbreviations; the quality of content is not as good as in news articles, for instance. In addition to the credibility of the content, the credibility of user profiles and their geo-references are questionable as well. Not all users have to provide their locations, or their city of residence in their profiles. As a result, the most obvious problem is uncertainty and the lack of rich and reliable data. Moreover, tweet density depends heavily on the population and Twitter usage in a region. Location estimation using the content in social networks has its own challenges, namely the uncontrolled and the limited content. Tweets enable writers to add geo-referenced locations to the tweet, but there are no guarantees regarding the spatiotemporal closeness to the traffic event's location and time (e.g., the user may be driving at the time of the event, and just tweets about it after reaching the destination).

The proposed computational framework shown in Figure 6.7 is able to: (i) classify a stream of traffic-related tweets adopting machine learning techniques, (ii) extract a set of contextual information such as: the location and type of event, (iii) geolocate the event on a map and (iv) the follow up of the incident i.e., monitoring incidents' evolution.

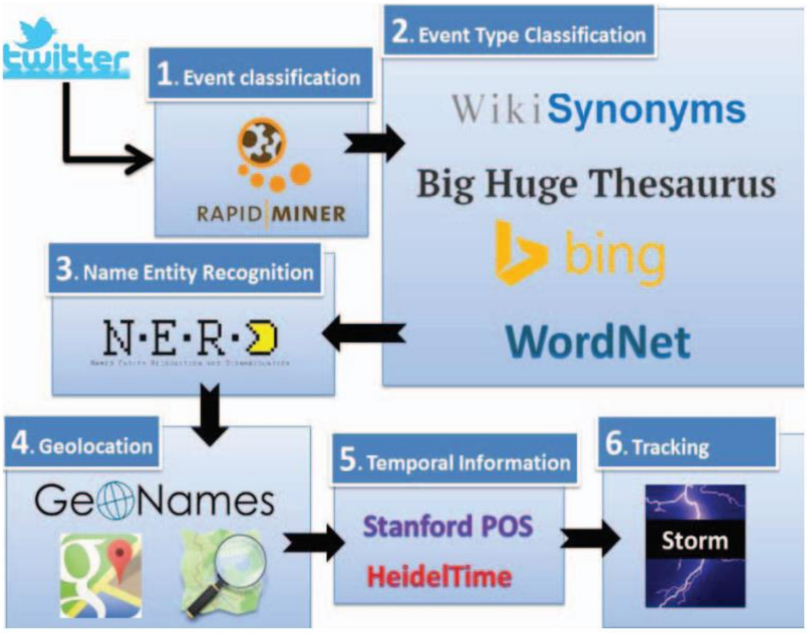


Figure 6.7 — Twitter Mining for Traffic Event Detection framework concept (Gutiérrez, Figueiras, Oliveira, Costa, & Jardim-Gonçalves, 2015)

This framework retrieves the data stream from Twitter's API and only uses a Big Data-based tool, one of the earlier versions of Apache Storm, to track tweets belonging to the same traffic event. All the other tools are traditional machine learning technologies, such as the RapidMiner (RapidMiner, Inc., 2013) Data Mining suite, the name-entity recognition engine NERD (Rizzo & Troncy, 2012), geocoding Web services and Part-of-Speech (StanfordPOS (Stanford NLP Group, 2003)) and temporal (HeidelTime (Strötgen, Zell, & Gertz, 2013)) taggers. The framework is a sequential pipeline in which each task is performed in a queue for each received tweet. The framework has the following modules:

- Classification: RapidMiner was used to implement the SVM learning model and the classification of new tweets. The SVM was trained with a dataset of 10.000 tweets, 5.000 tweets "positive" i.e., containing relevant information about traffic events and 5.000 tweets considered "negative", i.e., not containing traffic-related information. When a new tweet arrives in the stream, it is classified as positive or negative. If positive, the tweet continues through the pipeline, otherwise it is discarded. This model achieved a 95,5% accuracy in classifying tweets (Gutiérrez, Figueiras, Oliveira, Costa, & Jardim-Gonçalves, 2015).
- Event Type Classification: The event type classification is a multi-classification mechanism that relies on matching synonyms with terms available in the tweet to classify the type and cause of a traffic event. Several Web thesauri and dictionaries were used to extract synonyms regarding a list of traffic events, comprising traffic event types and words and synonyms that relate to each of the types (e.g., road closure event may be

matched with words such as "obstruction" or "barrier"). For instance, the tweet "N2 - one lane closed due to snow" contains the words "closed" and "snow", which will be matched to "Road Closure" event type and "Wind & Snow" event cause.

- Name-Entity Recognition: To extract locations from tweets, NERD was used for name-entity recognition. NERD extracts possible location names, such as names of streets, highway junctions, etc. This task was able to recognise 81% of the location-related entities in tests performed (Gutiérrez, Figueiras, Oliveira, Costa, & Jardim-Gonçalves, 2015).
- Temporal Information: To extract temporal information, such as time of day, from tweets, Part of Speech and temporal expression taggers were used. The idea is to understand the grammatical tense of the tweet message and to find temporal expressions in order to extract insights about the event timeline. These taggers are examples of sublinear time methods used for stream processing.
- Geolocation: For the task of extracting the location from the entities detected by NER engine, several geocoding Web services were used. The goal is to have the most accurate information about where the event has occurred. Several levels of accuracy are being considered here, namely: road, city, region and country level. It is possible that an event involves several places at the same time, such a serious accident which produces a traffic jam on a highway linking two cities. The more similar tweets about a particular event are analysed, the more accurate the geolocation of the event itself.
- Real-time Clustering: A first version of a real-time clustering mechanism that would aggregate over time tweets concerning the same traffic event was developed in an earlier version of Apache Storm. This step enabled the tracking of the traffic event, from the first tweet posted about the event to the last reference to the event.

The main conclusion taken from this example is that it is possible to have a framework based on traditional technologies able to tackle MobiTrafficBD. In this case, a mix of fast classification and matching methods, POS and temporal taggers that have sublinear time complexities, and the use of geolocation Web services, enable the discovery, classification, root cause analysis and spatiotemporal insights' extraction from simple, limited and unstructured text messages, such as tweets. This example brings to light several of the points tackled in this section: the use of sublinear time or similar methods over streaming data, in the case of the taggers, the modular application of both traditional and early versions of Big Data technologies to analyse streams of MobiTrafficBD, depending on the performance and accuracy requirements, and a combination of strategies to optimize stream processing tasks.

6.2 MobiTrafficBD Visualization: Tools and Methods

The final component of the logical components and data flows model in Figure 3.2 is the Data Visualisation and User Interaction component. Visualisation of MobiTrafficBD presents several challenges related to the Big Data and spatiotemporal characteristics of the data at hand. Since the subject of data modelling and standardization, which is a crucial factor for sharing MobiTrafficBD, was already discussed, this section will not delve into Data Sharing; rather, it will focus on Visualization, Visual Analytics and User Interfaces.

Aiming once again for a data-driven perspective, the choice for the right visualization strategy. i.e., the strategy that will grant a better understanding of the data itself and of underlying patterns and insights to RITMOs and other stakeholders, depends heavily on the nature and characteristics of the data set at hand. A visualization strategy involves the choice of a mix of techniques, tools and technologies that range from how data is casted to the visualization interface to the visualization method itself. Some common strategies for MobiTrafficBD visualization were overviewed in Chapter 2, such as animated visualizations, linked views and space-time cubes.

Batch, streaming and interactive data visualization strategies present different challenges depending on the volume and speed of MobiTrafficBD, while the spatiotemporal dimensions of MobiTrafficBD have a crucial role in the decision over the visualization strategies to use. Depending on both the data at hand and the scenario for which the visualization strategy is aimed, the guidelines and best practices will be focused on three main aspects:

- How will the data be communicated between a distributed environment to the interface or front-end? This is a major challenge when dealing with Big Data in general, but even more important due to the volume, speed and complexity of MobiTrafficBD. It focuses on data access and querying.
- What data will be applied to the visualization method? This challenge focuses on how data is prepared for visualization, dealing with spatial and temporal granularities (e.g., temporal granularities: hour, day, week; spatial granularities: road point, road section, entire road), data aggregation and summarization.
- What visualization methods to apply? The challenge of choosing a visualization method is the main challenge of data visualization, since it is dependent on the data characteristics, the use case to tackle and the challenges to surpass, and the final audience of the visualization.

These points are not exclusive, but complementary: for instance, in the case of batch data visualization, in which the volume of data to visualize can be substantial, a mix of the right data communication and data aggregation/summarization strategies may be one option to optimize the time to render the final visualization. Thus, it is time to dig deeper into the different data visualization strategies.

6.2.1 Batch Data Visualization Strategies

When the use case deals with batch data exploration, processing and analysis tasks, the visualization of the tasks' outputs presents several challenges specific to the data volumetry. First, the communication of data from the processing tasks to the visualization interfaces, more than often meaning a communication between servers and final users' machines through the Internet, is highly dependent on the volume of the data to broadcast. Second, even when the data is efficiently broadcasted to the client side (the user's machine), the data volumetry has consequences on how to select, filter and query the data to overcome several challenges, such as the overplotting problem: when visualizing very large data sets, the visualization may become hard to understand, due to the overlap and extreme density of data points. Third, the visualization strategy must account for what is being analysed, what are the main goals of the use case, and how can the visualization best support the use case.

Regarding the data communication between server and client, there are several strategies that can be adopted to ease the load of communicated data, depending on the goal of the analysis. One important batch data analysis is the exploratory analysis of data to uncover underlying patterns, trends, and anomalies that will lead to the actual data processing and analysis processes development. This is a crucial task for practitioners and researchers, since it provides an initial view on patterns and trends that leverage the design and development of the final processing and analysis methods.

In the case of exploratory analysis, the best practice is to use data sampling before communication, to reduce the volume of communicated data. For instance, if the data set comprises traffic sensor readings, in the form of GRTS, spanning years of data records and covering several types of roads and highways in a country, the strategy to adopt is often to select and analyse a data sample and extrapolate the outputs and insights of the sample's analysis to the whole data set. In the case of a data set spanning several years and covering a wide range of roads in a large geographical area, the sample must be chosen according to several considerations:

- The temporal span of the sample should enable the extraction of insights under several temporal granularities. This means that the sample must contain daily (e.g., traffic peak hours), weekly (e.g., weekdays versus weekends), monthly (e.g., monthly traffic patterns) or seasonal (e.g., traffic in the Summer versus traffic in the Winter) insights.

- The spatial cover of the sample should account for the different types of roads (e.g., national road, highway, freeway) and regulatory boundaries (e.g., urban setting versus rural setting). When extracting the sample, it must contain enough spatial information to characterize the spatial dispersion in the original data set.

For data sets with shorter time spans and shorter spatial dispersions, the latter considerations remain, although insights from higher granularities may not be possible to uncover. For instance, if the data set's time span is one year and the sensors are only deployed on highways, then insights about seasonality and about different traffic behaviours according to road type may be impossible to obtain. Hence, the shorter the spatial and temporal dispersion of the data set, the smaller the set of granularities from which insights can be obtained.

In other types of batch data processing and analysis tasks, more than often the outputs for such tasks are already aggregated or summarized as a result of the analysis method used, such as in the case of clustering, classification, pattern and anomaly detection, to name a few. Nevertheless, data aggregation, sampling and summarization techniques may still be used to further ease the load of data to be communicated. As an example, in the case of visualizing outputs from prediction methods, only a sample of the final predicted data set may be communicated for visualization.

Regarding the challenges of data volume on visualization, such as the overplotting problem, several strategies can aid in the resolution of such challenges. Data aggregation, summarization and sampling represent not only valid strategies in this case, but also can be seen as the main strategies to tackle these challenges. Other strategies often use data aggregation, summarization and sampling as the basis for tackling data volume-related challenges in visualization. Some examples are interactive, linked views and other visual querying methods. In, multi-window strategies, such as interactive and linked views (Figure 2.3), the spatial and temporal windows can be used to filter data by performing "pan" and "zoom" actions over them. For instance, if the user zooms in on or pans to a particular geographical area on the spatial view, the data shown in other views should reflect the spatial bounds chosen by the user (e.g., showing average speeds collected from sensors in roads within a limited geographical area), through data aggregation, summarization and sampling techniques. The same should happen when selecting a particular time range for the data visualization (e.g., visualizing traffic events occurring in a specific week or month). This means that each user interaction corresponds to a visual query, created from the pan and zoom actions. Linked views are a type of visual query method, but other examples are also relevant. For instance, a visual query method can be used a priori of the final visualization method in order to select what data (spatiotemporal range) to process, analyse and render through such method.

Another consideration should be the Level of Detail (LoD) for different spatial (e.g., city-wide spatial granularity) and temporal (e.g., hourly aggregated data points) granularities, as described in (Silva R. A., 2017). When visualizing large volumes of data in spatial and temporal views, the LoD needed for a good visualization, i.e., the best granularity for a specific spatial or temporal range, changes according to that range. The spatial or temporal ranges are selected by panning and zooming the spatial and temporal views, respectively. For instance, when visualizing a GRTS that spans several years on the temporal view, the higher the range, i.e., the more years to be shown in the visualization, the lower the LoD needed, i.e., the data can be aggregated in a higher granularity (e.g., if the range spans several years, a monthly aggregated granularity should have the necessary LoD, while if the range only spans one month, the granularity should be aggregated by hour, for instance). Hence, zoom and pan interactions performed by users on a particular view (spatial or temporal) dictates how data is aggregated or disaggregated on all other views. Data range and granularity in all views should be updated whenever an user interaction occurs.

Finally, the final visualization method to apply depends on the data set, on the use case and specifically on the processing and analysis methods used and outputs obtained. Regarding the data set, and in the specific case of MobiTrafficBD, the visualization needs to account for the spatiotemporal characteristics of the data set, by using interlinked, interactive spatial (maps), temporal (timelines) and other views, and for the high-volume nature of the data set, by using visual query methods and aggregatable or summarizable views, according to spatiotemporal pan and zoom user interactions. The scenario and the underlying challenges to address also play a key role in the decision about the visualization strategy to apply. For instance, the authors of (Wang S. , et al., 2018) define a correspondence between spatiotemporal analysis methods and visualization of big spatiotemporal data, under the scope of visual analytics.

An example of batch data visualization for exploratory analysis is comprised in the "Big Data harmonization pipeline" use case (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018; Figueiras, et al., 2016; Figueiras, et al., 2016). After the collection, harmonization and storage of MobiTrafficBD, the Big Data Harmonization pipeline provides a visualization dashboard to be used as a preliminary exploratory analysis tool for RITMOs and other stakeholders. View 5 (Appendix A.22, page 281) presents the resulting dashboard for data collected from toll sensors, registering the number of vehicles per vehicle class.

The top view in View 5 presents a box plot chart for several sensors (horizontal axis) and the dispersion of the number of light vehicles passing through the sensors (vertical axis). The second view presents the number of records (vertical axis) per sensor for several sensors (horizontal axis). Then, a table presents several statistics about the harmonized data set, such as minimum, maximum and average values for the number of passing vehicles captured by the sensors, per vehicle class, and the maximum and minimum record dates, i.e., the temporal range for the data set. The bottom view presents a sample corresponding to the first 10.000 records of the data set and shows the aggregated behaviour of the number of light and heavy vehicles for all sensors in the first 10.000 records.

Other views could be developed, such as a map view showing the spatial dispersion for the sensors in the data set. Again, the goal of this tool was to provide to researchers and practitioners an initial overview about the characteristics and ranges of the harmonized data set. They could then use this information to decide about the adequate processing and analysis strategies to apply over the harmonized data set. In this case, the tool was built using traditional Web-based programming languages and tools. Since the visualized data corresponded to a sample of the whole data set, there was no need to use Big Data tools or special communication and visualization strategies.

Another example of visualization of large volumes of batch data is comprised in the "Public transport network status analysis and visualization" use case (Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019). In this scenario, the main goal was to understand the commuting behaviour and the interdependence between different public transportation operators in the city of Lisbon. Particularly, the framework would analyse three types of behaviours: connections, transhipments and pendular movements (see sub-section 6.1.1). The visualization component of the framework was based on the Tableau data visualization desktop application (Tableau Software, 2003), which enables the construction of spatiotemporal visualization dashboards with interactive views, animated views and visual query methods.

The data set used in this scenario was the output of several processing and analysis processes, described in sub-section 6.1.1, and aggregated to obtain daily and weekly trends, focusing on workdays. Hence, prior to visualization, the outputs of the processing and analysis tasks were aggregated and stored in a database, for easy access from the visualization tool. Tableau was chosen as the main visualization tool since it provides built-in data access and visualization tools and techniques and, since the scenario goal was an offline analysis on the large volumes of public transportation data, the response times for rendering and presenting the visualization could be in the order of minutes. This broader response time requirement enables the extraction of larger visual querying results, the addition of more linked views and the visualization of larger data volumes, spanning bigger spatial and temporal ranges.

The visual query methods used enabled the creation of filters in relation to individual public transportation operators or specific stops and stations, besides the common temporal and spatial selection filters. These filters supported the application of data aggregation, sampling and summarization techniques that resulted in data-rich visualizations that do not present the issue of overplotting, for instance. The versatility in terms of visual querying capabilities, resulted in visualization insights that enable real-world changes in everyday public transportation commuting.

Some examples of insightful visualizations obtained in this use case are presented in Figure 6.8 and discussed below. For privacy reasons, the identification of each public transportation operator is hidden. Also, Figure 6.8 only presents examples of visualizations for transhipments and pendular movements because, connections within the same public transportation operator were a least significant insight than the other two. For instance, as expected in any city with a subway network, the operator with more connections, i.e., changing lines within the same operator, was the subway operator.

Figure 6.8 presents three sets of views. These are just a few examples of the overall number of visualizations created for this scenario, but they show the importance of batch data visualization for insight gathering and decision support. The first two views, 1. a) and 1. b), show the percentage of transhipments per destination operator, when the origin operator is Lisbon's subway operator. 1. a) shows transhipment percentage from 0h to 14h and 1. b) presents transhipment percentages from 15h to 23h. This division was made taking into account the daily movements to (in the morning) and from (in the afternoon) Lisbon's city centre. This set of views shows that when people use the subway to go to work in the morning, the perform transhipments to other operators that do "last mile" routes within the city centre (e.g., the sea blue coloured slice corresponds to a tram and bus operator that only operates in Lisbon's city centre) (1. a)), while when people go back home with the subway, in the afternoon/evening, the last public transportation operators used are the ones operating in the suburbs and areas outside the city centre (e.g., the green coloured slice represents a train operator that operates in the south area outside Lisbon's city centre) (1. b)).

This indicator is very important to understand the correlation between entities, enabling the creation of new combined tickets through the analysis of such information, for instance. This trend is also present in the second set of views, 2. a) and 2. b). These views show the average weekly number of transhipments (workdays) per operator (vertical axis), per hour of day (horizontal axis), following the same colour code for each operator. View 2. a) presents the number of transhipments starting operators, while view 2. b) represents the number of transhipments ending operators.

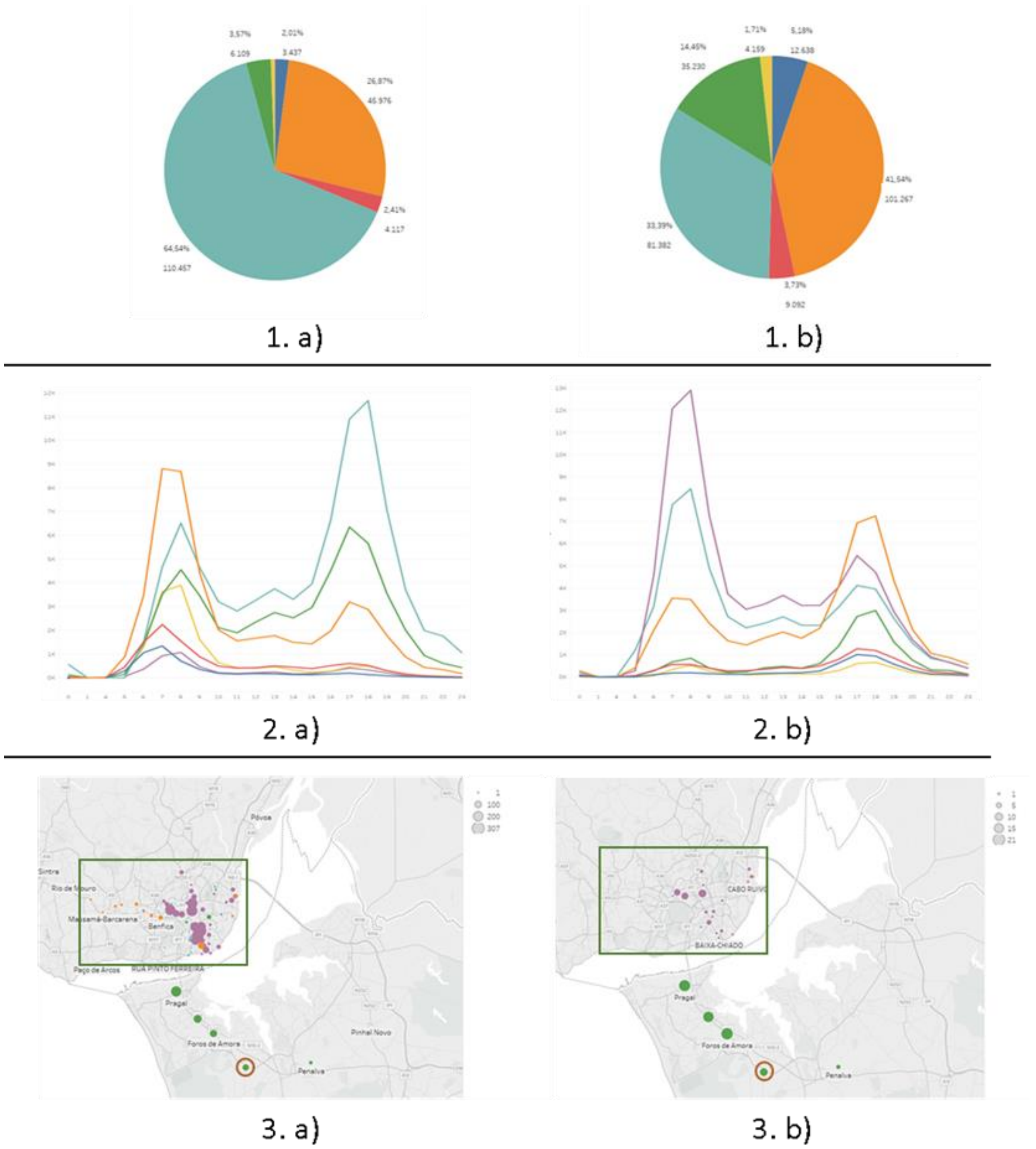


Figure 6.8 — Three sets of views from the public transport network status analysis and visualization framework (Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019). 1. a) Transhipments percentage in the morning (0h - 14h) per destination operator, when the origin operator is an intra-city operator; 1. b) Transhipments percentage in the evening (15h - 23h) per destination operator, when the origin operator is an intra-city operator; 2. a) Hourly sum of weekly (workdays) average transhipments per origin operator; 2. b) Hourly sum of weekly (workdays) average transhipments per destination operator; 3. a) Spatial dispersion and quantity of pendular movements in the morning (0h - 14h) starting from/ending in Fogueteiro station (red circle); 3. b) Spatial dispersion and quantity of pendular movements in the evening (15h - 23h) starting from/ending in Fogueteiro station (red circle);

The main insights taken from this set of views reinforces what was previously stated: intra-city operators (operating inside the city centre) are characterized by being the main starting points for transhipments in the afternoon, evening, when people are commuting back to their homes outside the city, and the main ending points in the morning, when people enter the city and perform their "last mile" journeys to work within the city centre; contrarily, extra-city operators (operating outside the city), are the main transhipment starting points in the morning, when people commute from their homes in the outskirts of the city to the city centre, and the main ending points, when people return home.

Finally, the third set of views analyses the pendular movements from the south area outside the city to Lisbon's city centre, in the morning (3. a)) and in the afternoon (3. b)). Particularly, the pendular movements' origin and destination station in the south area is the Fogueteiro train station, marked with a red circle, whereas Lisbon's city centre is marked with a green rectangle. When pendular movements start from Fogueteiro in the morning, in view 3. a), they occur more frequently and end up in Lisbon's city centre, while if the pendular movements start in the afternoon from the same station, the number of pendular movements is smaller and the main destination are stations in the same geographical area, to the south and outside of Lisbon's city centre. This indicator is important because it can be used to acquire a better understanding of human mobility in cities.

These examples are just a fraction of the possible visualizations with this framework, since these and other visual analyses could be performed for every specific station or stop, for every operator. The total number of data points presented in these visualizations ascended seventeen million, since the framework presented in Figure 6.4 detected 9,304,132 connections, 6.212.659 transhipments and 2.644.569 pendular movements for one month of data.

6.2.2 Streaming Data Visualization Strategies

If the biggest challenge of batch data visualization was the large volumetry of the data to visualize, in the case of streaming data visualization, the main challenge is the speed at which the data streams arrive. This challenge is also extended to the same three aspects described in sub-section 6.2.1: data communication, data preparation and data visualization methods to apply. Although each of these aspects have their own specificities, the main requirement is that the total execution time, for data preparation and access and rendering and presenting the final visualizations, must not exceed the stream arrival time or the sliding window time interval discussed in sub-section 6.1.2.

When visualizing data in real-time, which is a regular requirement for streaming data visualization use cases, the data communication between the stream processing engine and the visualization tool is a crucial factor, since depending on the speed of the stream, traditional communication protocols, such as the Hypertext Transfer Protocol (HTTP) (Fielding, et al., 1985) for communication over the Internet, may not be able to cope with the extremely fast arrival of data objects. In this case, publish-subscribe mechanisms and other fast message delivery strategies may be the adequate choice. One such case, which can be used as a data stream communication protocol over the Internet, and is complementary to HTTP, is the WebSocket protocol (Internet Engineering Task Force (IETF), 2011).

The HTTP protocol requires a request and a response for every communicated message between a server and a private computer. This means that, for a data stream, communication of each individual data object in the stream needs to be requested to the server and the server then sends the data object message as the response. This creates high latency in terms of data stream communication. In contrast, the WebSocket protocol resembles a publish-subscribe mechanism in which the user's machine opens a direct communication channel to the server once, and the server can then send the data stream to the user's machine without the need for a request for each single message in the stream. Other option is to use message queueing tools, such as Apache Kafka, to feed data to the visualization tools.

Regarding data preparation, the speed at which data is communicated also poses challenges on how the data is prepared, aggregated and summarized prior to visualization. Since data streams normally arrive at minute, second or even millisecond granularities, the time to process, analyse and prepare visualizations is short. Again, this time will always depend on the stream processing model adopted: stream or sliding window. The time for processing, analysing and presenting the data through visualization methods is shortest in the stream model, since each data object in the stream is individually processed and sent to visualization. In contrast, the processing and analysis time is extended in the sliding window model according to the sliding window's time range.

Streaming MobiTrafficBD, particularly in the case of GRTS, are often complex data sets, i.e., the Big Data and spatiotemporal nature of MobiTrafficBD introduces complexity to these data sets in the form of the spatial and temporal dimensions, the interrelations between these two dimensions and the measured values and the speed at which these complex data sets must be processed. Such complexity entails that the processing time for streaming MobiTrafficBD must be extended by performing data preparation and aggregation tasks prior to sending data streams for processing, i.e., data gathered in sensors can be aggregated in minute-range windows in an intermediate hardware system (e.g., a gateway) and then sent in a data stream to a stream processing framework. This is a common procedure, as represented by the traffic sensor data sets used in the example use cases: these data sets recurrently present 5-minute or 10-minute granularities, which result from the aggregation of all sensor readings in each 5-minute interval.

Hence, visual analytics for extremely fast MobiTrafficBD streams (in the order of milliseconds or less to seconds) should revolve around real-time, raw data visualization without any underlying processing or analysis, or performing very simple analyses, supported by sublinear time or similar processing and analysis methods. For slower MobiTrafficBD streams (in the order of tens of seconds to minutes), more accurate, complex visual analytics methods may be used, such as clustering, classification, etc. For MobiTrafficBD streams that do not have a fixed time between data objects, which is the main case for ST Events or for the outputs of complex event processing mechanisms, both simple and complex processing strategies may be used, as long as the chosen strategy is adequate for the shortest time possible between objects in the stream. For instance, if the ST Event data corresponds to traffic events, such as accident, traffic jams, road closures, etc. then the minimum time between two events in the stream should be in the order of minutes, while if the ST Event data corresponds to public transportation ticket validations, the minimum time between two events in the stream may be in the order of milliseconds.

Finally, regarding the methods and strategies for data stream visualization, the most common strategy to visualize data streams is through animated views. These animated views are realized through the overlap of consecutive snapshot views, creating a "film" of snapshots, each of which pertains to a specific message or group of messages, depending if the timeframe for the visualization is based on a stream or sliding window model. These animated views may be based on different visualization methods, such as maps, charts or any other method. The method's choice depends mainly on the type of processing and analysis tasks performed prior to the application of the visualization method, as described in (Wang S. , et al., 2018).

The "Real-time traffic flow analysis" example (Figueiras, et al., 2018) is a scenario that comprises all of the above considerations, such as the application of the WebSocket protocol for data stream communication, fast data aggregation and streaming data visualization supported by animated views. The visualization goal for the stream processing framework presented in Figure 6.5 was to provide animated views in real-time for visualization of extremely fast streaming data with different temporal granularities. As described in sub-section 6.1.2, in the first topology setting the goal was to aggregate lane-specific data objects in the stream, captured by traffic sensors in Slovenian roads and highways, into direction-specific data objects, while in the second topology, data was also aggregated in an hour-based sliding window, each window containing six data objects (each data object has a 10-minute temporal granularity).

Since the framework's objective was to process, analyse and render visualizations for extremely fast data, an extremely fast data stream was simulated by fast feeding data objects to a RabbitMQ instance, which could produce a data stream of 8.8 data objects per second. The stream was then passed to both Apache Storm topologies, already described in sub-section 6.1.2, and the outputs of these topologies had to be rendered in meaningful real-time visualizations. To be able to communicate the data stream to the Web-based visualization component, several streaming tools were integrated. The first step was to add one more Storm bolt to the workflow of Figure 6.6, as presented in Figure 6.9. The JmsBolt's goal is to write the outputs of both topologies, gathered from the AggregationBolt, into an Apache ActiveMQ (The Apache Software Foundation, 2004) queue. ActiveMQ is similar to RabbitMQ but is based on the Java language and integrates a Java Message Service (JMS) (Oracle Corporation, 1998) instantiation, which is a message-oriented middleware API.

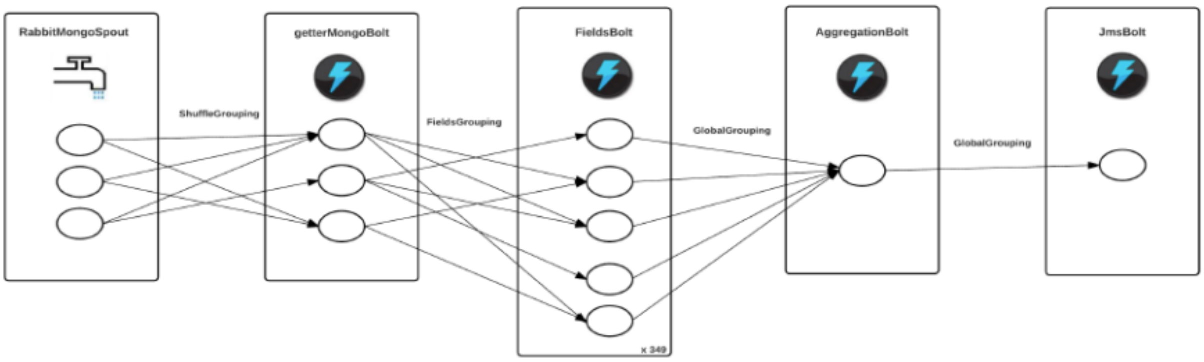


Figure 6.9 — Apache Storm Topology updated with a stream communication bolt, for streaming data visualization

Second, Apache Camel (The Apache Software Foundation, 2007) was integrated to handle the connection between JMS and WebSocket message communication. Apache Camel is a routing rules creation and mediation system that converts messages from and to several message transport models, such as HTTP, ActiveMQ and JMS, among others. Finally, a WebSocket client was developed in JavaScript to connect the users' machines to the framework. This suite of tools and technologies enabled the rendering of animated views for fast data streams. Figure 6.10 and Figure 6.11 present some snapshots of animated views created by the framework.

In Figure 6.10, the top map view (1) is an animated view in which road occupancies for several traffic sensors in Slovenia are represented by animated three-dimensional vertical bars. The height of the bars changes in real-time depending on the information regarding the road occupancy. In the same way, the bottom map view (2) presents real-time variations in vehicle speeds for several traffic sensors as circles, in which real-time speed values are represented by varying the colour intensity and diameter of the circle (the bigger the speed, the bigger the circle's diameter and stronger the colour intensity).

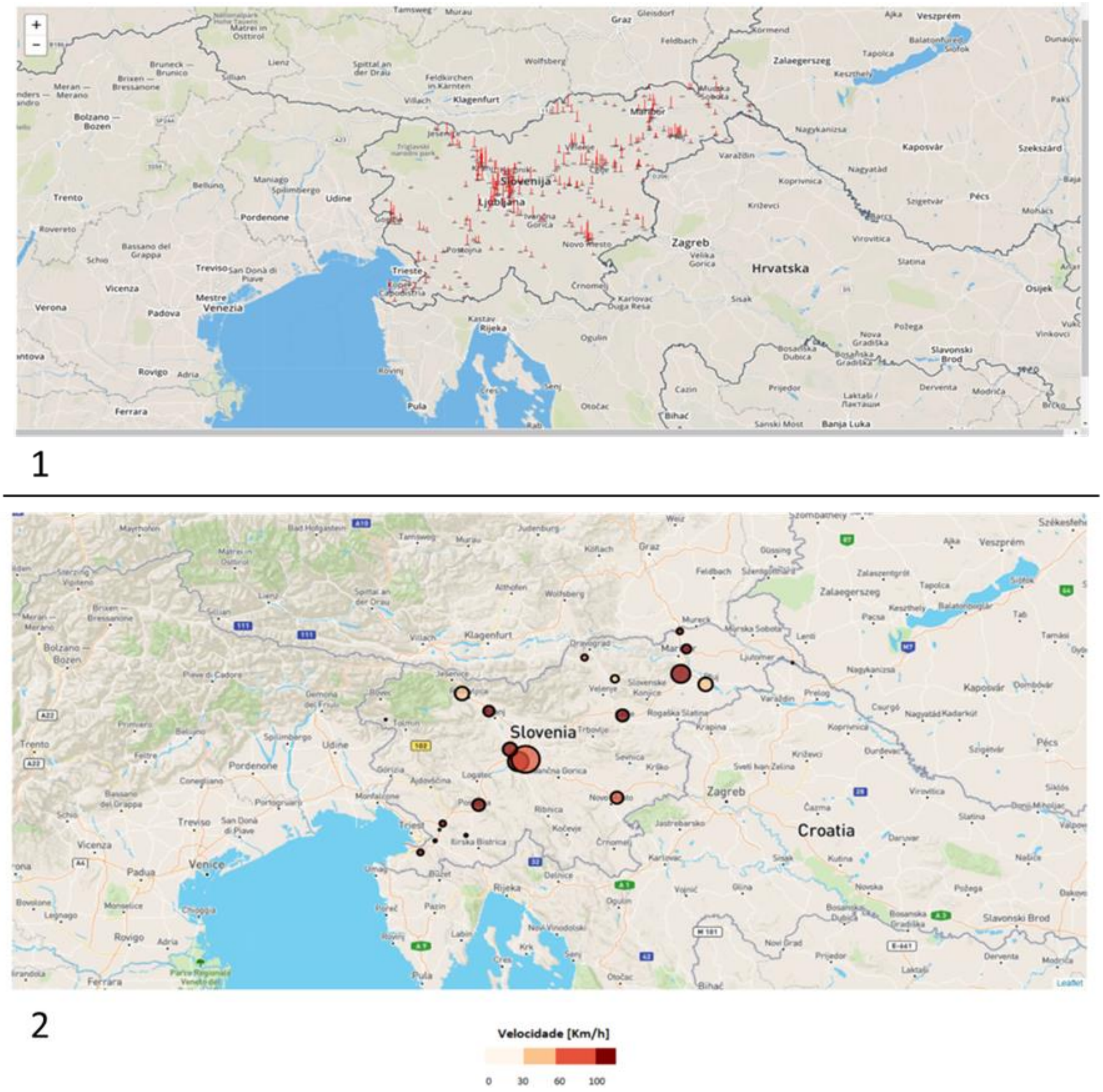


Figure 6.10 — Snapshots of map-based animated views for several Slovenian traffic sensors: 1. Road occupancy represented by three-dimensional bars with varying heights; 2. Average vehicle speed (Km/h) represented by coloured circles, with varying colour intensity and circle diameter

In Figure 6.11, two animated bar chart visual approaches were used to visualize the real-time fluctuations of vehicle speed and road occupancy values. View A represents the hourly-aggregated outputs of the second topology for average speeds (top) and average road occupancies (bottom), whereas View B presents the outputs for the first topology, providing real-time variations for vehicle speed (left) and road occupancies (right) in each road direction. These views enable visual comparison analyses between several sensors simultaneously, whether focusing on a sliding window model to slow down the animation (View A) or by comparing several sensor readings in both directions.

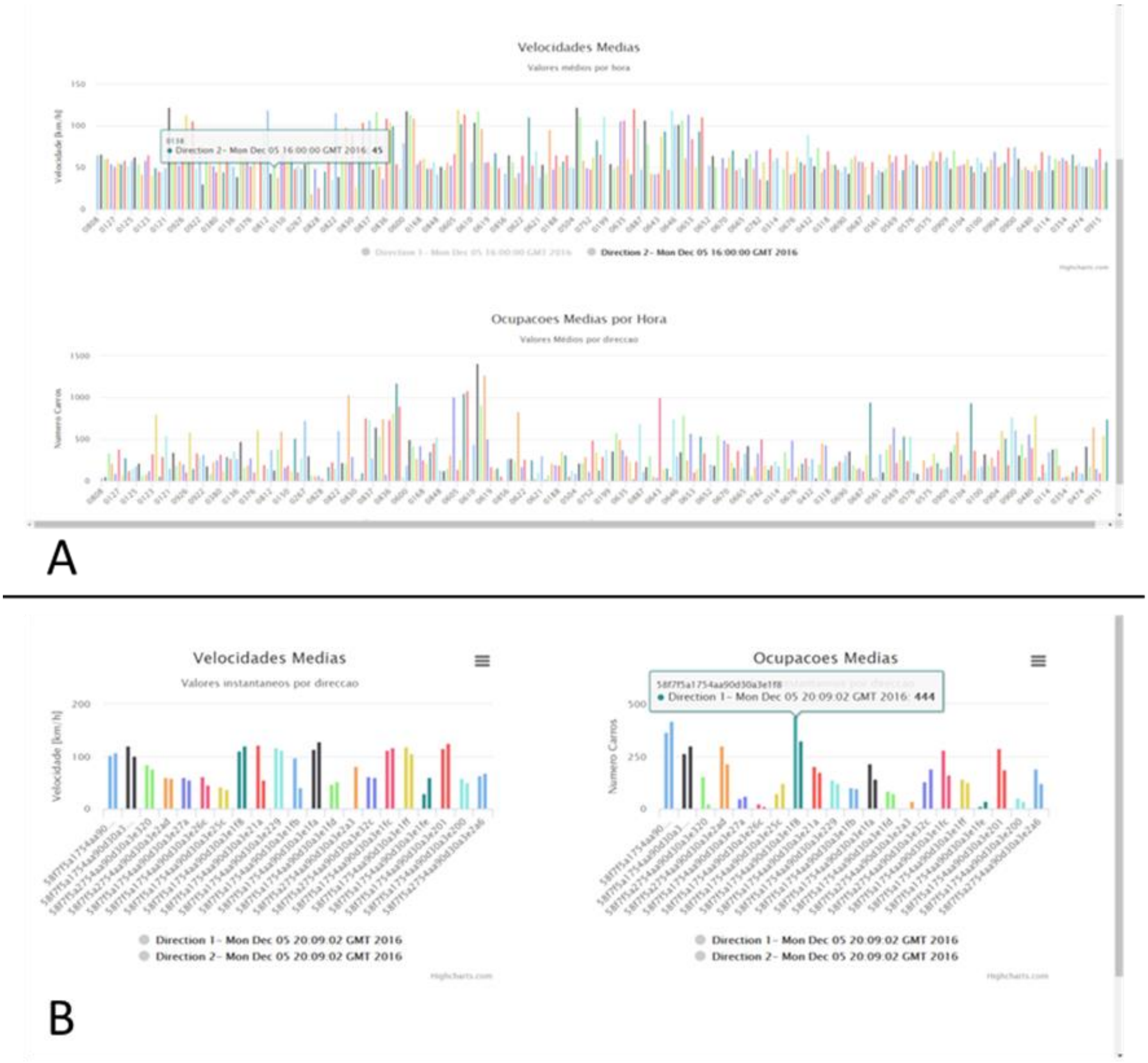

Velocidades Medias
Valores médios por hora
Velocidade [km/h]
150
100
50
0
0134
● Direction 2 - Mon Dec 05 16:00:00 GMT 2016: 45
Direction 1 - Mon Dec 05 16:00:00 GMT 2016
Direction 2 - Mon Dec 05 16:00:00 GMT 2016
Highcharts.com
Ocupacoes Medias por Hora
Valores Médios por direccao
Numero Carros
1500
1000
500
0
A
Velocidades Medias
Valores instantaneos por direccao
Velocidade [km/h]
200
100
0
Direction 1 - Mon Dec 05 20:09:02 GMT 2016
Direction 2 - Mon Dec 05 20:09:02 GMT 2016
Highcharts.com
Ocupacoes Medias
587f5a1754aa90d30a3e1f8
● Direction 1 - Mon Dec 05 20:09:02 GMT 2016: 444
Numero Carros
500
250
0
Direction 1 - Mon Dec 05 20:09:02 GMT 2016
Direction 2 - Mon Dec 05 20:09:02 GMT 2016
Highcharts.com
B

Figure 6.11 — Snapshots of real-time animated bar charts for real-time visualization of speed and occupancy: A. Real-time visualization of hourly aggregated data (second topology) for speed (top) and occupancy (bottom); B. Real-time visualization of direction-specific (first topology) speed (left) and occupancy (right).

6.3 Guidelines and Best Practices

This chapter served as a prescriptive, methodological approach for batch and streaming MobiTrafficBD processing, analytics and Visual Analytics processes. Although Data Analytics and Visualization methods and strategies can vary greatly according to the specific data and use case at hand, some general considerations were discussed towards the choice of tools, strategies and techniques to be integrated in the design and development of any MobyTrafficBD Framework. The general methodology and discussion about considerations, strategies and tools was further reinforced with the example use cases, which showcase some of the discussed strategies and tools. The following guidelines and best practices serve as a summary of the main recommendations made in this chapter:

- **<u>MobiTrafficBD Analytics</u>**
    1. The three main considerations when choosing the adequate tools for MobiTrafficBD processing and analysis should be: 1. The data at hand; 2. The use case at hand or analysis to be performed; 3. The performance and scalability of the distributed environment in which the MobiTrafficBD Framework will be deployed.
    2. When processing and analysing large volumes of MobiTrafficBD, several strategies are recommended, such as distributed data processing strategies or efficient data access strategies (data indexing in databases, data sampling for data exploration tasks, etc.).
    3. In the case of the application of distributed processing and analysis technologies, the choice of the data processing and analysis methods and algorithms must take into account the performance of the distributed environment.
    4. When processing and analysing large volumes of MobiTrafficBD, it is highly recommended the definition of spatiotemporal and unique identifier indexes in databases, for easier data access and querying.
    5. When performing exploratory analyses over large batches of MobiTrafficBD, it is highly recommended the application of data sampling techniques, to extract underlying, preliminary insights, trends and relationships that can be extrapolated to the entire data set.
    6. When using data sampling techniques, the sample size choice is always dependent on the type and comprehensiveness of the analysis and, ultimately, of insights that the researcher or practitioner wants to achieve.
    7. When processing and analysing fast streams of MobiTrafficBD, the decision between sublinear time and similar methods and online versions of batch data-driven algorithms or between a stream processing model or a sliding window one, must take into account data stream characteristics, such as the rate of reception of each data object, and requirements for the analysis to be performed over the stream, as in the case of execution time and accuracy of the method.
    8. Hybrid methods for both batch and streaming MobiTrafficBD processing and analysis, are often based on micro-batches and the sliding window model, with the sliding window size varying depending on the nature of the data and the use case at hand.
- **<u>MobiTrafficBD Visualization</u>**

1. Data aggregation, summarization and sampling are recommended strategies to tackle the challenges of visualization of large volumes of MobiTrafficBD, such as the overplotting challenge. Other existing strategies often use data aggregation, summarization and sampling as the basis for tackling data volume-related challenges in visualization, such as in the case of visual querying methods or linked views.
2. In the case of an exploratory analysis of large volumes of MobiTrafficBD, it is recommended for data sampling techniques to be applied before data communication to visualization tools, to reduce the volume of communicated data.
3. When using data samples for exploratory MobiTrafficBD visual analyses, samples must contain insights and trends for multiple temporal granularities and enough spatial information to characterize the spatial dispersion in the original data set.
4. When using linked views for MobiTrafficBD visualization, zoom and pan interactions performed by users on a particular view (spatial or temporal) dictates how data is aggregated or summarized on all other views. Data range and granularity in all views should be updated whenever a user interaction occurs.
5. When visualizing data in real-time, which is a regular requirement for streaming MobiTraffiBD visualization use cases, the data communication between the stream processing engine and the visualization tool is a crucial factor, since depending on the speed of the stream, traditional communication protocols may not be able to cope with the extremely fast arrival of data objects.
6. The complexity of MobiTrafficBD entails that the processing time for streaming MobiTrafficBD must be extended by performing data preparation and aggregation tasks prior to sending data streams for processing.
7. Visual analytics for extremely fast MobiTrafficBD streams (in the order of milliseconds or less to seconds) should revolve around real-time, raw data visualization without any underlying processing or analysis, or performing very simple analyses, supported by sublinear time or similar processing and analysis methods.
8. For slower MobiTrafficBD streams (in the order of tens of seconds to minutes), more accurate, complex visual analytics methods may be used, such as clustering, classification, etc.
9. For MobiTrafficBD streams that do not have a fixed time between data objects, which is the main case for ST Events or for the outputs of complex event processing mechanisms, both simple and complex processing strategies may be used, as long as the chosen strategy is adequate for the shortest time possible between objects in the stream.

7

CONCLUSIONS

Mobility- and Traffic-related Big Spatiotemporal Data processing, analysis and lifecycle management is already an important research area and is expected to grow even further along with new advances in the ICT domain, and particularly on the ITS field of study, such as autonomous and electrical vehicles or "smart" road infrastructure, traffic management and public transportation, just to name a few research areas under evident growth. Such growth must be complemented with standard or *de facto* methodologies, tools, techniques, methods and tools that serve as a basis for researchers and practitioners to push the evolution of ITS and, specifically, to support Mobility and Traffic-related stakeholders, such as RITMOs, seeking to manage, leverage and capitalize on extremely large and extremely fast MobiTrafficBD sources available, in their decision-making processes.

The work presented in this document strives to be a step forward towards this end, by proposing a data-driven, prescriptive methodology that supports the design, creation and deployment of any framework that tackles MobiTrafficBD lifecycle management and/or processing and analysis challenges. Such challenges reflect the Big Data and Spatiotemporal characteristics that define MobiTrafficBD:

- Parallel/distributed storage and processing of large amounts of spatiotemporal data, particularly GRTS and ST Events.
- Scalability (accommodate more data, users, and analyses) and elasticity, using distributed hardware to lower the costs of implementation and maintenance and optimizing the overall performance of MobiTrafficBD lifecycle management, processing and analysis processes.
- Flexible storage that can cope with Big Data nature and spatiotemporal characteristics of MobiTrafficBD.
- Real-time capabilities for MobiTrafficBD (stream processing, low-latency and high-frequency updates), even in the presence of complex, highly dimensional data sets.

- MobiTraffiBD interoperability, through the use of standard or *de facto* methodologies, models and methods, in and integrated environment with multiple technologies.
- Mixed and complex analytics for MobiTrafficBD (e.g., *ad hoc* or exploratory analysis, data mining, text mining, statistics, machine learning, reporting, decision-making support, visual analytics, advanced visualizations, and linked and animated views).

Considering the state-of-the-art in MobiTrafficBD, it can be concluded that there is no common approach for the design, development and deployment of MobiTrafficBD Frameworks. Furthermore, there are panoplies of Big Data- and spatiotemporal data-driven considerations, requirements, tools, technologies, models or methods, which generate barriers in the design and implementation of MobiTrafficBD Frameworks, whichever the individual frameworks' requirements are. Current reference and logical architectures for both Big Data and spatiotemporal data only solve their own part of the barriers, but ambiguity regarding the adequacy of MobiTrafficBD lifecycle management, processing and analysis techniques and technologies according to the context and scenario at hand, still prevails.

Due to the ever-growing and increasingly faster volumes of Big Data, a shift from a use case-driven paradigm to a data-driven one has become pervasive across all fields that seek to capitalize on this Big Data explosion, such as in the case of ITS. Traditional spatiotemporal data processing and analysis systems are based on use case-driven approaches, in which the selection of strategies, tools and methods is enforced by the scenario itself and its characteristics and requirements. While scenario requirements and contextual characteristics were the main concern, current Big Data-driven spatiotemporal data processing and analysis systems, such as MobiTrafficBD Frameworks, are turning their focus to data-driven approaches. For instance, if, in traditional systems, the focus was to select or develop the most adequate method for processing and analyzing spatiotemporal data, depending on the use case's goals and requirements, nowadays, the main concerns for researchers and practitioners are the choice of adequate Big Data tools, techniques and models to better handle Big Spatiotemporal Data and how to optimize existing spatiotemporal data modelling and analysis processes to cope with the Big Data and spatiotemporal characteristics of data, as in the case of MobiTrafficBD.

Thus, until now, there is no structured and general-purpose approach describing and guiding researchers, practitioners and other interested stakeholders, on how to design and implement a MobiTrafficBD Framework, independently of the specific MobiTrafficBD or the scenario at hand, and with adequately evaluated models (representations of logical and technological components and data flows), methods (strategies, guidelines and best-practices), and instantiations (e.g., demonstration cases through prototyping and benchmarking). This scientific and technical gap is the main motivation for the presented work, since, in the author's modest opinion, existing logical architectures, guidelines and best practices, in the specific contexts of Big Data and spatiotemporal data, did not provide an integrated, general-purpose, detailed and evaluated approach that practitioners could rely on to design and implement MobiTrafficBD Frameworks according to their characteristics.

Further, a clear gap between "this is what a MobiTrafficBD should be", i.e., the presentation of individual MobiTrafficBD Framework instances in the literature, and "this is how you design and implement a MobiTrafficBD Framework" motivated the proposal of this approach, an integrated, detailed, general-purpose, data-driven and prescriptive contribution to design and implement MobiTrafficBD Frameworks, using strategies, methodologies, standards, models and methods that were adequately evaluated through different demonstration scenarios.

Nevertheless, the author recognizes the possibly large ambition of the proposed approach, but also considers that the approach's main goal was achieved, since researchers and practitioners now have a set of artifacts that can be used to design, build and deploy any MobiTrafficBD Framework and to promote future research endeavours in this field, as techniques and technologies evolve and new strategies emerge. The following sections describe the staged work and achieved results, the main contributions to current state-of-the art knowledge and possible future work pathways.

7.1 Staged Work and Achieved Results

Considering the research goal and objectives of this doctoral thesis, one can state that the staged work and achieved results are divided into six main work fronts, namely the proposed approach for MobiTrafficBD Frameworks and the five example use cases: Big Data harmonization pipeline (Figueiras, Guerreiro, Silva, Costa, & Jardim-Gonçalves, 2018; Figueiras, et al., 2016; Figueiras, et al., 2016); CEP for traffic event detection (Figueiras, Antunes, Guerreiro, Costa, & Jardim-Gonçalves, 2018; Antunes H. A., 2017); Real-time traffic flow analysis (Figueiras, et al., 2018; Rosa, 2017); Public transport network status analysis and visualization (Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019; Antunes, Figueiras, Costa, Teixeira, & Jardim-Gonçalves, 2019); Twitter mining for traffic event detection (Gutiérrez, Figueiras, Oliveira, Costa, & Jardim-Gonçalves, 2015). These six work fronts encompassed several activities of the DSRM research methodology for Information Systems (Figure 1.1), including design and development, demonstration, and evaluation.

The creation of the proposed approach, consisting of the prescriptive data-driven methodology for designing, developing and deploying MobiTrafficBD Frameworks, comprised the definition of common models, design considerations, requirements, strategies, guidelines and best-practices to design and develop a MobiTrafficBD Framework. These artifacts went through a continuous refinement process, in which example use cases supported not only the evaluation and validation of the approach, but also the iterative refinement of the overall methodology. Finished this doctoral thesis, the following artifacts can be highlighted:

1. A set of generic characteristics, design considerations and functional and non-functional requirements that are common to any MobiTrafficBD Framework (Section 3.1), serving as a starting point for the design of such frameworks.
2. A model of logical components and data flows (Figure 3.2, Section 3.2), which illustrates and describes the components that should be considered in the design and development of a MobiTrafficBD Framework, how they interoperate and how data flows through the framework. The model comprises modular components related to Data Management (data collection, harmonization, cleaning and storage), Data Processing (data access and querying, aggregation, fusion and integration), Data Analytics (Machine Learning, Deep Learning, Data Mining), Data Visualization and User Interaction (data sharing, user interaction and Visual Analytics) as well as transversal Communication, Infrastructure, Orchestration, Security and Privacy components.
3. A technological infrastructure model (Section 3.3), resulting from an extensive research and development to identify and test several technologies suitable to instantiate the different components proposed in the model of logical components and data flows. The technological infrastructure model presents several alternatives that can be used to implement a MobiTrafficBD Framework, including data collection and transformation and ETL pipelines, storage, querying and access, data processing and Data Analytics, and data visualization and sharing technologies.
4. A set of general, methodological guidelines and best practices concerning logical components and data flows and technological infrastructure models (Section 3.4), such as on how to deploy MobiTrafficBD Frameworks on cloud environments or on-premises.
5. Sets of guidelines and best practices about data collection, modelling, harmonization and storage (Section 5.3), and data processing, Data analytics and visualization (Section 6.3) of MobiTrafficBD, along with real-world example use cases of the definition and application of such guidelines and best practices.

Taking these contributions into consideration, the proposed approach can be used by practitioners and researchers as a structured, integrated, and general-purpose approach that can be prescribed to solve several real-world MobiTrafficBD challenges, aiming to support MobiTrafficBD management, processing and analytics on Big Data environments while taking advantage of MobiTrafficBD characteristics. Furthermore, the approach was evaluated and refined across several demonstration scenarios applied in this doctoral thesis, which provides a solid scientific and technical basis. Particularly, one may consider that there is a symbiotic relationship between the proposed prescriptive methodology and the demonstration scenarios: If, on one hand, the artifacts contained in the proposed methodology were defined during the research work performed by the author throughout the years, in the various projects from which the example scenarios were extracted, on the other, these artifacts were already applied in the design, development and deployment of each individual framework that served as solution to the example scenarios. Some frameworks that fit as examples of the possible application of the proposed methodology are the SMASH framework (Wu, Morandini, & Sinnott, 2015) or the OPTIMUM project's OODA framework (Figueiras, et al., 2019). Hence, the proposed hypothesis that answers the initial research question is affirmative and is proposed as a proper thesis.

7.2 Contributions to the State-of-the-art

As stated in Section 7.1, to the best of the author's knowledge, the conclusion is that the proposed methodological approach represents a relevant contribution to the scientific and technical community, by providing a set of artifacts for MobiTrafficBD Framework design and implementation that not only paves the way for future research but can also support researchers and practitioners build these complex systems, which otherwise would typically fall into a use case and *ad hoc* driven process.

The models, strategies and guidelines proposed in this work were scientifically backed up by a DSRM for Information Systems research process using five demonstration cases that allowed the evaluation of the methodological approach mainly in terms of the Big and spatiotemporal characteristics of MobiTrafficBD, suitability, effectiveness, complexity, latency, and, when applicable, resource considerations. Consequently, this approach successfully fulfills the scientific gap previously identified, i.e., the lack of a prescriptive and integrated methodological contribution for the design and implementation of MobiTrafficBD Frameworks, with adequately evaluated models, guidelines and best practices.

Nevertheless, this work was mainly supported by previously existing contributions, reason why this approach is built upon some general constructs and guidelines in the areas of Big Data and Spatiotemporal Data, provided by the BDVA-RM (Big Data Value Association, 2020), the NIST Big Data Architecture (NBD-PWG, 2015), the Big Data Processing Flow proposed by (Krishnan, 2013), the guidelines for quality Advanced Traveller Information Systems data (America's Advanced Traveller Information Systems Committee, 2000), the Spatiotemporal

Aspects of Big Data proposed in (Karim, Soomro, & Burney, 2018) or the Big Data Warehousing guidelines from (Costa C. F., 2019), just to name a few.

Further, this work's contribution to the state-of-the-art in MobiTrafficBD Frameworks was only possible due to previously explored paths and the relevant contributions of several related works including the vast amounts of scientific and technical works related to Big Data, Spatiotemporal Data, Data Processing and Analytics, Visualization, Data Harmonization and Interoperability, Data Storage, Data Collection or Knowledge Discovery and Data Mining, among others fields, ranging several academic and professional areas, whose absence would otherwise make unfeasible the advancements regarding MobiTrafficBD Frameworks.

Science and technology mainly owe their progress to disruptive discoveries, but these are more than often supported by solid scientific and technical foundations defined by research works that formalize previous knowledge in a methodological way. This work strived to propose such a foundation for the design and implementation of MobiTrafficBD Frameworks, which was relatively difficult to accomplish, considering the lack of maturity when it comes to the integration of Big and Spatiotemporal data in Mobility and Traffic contexts.

Focusing on the communication activity of the DSRM for Information Systems methodology, several scientific publications related to this research work have been positively reviewed and accepted by the scientific community, which allowed the dissemination of several results. Moreover, technical content related to the work proposed here was also presented in practice-oriented forums and released as a book chapter. The following publications (summarized in Table 7.1) represent the communication activity associated with this doctoral thesis:

- Journal Publications:
    - Figueiras, P.; Gonçalves, D.; Costa, R.; Guerreiro, G.; Georgakis, P.; Jardim-Gonçalves, R. "Novel Big Data-supported dynamic toll charging system: Impact assessment on Portugal's shadow toll highways". In Computers & Industrial Engineering, 135, September 2019, 476-491, 2019. DOI: 10.1016/j.cie.2019.06.043
- Conference Proceedings:
    - Costa, R.; Figueiras, P.; Oliveira, P.; Jardim-Gonçalves, R. "Understanding Personal Mobility Patterns for Proactive Recommendations". In OTM 2015: On the Move to Meaningful Internet Systems: OTM 2015 Workshops, Rhodes, Greece, 2015. DOI: 10.1007/978-3-319-26138-6_16
    - Figueiras, P.; Guerreiro, G.; Costa, R.; Bradesko, L.; Stojanovic, N.; Jardim-Gonçalves, R. "Big Data Harmonization for Intelligent Mobility: a Dynamic Toll-charging Scenario". In OTM 2016: On the Move to Meaningful Internet

Systems: OTM 2016 Workshops, Rhodes, Greece, 2016. DOI: 10.1007/978-3-319-55961-2_8

- Figueiras, P.; Silva, R.; Ramos, A.; Guerreiro, G.; Costa, R.; Jardim-Gonçalves, R. “Big Data Processing and Storage Framework for ITS: A Case Study on Dynamic Tolling”. In ASME 2016 International Mechanical Engineering Congress and Exposition (IMECE), Phoenix, Arizona, USA, 2016. DOI: 10.1115/IMECE2016-68069
- Costa, R.; Jardim-Gonçalves, R.; Figueiras, P.; Forcolin, M.; Jermol, M.; Stevens, R. “Smart Cargo for Multimodal Freight Transport: When “Cloud” becomes ‘Fog’”. In 8th IFAC Conference on Manufacturing Modelling, Management and Control (MIM), Troyes, France, 2016. DOI: 10.1016/j.ifacol.2016.07.561
- Guerreiro, G.; Figueiras, P.; Silva, R.; Costa, R.; Jardim-Gonçalves, R. “An architecture for big data processing on intelligent transportation systems. An application scenario on highway traffic flows”. In IEEE 8th International Conference on Intelligent Systems (IS), Sofia, Bulgaria, 2016. DOI: 10.1109/IS.2016.7737393
- Costa, R.; Figueiras, P.; Guerreiro, G.; Bradesko, L.; Stojanovic, N.; Georgakis, P.; Bothos, E.; Magoutas, B. “Proactive recommendations for Intelligent Mobility - An approach based on real-time big data processing”. In I-ESA 2016 - Interoperability for Enterprise Systems and Applications, Guimarães, Portugal, 2016.
- Figueiras, P.; Costa, R.; Guerreiro, G.; Antunes, H.; Rosa, A.; Jardim-Gonçalves, R. “User interface support for a big ETL data processing pipeline an application scenario on highway toll charging models”. In 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC), Madeira, Portugal, 2017. DOI: 10.1109/ICE.2017.8280052
- Figueiras, P.; Herga, Z.; Guerreiro, G.; Rosa, A.; Costa, R.; Jardim-Gonçalves, R. “Real-Time Monitoring of Road Traffic Using Data Stream Mining”. In 2018 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC), Stuttgart, Germany, 2018. DOI: 10.1109/ICE.2018.8436271
- Figueiras, P.; Antunes, H.; Guerreiro, G.; Costa, R.; Jardim-Gonçalves, R. “Visualisation and Detection of Road Traffic Events Using Complex Event Processing”. In ASME 2018 International Mechanical Engineering Congress and Exposition (IMECE), Pittsburgh, Pennsylvania, USA, 2018. DOI: 10.1115/IMECE2018-87909
- Antunes, H.; Figueiras, P.; Costa, R.; Teixeira, J.; Jardim-Gonçalves, R. “Analyzing Public Transport data through the use of Big Data technologies for urban

mobility". In 2019 International Young Engineers Forum (YEF-ECE), Caparica, Portugal, 2019. DOI: 10.1109/YEF-ECE.2019.8740816

- Book Chapters:
  - Figueiras, P.; Guerreiro, G.; Silva, R.; Costa, R.; Jardim-Gonçalves, R. "Data Processing and Harmonization for Intelligent Transportation Systems: An Application Scenario on Highway Traffic Flows". In Learning Systems: From Theory to Practice, Springer International Publishing AG, Cham, Switzerland, 2018. DOI: 10.1007/978-3-319-75181-8_14

Table 7.1 — Scientific Publications

Type	Number	Detail
Scimago Q1 Journals	1 publication	(1) Journal of Computers & Industrial Engineering
Core b conferences or similar	4 publications	(2) ASME International Mechanical Engineering Congress and Exposition (1) IFAC Conference on Manufacturing Modelling, Management and Control (1) IEEE International Conference on Intelligent Systems
Book chapters	1 publication	(1) Learning Systems: From Theory to Practice, Springer
Other conferences of international scientific circulation and review	6 publications	(2) On the Move to Meaningful Internet Systems (2) International Conference on Engineering, Technology and Innovation (1) Interoperability for Enterprise Systems and Applications (1) International Young Engineers Forum

7.3 Prospects for Future Work

Regarding future work, there is space for further exploration and contributions, namely in terms of the encompassing of new data sources outside the scope of GRTS and ST Events, the emergence of new technologies, tools and techniques for both Big Data and spatiotemporal data analysis and lifecycle management and, ultimately, the evolution of new paradigms that may enable the extension of the proposed approach with new components, data flows and inherent guidelines and best practices to be adopted in the near future.

Although the proposed approach only contemplates GRTS and ST Events, it may encompass other MobiTrafficBD types or even data outside the realm of ITS, although new, specific guidelines and best practices may apply. Nevertheless, the logical components and data flows model (Figure 3.2) and the technological infrastructure model (Figure 3.6) were purposedly designed to be generic enough to be applied in other contexts and to different data types. Specific guidelines and best practices should be presented for these new contexts, but both models could be directly applied. The main data types that can be encompassed correspond to spatially and temporally dynamic data types mentioned in sub-section 2.2.1(Figure 2.1), such as trajectories or moving points, and will enable new types of analyses through the fusion of different data sets from specific types. One possible use case could be the analysis of the "heartbeat" of urban mobility based on public transportation trajectories (e.g., taxis), GRTS derived from traffic sensors, ST events representing different events in the urban fabric (e.g., traffic-related, such as accidents and traffic jams, or social, such as public events that may have consequences for urban traffic) and meteorological moving point data (e.g., mobile weather stations scattered throughout the city). The addition of new data types would also account for new data-driven challenges and the necessary guidelines and best practices to mitigate them.

As in the case of new data types, new technologies and tools may also be encompassed in the proposed approach. The technological infrastructure model presents some example technologies and tools that can be combined and used in a framework to tackle the lifecycle and analysis of MobiTrafficBD, but it is not limitative, since other, present or future, technologies, tools and associated methods may be applied. Even so, and since the several layers and components of the model of Figure 3.2 are based on generic Reference Architectures, it is envisaged that new technologies that fall under the scope of such layers and components can be easily applied under the proposed approach. Whether or not the addition of these new technologies presents new challenges to the overall logical components and data flows model and to the technological infrastructure model, it may also impose the addition of new guidelines and best practices for the adoption of such technologies in MobiTrafficBD-driven scenarios.

Moreover, the evolution of both data and technology and the emergence of new paradigms in the IT sector, offer an opportunity for the proposed methodology to be extended with self-awareness mechanisms, so as to monitor the gap between existing models, guidelines and best practices and future paradigms on both technology and data. This extension could be fashioned in two distinct flavors: The self-awareness mechanisms could be applied to the methodology itself or to a MobiTrafficBD framework that is designed using the methodology.

In the first case, a formalization of the methodology would be in place, encompassing the various models, guidelines and best practices, and the recommended technologies, tools and data flows. This formalization would then serve as a baseline for further recommendations of a self-monitoring system that would advise on the changes needed for the methodology to be updated with new technologies, data types and flows. This would allow the methodology to be aware of new paradigms or evolutions and propose the update of both the models and the guidelines and best practices. In the second case, the self-awareness mechanisms would be added as a component in the orchestration layer of a MobiTrafficBD, which should be transversal to all components in the framework, enabling the framework's self-monitoring and awareness about the main paradigm shifts in terms of data types and flows or emerging technologies, according to the framework's use case requirements. These self-awareness mechanisms can leverage frameworks' context knowledge and new technologies and tools to control platform and application functions and their interaction, in terms of performance (e.g., by recommending more efficient resource allocation or new technological paradigms that have better performance), security (e.g., through the addition of security guidelines and best practices for the usage of the technologies and data flows by the framework) or even continuous self-configuration of data storage (e.g., to cope with new data types and flows and emerging storage technologies, such as in the case of extreme analytics databases).

8

REFERENCES

Abdul, J., Alkathiri, M., & Potdar, M. B. (2016). Geospatial Hadoop (GS-Hadoop) an efficient mapreduce based engine for distributed processing of shapefiles. *International Conference on Advances in Computing, Communication, & Automation*. Bareilly, India. doi:10.1109/ICACCAF.2016.7748956

Abeykoon, V., Kamburugamuve, S., Govindarajan, K., Wickramasinghe, P., Widanage, C., Perera, N., . . . Von Laszewski, G. (2019). Streaming Machine Learning Algorithms with Big Data Systems. *IEEE International Conference on Big Data.* Los Angeles, CA, USA. doi:10.1109/BigData47090.2019.9006337

Adhikari, R., & Agrawal, R. K. (2013). An Introductory Study on Time Series Modeling and Forecasting. *CoRR - Computing Research Repository, abs/1302.6613*.

Agarwal, S., & Rajan, K. S. (2016). Performance analysis of MongoDB versus PostGIS/PostGreSQL databases for line intersection and point containment spatial queries. *Spatial Information Research volume, 24*, 671-677. doi:10.1007/s41324-016-0059-1

Aggarwal, C. C. (2017). *Outlier Analysis.* Yorktown Heights, New York, USA: Springer.

Aigner, W., Miksch, S., Schumann, H., & Tominski, C. (2011). *Visualization of Time-Oriented Data.* Vienna, Austria: Springer.

Aji, A., Wang, F., Vo, H., Lee, R., Liu, Q., Zhang, X., & Saltz, J. (2013). Hadoop-GIS: A High Performance Spatial Data Warehousing System over MapReduce. *Proceedings of the VLDB Endowment.* Trento, Italy. doi:10.14778/2536222.2536227

Akbari, M., Samadzadegan, F., & Weibel, R. (2015). A generic regional spatio-temporal co-occurrence pattern mining model: a case study for air pollution. *Journal of Geographical Systems, 17*(3), 249-274.

Alarabi, L., Mokbel, M. F., & Musleh, M. (2018). ST-Hadoop: a MapReduce framework for spatio-temporal data. *GeoInformatica, 22*(4), 785-813.

America's Advanced Traveller Information Systems Committee. (2000). *Closing the Data Gap: Guidelines for Quality Advanced Traveler Information Systems Data.* US: U. C. Berkeley.

Anbaroglu, B., Heydecker, B., & Cheng, T. (2014). Spatio-temporal clustering for non-recurrent traffic congestion detection on urban road networks. *Transportation Research Part C: Emerging Technologies, 48*(November 2014), 47-65.

Andrienko, G., & Andrienko, N. (2009). Interactive cluster analysis of diverse types of spatio-temporal data. *ACM SIGKDD Explorations Newsletter, 11*(2), 19-28. doi:10.1145/1809400.1809405

Andrienko, G., Andrienko, N., Demsar, U., Dransch, D., Dykes, J., Fabrikant, S. I., . . . Tominski, C. (2010). Space, time and visual analytics. *International Journal of Geographical Information Science, 24*(10), 1577-1600.

Andrienko, G., Andrienko, N., Fuchs, G., & Wood, J. (2017). Revealing Patterns and Trends of Mass Mobility Through Spatial and Temporal Abstraction of Origin-Destination Movement Data. *IEEE Transactions on Visualization and Computer Graphics, 23*(9), 2120-2136.

Andrienko, G., Andrienko, N., Mladenov, M., Mock, M., & Pölitz, C. (2012). Identifying Place Histories from Activity Traces with an Eye to Parameter Impact. *IEEE Transactions on Visualization and Computer Graphics, 18*(5), 675-688.

Ansari, M. Y., Ahmad, A., Khan, S. S., & Bhushan, G. (2020). Spatiotemporal clustering: a review. *Artificial Intelligence Review, 53*, 2381-2423. doi:10.1007/s10462-019-09736-1

Antunes, H. A. (2017). *Visualização e deteção offline de eventos de tráfego usando o processamento de eventos complexos.* Lisbon, Portugal: FCT-UNL. Retrieved from https://run.unl.pt/bitstream/10362/31879/1/Antunes_2017.pdf

Antunes, H., Figueiras, P., Costa, R., Teixeira, J., & Jardim-Gonçalves, R. (2019). Analysing Public Transport data through the use of Big Data tecnhologies for urban mobility. *International Young Engineers Forum (YEF-ECE).* Costa da Caparica, Portugal. doi:10.1109/YEF-ECE.2019.8740816

Antunes, H., Figueiras, P., Costa, R., Teixeira, J., & Jardim-Gonçalves, R. (2019). Big data analytics for extracting mobility patterns in a large urban center. *ICIST 2019 - 9th International Conference on Information Society and Techology.* Kopaonik, Serbia.

Antunes, H., Figueiras, P., Costa, R., Teixeira, J., & Jardim-Gonçalves, R. (2019). Discovery of Public Transportation Patterns Through the Use of Big Data Technologies for Urban Mobility. *ASME 2019 International Mechanical Engineering Congress and Exposition.* Salt Lake City, UT, USA. doi:10.1115/IMECE2019-11415

Anwar, A., Nagel, T., & Ratti, C. (2014). Traffic Origins: A Simple Visualization Technique to Support Traffic Incident Analysis. *IEEE Pacific Visualization Symposium (PACIFICVIS '14).* Washington, DC, USA.

Asadi, R., & Regan, A. (2019). Spatio-Temporal Clustering of Traffic Data with Deep Embedded Clustering. *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility.* Chicago, IL, USA. doi:10.1145/3356995.3364537

Atluri, G., Karpatne, A., & Kumar, V. (2018). Spatio-Temporal Data Mining: A Survey of Problems and Methods. *ACM Computing Surveys, 51*(4).

Azevedo, A., & Santos, M. F. (2008). KDD, SEMMA and CRISP-DM: A Parallel Overview. *IADIS European Conference on Data Mining.* Amsterdam, The Netherlands.

Bach, B., Dragicevic, P., Archambault, D., Hurter, C., & Carpendale, S. (2014). A Review of Temporal Data Visualizations Based on Space-Time Cube Operations. *EuroVis 2014: Eurographics/IEEE Conference on Visualization.* Swansea, Wales, UK.

Bala, M., Boussaid, O., & Alimazighi, Z. (2016). Extracting-Transforming-Loading Modeling Approach for Big Data Analytics. *International Journal of Decision Support System Technology, 8*(4), 50-69.

Banaei-Kashani, F., Shahabi, C., & Pan, B. (2011). Discovering patterns in traffic sensor data. *2nd ACM SIGSPATIAL International Workshop on GeoStreaming.* Chicago, IL, USA. doi:10.1145/2064959.2064963

Baptista e Silva, F., Herrera, M. M., Rosina, K., Barranco, R. R., Freire, S., & Schiavina, M. (2018). Analysing spatiotemporal patterns of tourism in Europe at high-resolution with conventional and big data sources. *Tourism Management, 68*(October 2018), 101-115.

Bartoszewski, D., Piorkowski, A., & Lupa, M. (2019). The Comparison of Processing Efficiency of Spatial Data for PostGIS and MongoDB Databases. *BDAS 2019: Beyond Databases, Architectures and Structures. Paving the Road to Smart Data Processing and Analysis* . Ustroń, Poland,.

Batran, M., Mejia, M. G., Kanasugi, H., Sekimoto, Y., & Shibasaki, R. (2018). Inferencing Human Spatiotemporal Mobility in Greater Maputo via Mobile Phone Big Data Mining . *International Journal of Geo-Information, 7*(7), 259.

Beimborn, E., Horowitz, A., Vijayan, S., & Bordewin, M. (1999). Land Use - Transportation Interaction. In *An Overview: Land Use and Economic Development in Stadewide Transportation Planning* (pp. 10-24). Milwakee, US: Federal Highway Administration.

Big Data Value Association. (2020). *European Big Data Value Strategic Research and Innovation Agenda 4.0.* Brussels, Belgium: BDVA.

Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. *Data & Knowledge Engineering, 60*(1), 208-221. doi:10.1016/j.datak.2006.01.013

Bollier, D. (2010). *The Promise and Peril of Big Data.* Washington, D.C.: Communications and Society Program, The Aspen Institute.

Bouattou, Z., Laurini, R., & Belbachir, H. (2017). Animated chorem-based summaries of geographic data streams from sensors in real time. *Journal of Visual Languages & Computing, 41*(August 2017), 54-69.

Brahim, M. B., Drira, W., Filali, F., & Hamdi, N. (2016). Spatial data extension for Cassandra NoSQL database. *Journal of Big Data, 3*, Article 11. doi:10.1186/s40537-016-0045-4

Brundson, C., Fotheringham, S., & Charlton, M. (1998). Geographically Weighted Regression-Modelling Spatial Non-Stationarity. *Journal of the Royal Statistical Society. Series D (The Statistician), 47*(3), 431-443.

Cao, N., Lin, C., Zhu, Q., Lin, Y.-R., Teng, X., & Wen, X. (2018). Voila: Visual Anomaly Detection and Monitoring with Streaming Spatiotemporal Data. *IEEE Transactions on Visualization and Computer Graphics, 24*(1), 23-33.

Chae, J., Thom, D., Bosch, H., Jang, Y., Maciejewski, R., Ebert, D. S., & Ertl, T. (2012). Spatiotemporal social media analytics for abnormal event detection and examination using seasonal-trend decomposition. *IEEE Conference on Visual Analytics Science and Technology (VAST).* Seattle, WA, USA.

Chambon, S., Galtier, M. N., Arnal, P. J., Wainrib, G., & Gramfort, A. (2018). A Deep Learning Architecture for Temporal Sleep Stage Classification Using Multivariate and Multimodal Time Series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering, 26*(4), 758-769.

Chan, E. S., Gawlick, D., Ghoneimy, A., & Liu, Z. H. (2014). Situation aware computing for big data. *IEEE International Conference on Big Data (Big Data).* Washington, DC, USA. doi:10.1109/BigData.2014.7004415

Chen, C. P., & Zhang, C.-Y. (2014). Data-intensive applications, challenges, techniques and technologies: A survey on Big Data. *Information Sciences, 275*(August 2014), 314-347.

Chen, L., & Englund, C. (2016). Cooperative Intersection Management: A Survey. *IEEE Transactions on Intelligent Transportation Systems, 17*(2), 570-586.

Chen, M., Mao, S., & Liu, Y. (2014). Big Data: A Survey. *Mobile Networks and Applications, 19*(2), 171-209.

Chen, R., & Xie, J. (2008). Open Source Databases and Their Spatial Extensions. In *Open Source Approaches in Spatial Data Handling* (pp. 105-129). Berlin, Heidelberg: Springer. doi:10.1007/978-3-540-74831-1_6

Cheng, H. (2016). Modeling and querying fuzzy spatiotemporal objects. *Journal of Intelligent & Fuzzy Systems, 31*(6), 2851-2858.

Choi, C., & Hong, S.-Y. (2021). MDST-DBSCAN: A Density-Based Clustering Method forMulti-dimensional Spatiotemporal Data. *International Journal of Geo-Information, 10*(6), 391-407. doi:10.3390/ijgi10060391

Chollet, F. (2015). *Keras: The Python Deep Learning API.* Retrieved from https://keras.io/

Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (2012). *Non-Functional Requirements in Software Engineering.* Heidelberg, Germany: Springer.

Clegg, D. (2015). Evolving data warehouse and BI architectures: The big data challenge. *TDWI Business Intelligence Journal, 20*(1), 19-24.

Collobert, R., Bengio, S., & Mariéthoz, J. (2017). *Torch: A Scientific Computing Framework for LuaJIT.* Retrieved from http://torch.ch/

Comber, A., & Wulder, M. (2019). Considering spatiotemporal processes in big data analysis: Insights from remote sensing of land cover and land use. *Transactions in Geographic Information Systems, 23*(5), 879-891.

Cook, K. A., & Thomas, J. J. (2005). *Illuminating the Path: The Research and Development Agenda for Visual Analytics.* United States: USDOE.

Costa, C. F. (2019). *Advancing the Design and Implementation of Big Data Warehousing Systems.* Escola de Engenharia. Universidade do Minho.

Costa, C., & Santos, M. Y. (2017). Big Data: state-of-the-art concepts, techniques, technologies, modeling approaches and research challenges. *IAENG International Journal of Computer Science, 44*(3), 285-301.

Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. (2012). *Distributed Systems: Concepts and Design.* Boston, MA, USA: Addison-Wesley.

Cuzzocrea, A., Gaber, M. M., Lattimer, S., & Grasso, G. M. (2016). Clustering-Based Spatio-Temporal Analysis of Big Atmospheric Data. *Proceedings of the International Conference on Internet of things and Cloud Computing.* Cambridge, UK. doi:10.1145/2896387.2900326

Dagaeva, M., Garaeva, A., Anikin, I., Makhmutova, A., & Minnikhanov, R. (2019). Big spatio-temporal data mining for emergency management information systems. *IET Intelligent Transportation Systems, 13*(11), 1649-1657. doi:10.1049/iet-its.2019.0171

Dargay, J., Gately, D., & Sommer, M. (2007). Vehicle ownership and income growth, world-wide: 1960-2030. *Energy Journal, 28*(4), 143-170.

Data Interoperability Standards Consortium. (2021). *What is Data interoperability?* (Data Interoperability Standards Consortium) Retrieved from https://datainteroperability.org/

DATEX II. (2018). *D2Light.* (DATEX II) Retrieved from https://docs.datex2.eu/downloads/d2light.html

De Mauro, A., Greco, M., & Grimaldi, M. (2014). What is Big Data? A Consensual Definition and a Review of Key Research Topics. *4th International Conference on Integrated Information.* Madrid: AIP.

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM, 51*(1), 107-113. doi:10.1145/1327452.1327492

Demchenko, Y., de Laat, C., & Membrey, P. (2014). Defining Architecture Components of the Big Data Ecosystem. *International Conference on Collaboration Technologies and Systems.* Minneapolis, MN, USA.

Deng, D., Shahabi, C., Demiryurek, U., Zhu, L., Yu, R., & Liu, Y. (2016). Latent Space Model for Road Networks to Predict Time-Varying Traffic. *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* San Francisco, CA, USA.

Diebold, F. (2012). *On the Origin(s) and Development of the Term "Big Data".* Philadelphia, US: Penn Institute for Economic Research, Department of Economics, University of Pennsylvania.

Docker, Inc. (2013). *Docker*. Retrieved from https://www.docker.com/

Doraiswamy, H., Vo, H. T., Silva, C. T., & Freire, J. (2016). A GPU-based index to support interactive spatio-temporal queries over historical data. *IEEE 32nd International Conference on Data Engineering.* Helsinki, Finland.

Dubé, J., & Legros, D. (2014). *Spatial Econometrics Using Microdata.* New Jersey, USA: John Wiley & Sons, Inc.

Easyway. (2011). *DATEX II –The standard for ITS on European Roads.* Retrieved from http://www.datex2.eu/

ECMA Technical Committee 39. (2017). The JSON Data Interchange Syntax. *ECMA 404*. Retrieved from http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf

Elasticsearch B.V. (2010). *ElasticSearch*. (Elasticsearch B.V.) Retrieved from https://www.elastic.co/

Eldawy, A., & Mokbel, M. F. (2015). SpatialHadoop: A MapReduce framework for spatial data. *IEEE 31st International Conference on Data Engineering.* Seoul, South Korea. doi:10.1109/ICDE.2015.7113382

Eldawy, A., Mokbel, M. F., Alharthi, A., Tarek, K., & Ghani, S. (2015). SHAHED: A MapReduce-based system for querying and visualizing spatio-temporal satellite data. *IEEE 31st International Conference on Data Engineering.* Seoul, South Korea.

Endel, F., & Piringer, H. (2015). Data Wrangling: Making data useful again. *IFAC-PapersOnLine, 48*(1), 111-112.

ESRI. (2020). *What is the ArcGis Data Store?* (ESRI) Retrieved from https://enterprise.arcgis.com/en/data-store/latest/install/windows/what-is-arcgis-data-store.htm

European Commission. (2010). European Interoperability Framework (EIF) for European public services. *Publications Office of the European Union*, 1-40.

European Commission. (2012). *Cordis: FP7 MobiS Project - Personalized Mobility Services for Energy Efficiency and Security through Avanced Artificial Intelligence Techniques*. Retrieved from https://cordis.europa.eu/project/id/318452

European Commission. (2015). *OPTIMUM: Multi-source Big Data Fusion Driven Proactivity for Intelligent Mobility.* Retrieved from CORDIS EU Research Results: https://cordis.europa.eu/project/id/636160

European Commission. (2018). *CORDIS: BOOST 4.0 Project*. Retrieved from https://cordis.europa.eu/project/id/780732

European Commission. (2020, 08 28). *INSPIRE Knowledge Base*. (European Commission) Retrieved from https://inspire.ec.europa.eu/

European ITS Platform. (2016, 07 18). *The EasyWay Programme (2007-2020) and its Projects*. Retrieved from European ITS Platform: https://www.its-platform.eu/highlights/easyway-programme-2007-2020-and-its-projects

European Telecommunications Standards Institute. (2012). *Intelligent Transport Systems*. Retrieved from http://www.etsi.org

European Telecommunications Standards Institute. (2015). Retrieved from ETSI: http://www.etsi.org/

F5, Inc. (2004). *Nginx: High Performance Load Balancer, Web Server and Reverse Proxy*. Retrieved from https://www.nginx.com/

Fan, W., & Bifet, A. (2013). Mining big data: current status, and forecast to the future. *ACM sIGKDD Explorations Newsletter, 14*(2), 1-5.

Fan, Y., Yang, J., Zhu, D., & Wei, K. (2010). A time-based integration method of spatio-temporal data at spatial database level. *Mathematical and Computer Modelling, 51*(11-12), 1286-1292.

Fayyad, U. M., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery: an overview. In *Advances in knowledge discovery and data mining* (pp. 1-34). AAAI Press / The MIT Press.

Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., & Berners-Lee, T. (1985). Hypertext Transfer Protocol -- HTTP/1.1. *RFC 2616*. doi:10.17487/RFC2616

Figueiras, P., Antunes, H., Guerreiro, G., Costa, R., & Jardim-Gonçalves, R. (2018). Visualisation and Detection of Road Traffic Events Using Complex Event Processing. *ASME 2018 International Mechanical Engineering Congress and Exposition.* Pittsburgh, PA, USA. doi:10.1115/IMECE2018-87909

Figueiras, P., Gonçalves, D., Costa, R., Guerreiro, G., Georgakis, P., & Jardim-Gonçalves, R. (2019). Novel Big Data-supported dynamic toll charging system: Impact assessment on

Portugal's shadow-toll highways. *Computers & Industrial Engineering, 135*, 476-491. doi:10.1016/j.cie.2019.06.043

Figueiras, P., Guerreiro, G., Costa, R., Bradesko, L., Stojanovic, N., & Jardim-Gonçalves, R. (2016). Big Data Harmonization for Intelligent Mobility: A Dynamic Toll-Charging Scenario. *On the Move to Meaningful Internet Systems: OTM 2016 Workshops.* Rhodes, Greece.

Figueiras, P., Guerreiro, G., Silva, R., Costa, R., & Jardim-Gonçalves, R. (2018). Data Processing and Harmonization for Intelligent Transportation Systems: An Application Scenario on Highway Traffic Flows. In *Learning Systems: From Theory to Practice* (pp. 281-301). Cham, Switzerland: Springer.

Figueiras, P., Herga, Z., Guerreiro, G., Rosa, A., Costa, R., & Jardim-Gonçalves, R. (2018). Real-Time Monitoring of Road Traffic Using Data Stream Mining. *IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC).* Stuttgart, Germany. doi:10.1109/ICE.2018.8436271

Figueiras, P., Silva, R., Ramos, A., Guerreiro, G., Costa, R., & Jardim-Gonçalves, R. (2016). Big Data Processing and Storage Framework for ITS: A Case Study on Dynamic Tolling. *ASME 2016 International Mechanical Engineering Congress and Exposition.* Phoenix, AZ, USA.

Floratou, A., Minhas, U. F., & Özcan, F. (2014). SQL-on-Hadoop: full circle back to shared-nothing database architectures. *Very Large Data Bases Endowment.* Hangzhou, China.

Foraita, R., Spallek, J., & Zeeb, H. (2014). Directed Acyclic Graphs. *Handbook of Epidemiology*, 1481-1517. doi:10.1007/978-0-387-09834-0_65

Fox, A., Eichelberger, C., Hughes, J., & Lyon, S. (2013). Spatio-temporal indexing in non-relational distributed databases. *IEEE International Conference on Big Data.* Silicon Valley, CA, USA. doi:10.1109/BigData.2013.6691586

Funkhouser, H. G. (1936). A note on a tenth century graph. *Osiris*, 260-262.

Galić, Z. (2016). *Spatio-Temporal Data Streams.* Boston, MA, USA: Springer.

Galić, Z., Mešković, E., & Osmanović, D. (2017). Distributed processing of big mobility data as spatio-temporal data streams. *Geoinformatica, 21*(2), 263-291. doi:10.1007/s10707-016-0264-z

Galinec, D., & Steingartner, W. (2013). A Look at Observe, Orient, Decide and Act Feedback Loop, Pattern-Based Strategy and Network Enabled Capability for Organizations Adapting to Change. *Acta Electrotechnica et Informatica, 13*(2), 39-49. doi:10.2478/aeei-2013-0027

Gantz, J., & Reinsel, D. (2011). *Extracting Value from Chaos.* IDC iView.

Garber, L. (2012). Using In-Memory Analytics to Quickly Crunch Big Data. *Computer, 45*(10), 16-18.

Gartner, Inc. (2013). *Gartner IT Glossary: Big Data Definition*. (Gartner, Inc.) Retrieved 2017, from http://www.gartner.com/it-glossary/big-data

Gatalsky, P., Andrienko, N., & Andrienko, G. (2004). Interactive analysis of event data using space-time cube. *8th International Conference on Information Visualisation.* London, UK.

Giao, B. C., & Anh, D. T. (2015). Similarity search in multiple high speed time series streams under dynamic time warping. *National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS).* Ho Chi Minh City, Vietnam.

Giao, B. C., & Anh, D. T. (2016). Similarity search for numerous patterns over multiple time series streams under dynamic time warping which supports data normalization. *Vietnam Journal of Computer Science, 3*(3), 181-196.

Gohar, M., Muzammal, M., & Rahman, A. U. (2018). SMART TSS: Defining transportation system behavior using big data analytics in smart cities. *Sustainable Cities and Society, 41*, 114-119. doi:10.1016/j.scs.2018.05.008

Goldman Sachs. (2017). *Cars 2025*. (Goldman Sachs) Retrieved from http://www.goldmansachs.com/our-thinking/technology-driving-innovation/cars-2025/

Gong, Y., Sinnott, R., & Rimba, P. (2018). RT-DBSCAN: Real-Time Parallel Clustering of Spatio-Temporal Data Using Spark-Streaming. *International Conference on Computational Science.* Wuxi, China. doi:10.1007/978-3-319-93698-7_40

González, D., Pérez, J., Milanés, V., & Nashashibi, F. (2016). A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems, 17*(4), 1135-1145.

Google Brain Team. (2015). *TensorFlow*. (Google, Inc.) Retrieved from https://www.tensorflow.org

Google, Inc. (2005). *Google Cloud BigTable*. (Google, Inc.) Retrieved from https://cloud.google.com/bigtable/

Google, Inc. (2006). *General Transit Feed Specification*. (Google, Inc.) Retrieved from https://gtfs.org/

Grafana Labs. (2018). *Grafana: The open observability platform*. Retrieved from https://grafana.com/

Grilo, A., & Jardim-Gonçalves, R. (2010). Value proposition on interoperability of BIM and collaborative working environments. *Automation in Construction, 19*(5), 522-530.

Grolinger, K., Higashino, W. A., Tiwari, A., & Capretz, M. A. (2013). Data management in cloud environments: NoSQL and NewSQL data stores. *Journal of Cloud Computing: Advances, Systems and Applications, 2*(1), 22.

Guan, X., Xie, C., Han, L., Zeng, Y., Shen, D., & Xing, W. (2020). MAP-Vis: A Distributed Spatio-Temporal Big Data Visualization Framework Based on a Multi-Dimensional Aggregation Pyramid Model. *MDPI Applied Sciences, 10*(2), 598-617. doi:10.3390/app10020598

Guo, J., Huang, W., & Williams, B. M. (2015). Real time traffic flow outlier detection using short-term traffic conditional variance prediction. *Transportation Research Part C - Emerging Technologies, 50*(SI), 160-172.

Gutiérrez, C., Figueiras, P., Oliveira, P., Costa, R., & Jardim-Gonçalves, R. (2015). Twitter mining for traffic events detection. *Science and Information Conference.* London, UK. doi:10.1109/SAI.2015.7237170

Hagedorn, S., & Tonndorf, J. (2016). *STARK - Spatio-Temporal Data Analytics on Spark*. (GitHub) Retrieved from https://github.com/dbis-ilm/stark

Harris, P., Brundson, C., & Charlton, M. (2013). The comap as a diagnostic tool for non-stationary kriging models. *International Journal of Geographical Information Science, 27*(3), 511-541.

He, F., Gu, L., Wang, T., & Zhang, Z. (2017). The synthetic geo-ecological environmental evaluation of a coastal coal-mining city using spatiotemporal big data: A case study in Longkou, China. *Journal of Cleaner Production, 142, Part B*(January 2017), 854-866.

He, Y., Tan, H., & Luo, W. (2014). MR-DBSCAN: a scalable MapReduce-based DBSCAN algorithm for heavily skewed data. *Frontiers of Computer Science, 8*(1), 83-99. doi:10.1007/s11704-013-3158-3

He, Z., Deng, M., Cai, J., Xie, Z., Guan, Q., & Yang, C. (2020). Mining spatiotemporal association patterns from complex geographic phenomena. *International Journal of Geographical Information Science, 34*(6), 1162-1187. doi:10.1080/13658816.2019.1566549

Heuvelink, G. B., & Brown, J. D. (2017). Uncertain Environmental Variables in GIS. In *Encyclopedia of GIS* (pp. 1184-1189). Cham, Switzerland: Springer International Publishing .

Heuvelink, G. B., Pebesma, E., & Gräler, B. (2017). Space-Time Geostatistics. In S. Shekhar, H. Xiong, & X. Zhou (Eds.), *Encyclopedia of GIS* (2 ed., pp. 1919-1926). Cham, Switzerland: Springer International Publishing AG.

Highways England. (2008). *NTIS DATEX II Service.* Retrieved from https://datex2.eu/sites/default/files/NIS%20P%20TIH%20008%20NTIS%20DATEXII%20v8.pdf

Highways England. (2015). *National Traffic Information System*. Retrieved from http://www.trafficengland.com

Hitachi Vantara Corporation. (2017). *Pentaho Business Analytics*. Retrieved 2019, from https://www.hitachivantara.com/en-us/products/big-data-integration-analytics/pentaho-business-analytics.html

Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K. Q. (2016). Deep Networks with Stochastic Depth. *European Conference on Computer Vision.* Amsterdam, The Netherlands. doi:10.1007/978-3-319-46493-0_39

Iamwan, A., Indikawati, F. I., Kwon, J., & Rao, P. (2016). Querying and Extracting Timeline Information fromRoad Traffic Sensor Data. *Sensors, 16*(9), 1340-1377.

InfluxData Inc. (2019). *Time series database (TSDB) explained*. Retrieved 2019, from https://www.influxdata.com/time-series-database/

InfluxData, Inc. (2013). *InfluxDB*. Retrieved 2019, from https://www.influxdata.com/products/influxdb-overview/

Internet Engineering Task Force (IETF). (2011, December). *The WebSocket Protocol*. doi:10.17487/RFC6455

Jain, A., Zamir, A. R., Savarese, S., & Saxena, A. (2016). Structural-RNN: Deep learning on spatio-temporal graphs. *IEEE Conference on Computer Vision and Pattern Recognition.* Las Vegas, NV, USA.

Jern, M., & Franzen, J. (2006). GeoAnalytics - Exploring spatio-temporal and multivariate data. *10th International Conference on Information Visualisation (IV'06).* London, England, UK.

Jia, D., & Ngoduy, D. (2016). Platoon based cooperative driving model with consideration of realistic inter-vehicle communication. *Transportation Research Part C: Emerging Technologies, 68*, 245-264.

Jia, X., Khandelwal, A., Nayak, G., Gerber, J., Carlson, K., West, P., & Kumar, V. (2017). Incremental Dual-memory LSTM in Land Cover Prediction. *23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* Halifax, NS, Canada.

Jia, X., Khandelwal, A., Nayak, G., Gerber, J., Carlson, K., West, P., & Kumar, V. (2017). Predict land covers with transition modeling and incremental learning. *17th SIAM International Conference on Data Mining.* Houston, USA.

John, A., Sugumaran, M., & Rajesh, R. S. (2016). Indexing and Query Processing Techniques in Spatio-temporal Data. *ICTACT Journal on Soft Computing, 6*(3), 1198-1217.

Karim, S., Soomro, T. R., & Burney, S. A. (2018). Spatiotemporal Aspects of Big Data. *Applied Computer Systems, 23*(2), 90-100. doi:10.2478/acss-2018-0012

Khaleghi, B., Khamis, A., & Karray, F. O. (2013). Multisensor data fusion: A review of the state-of-theart. *Information Fusion, 14*, 28-44.

Kim, D. H., Ryu, K. H., & Kim, H. S. (2000). A spatiotemporal database model and query language. *Journal of Systems and Software, 27 December 2000*, 129-149. doi:10.1016/S0164-1212(00)00066-2

Kim, S.-W., Park, S., & Chu, W. W. (2001). An Index-Based Approach for Similarity Search Supporting Time Warping in Large Sequence Databases. *Data Engineering.* Heidelberg, Germany.

Kisilevich, S., Mansmann, F., Nanni, M., & Rinzivillo, S. (2009). Spatio-temporal clustering. In *Data mining and knowledge discovery handbook* (pp. 855-874). Boston, MA, USA: Springer Press.

Konstantaras, A. (2020). Deep Learning and Parallel Processing Spatio-Temporal Clustering Unveil New Ionian Distinct Seismic Zone. *informatics, 7*(4), 39. doi:10.3390/informatics7040039

Kraak, M.-J. (2003). The space-time cube revisited from a geovisualization perspective. *21st International Cartographic Conference (ICC).* Durban, South Africa.

Kraak, M.-J., & Ormeling, F. (2010). *Cartography: visualisation of geospatial data.* Essex, UK: Pearson Education, Ltd.

Krishnan, K. (2013). *Data Warehousing in the Age of Big Data.* Elsevier Inc. doi:10.1016/C2012-0-02737-8

Kristensson, P. O., Dahlback, N., Anundi, D., Bjornstad, M., Gillberg, H., Haraldsson, J., . . . Stahl, J. (2008). An Evaluation of Space Time Cube Representation of Spatiotemporal Patterns. *IEEE Transactions on Visualization and Computer Graphics, 15*(4), 696-702.

Lamigueiro, O. P. (2014). *Displaying Time Series, Spatial, and Space-Time Data with R.* Boca Raton, FL, USA: CRC Press, Taylor & Francis Group.

Landensberger, T. v., Bremm, S., Andrienko, N., Andrienko, G., & Tekušová, M. (2012). Visual analytics methods for categoric spatio-temporal data. *IEEE Conference on Visual Analytics Science and Technology (VAST).* Seattle, WA, USA.

Laube, P., & Purves, R. (2010). Cross-scale movement trajectory analysis . *Proceedings of the GIS Research UK 18th Annual Conference GISRUK 2010.* London, UK.

Levine, N. (2017). Hot Spot (CrimeStat: A Spatial Statistical Program for the Analysis of Crime Incidents). In S. Shekhar, H. Xiong, & X. Zhou (Eds.), *Encyclopedia of GIS* (2 ed., p. 866). Cham, Switzerland: Springer International Publishing AG.

Li, S., Ye, X., Lee, J., Gong, J., & Qin, C. (2017). Spatiotemporal Analysis of Housing Prices in China: A Big Data Perspective. *Applied Spatial Analysis and Policy , 10*, 421-433.

Li, Z., Hu, F., Schnase, J. L., Duffy, D. Q., Lee, T., Bowen, M. K., & Yang, C. (2017). A spatiotemporal indexing approach for efficient processing of big array-based climate data with MapReduce. *International Journal of Geographical Information Science, 31*(1), 17-35.

Li, Z., Yang, C., Sun, M., Li, J., Xu, C., Huang, Q., & Liu, K. (2013). A High Performance Web-Based System for Analyzing and Visualizing Spatiotemporal Data for Climate Studies. *International Symposium on Web and Wireless Geographical Information Systems.* Banff, AB, Canada.

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. *ACM SIGMOD workshop on Research issues in data mining and knowledge discovery.* San Diego, CA, USA.

Lin, P., Chang, S., Wang, H., Huang, Q., & He, J. (2019). SpikeCD: a parameter-insensitive spiking neural network with clustering degeneracy strategy. *Neural Computing and Applications, 31*(8), 3933-3945.

Luckham, D. (2008). The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. *International Symposium on Rule Representation, Interchange and Reasoning on the Web.* Orlando, FL, USA. doi:10.1007/978-3-540-88808-6_2

Luo, X., Niu, L., & Zhang, S. (2018). An Algorithm for Traffic Flow Prediction Based on Improved SARIMA and GA. *KSCE Journal of Civil Engineering, 22*(10), 4107-4115.

Luyi, B., Yan, L., & Ma, Z. M. (2014). Querying fuzzy spatiotemporal data using XQuery. *Integrated Computer-Aided Engineering, 21*(2), 147-162.

Maciejewski, R., Rudolph, S., Hafen, R., Abusalah, A., Yakout, M., Ouzzani, M., . . . Ebert, D. S. (2010). A Visual Analytics Approach to Understanding Spatiotemporal Hotspots. *IEEE Transactions on Visualization and Computer Graphics, 16*(2), 205-220.

MacLeod, M. (2018). What makes interdisciplinarity difficult? Some consequences of domain specificity in interdisciplinary practice. *Synthese, 195*, 697-720. doi:10.1007/s11229-016-1236-4

Magdy, A., Aly, A. M., Mokbel, M. F., Elnikety, S., He, Y., & Nath, S. (2014). Mars: Real-time spatio-temporal queries on microblogs. *IEEE 30th International Conference on Data Engineering.* Chicago, IL, USA.

Magdy, A., Mokbel, M. F., Elniteky, S., Nath, S., & He, Y. (2014). Mercury: A memory-constrained spatio-temporal real-time search on microblogs. *IEEE 30th International Conference on Data Engineering.* Chicago, IL, USA.

Mahood, N., Burney, S. M., Rizwan, K., Shah, A., & Nadeem, A. (2017). Building Spatio-Temporal Database Model Based on Ontological Approach using Relational Database Environment. *Mehran University Research Journal of Engineering and Technology, 36*(4), 891-900.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., & Byers, A. H. (2011). *Big Data: The next frontier for Innovation, Competition and Productivity.* McKinsey Global Institute.

MariaDB. (2018). *ClustrixDB:*. Retrieved 2018, from https://mariadb.com/products/clustrixdb/

Martinez-Llario, J., & Gonzalez-Alcaide, M. (2011). Design of a Java spatial extension for relational databases. *Journal of Systems and Software, 84*(12), 2314-2323. doi:10.1016/j.jss.2011.06.072

Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Kull, M., Lachiche, N., . . . Flach, P. (2019). CRISP-DM Twenty Years Later:From Data Mining Processesto Data Science Trajectories. *IEEE Transactions on Knowledge and Data Engineering, Early Access*. doi:10.1109/TKDE.2019.2962680

Marz, N., & Warren, J. (2015). *Big Data: Principles and best practices of scalable realtime data systems.* Manning Publications Co.

McCarthy, C. (2014). *Does NoSQL have a place in GIS? - An open-source spatial database performance comparison with proven RDBMS.* Edinburgh: The University of Edinburgh.

McKinsey & Company. (2020, May 4). *The New Normal: The impact of COVID-19 on future mobility solutions* . Retrieved from https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-impact-of-covid-19-on-future-mobility-solutions

Meirelles, I. (2013). Spatio-temporal Structures. In *Design for Information An Introduction to the Histories, Theories, and Best Practices Behind Effective Information Visualizations* (pp. 158-183). Beverly, MA, USA: Rockport Publishers.

Melo-Castillo, A., Bureš, P., Herrera-Quintero, L. F., & Banse, K. (2017). Design and implementation of DATEX II profiles for truck parking systems. *15th International Conference on ITS Telecommunications (ITST).* Warsaw, Poland. doi:10.1109/ITST.2017.7972220

Melo-Castillo, A., Canon-Lozano, Y., Herrera-Quintero, L. F., Bureš, P., & Banse, K. (2016). Design of an exchange information component of O/D matrix for mass transportation systems based on DATEX II approach. *8th Euro American Conference on Telematics and Information Systems (EATIS).* Cartagena, Colombia. doi:10.1109/EATIS.2016.7520097

Microsoft. (2011). *PowerBI*. Retrieved from https://powerbi.microsoft.com

Microsoft. (2019). *Microsoft SQL Server 2019*. Retrieved from https://www.microsoft.com/en-ca/sql-server/sql-server-2019

Miller, H. J., & Han, J. (2009). Geographic Data Mining and Knowledge Discovery: An Overview. In *Geographic Data Mining and Knowledge Discovery* (pp. 1-26). London, UK: Chapman & Hall/CRC.

Mireo d. d. (2020). *Spacetime*. (Mireo d. d.) Retrieved from https://www.mireo.hr/spacetime

MongoDB, Inc. (2015). *MongoDB*. Retrieved 2018, from https://www.mongodb.org/

Monino, J.-L. (2021). Data Value, Big Data Analytics, and Decision-Making. *Journal of the Knowledge Economy, 12*, 256-267. doi:10.1007/s13132-016-0396-2

Movchan, A., & Zymbler, M. (2015). Time Series Subsequence Similarity Search Under Dynamic Time Warping Distance on the Intel Many-core Accelerators. *International Conference on Similarity Search and Applications.* Glasgow, Scotland, UK.

Murata, M., St. Laurent, S., & Kohn, D. (2001). eXtended Markup Language Media Types. *RFC 3023*. Retrieved from https://tools.ietf.org/html/rfc3023

Nakaya, T., & Yano, K. (2010). Visualising Crime Clusters in a Space-time Cube: An Exploratory Data-analysis Approach Using Space-time Kernel Density Estimation and Scan Statistics. *Transactions in GIS, 14*(3), 223-239.

Nallaperuma, D., Nawaratne, R., Bandaragoda, T., Adikari, A., Nguyen, S., Kempitya, T., . . . Pothuhera, D. (2019). Online Incremental Machine Learning Platform for Big Data-Driven Smart Traffic Management. *IEEE Transactions on Intelligent Transportation Systems, 20*(12), 4679-4690. doi:10.1109/TITS.2019.2924883

National Electrical Manufacturers Association . (1996). *National Transportation Communications for Intelligent Transportation Systems Protocol*. (National Electrical Manufacturers Association) Retrieved from https://www.ntcip.org/

NBD-PWG. (2015). *NIST Big Data Interoperability Framework: Volume 6, Reference Architecture.* Gaithersburg, MD, USA: National Institute of Standards and Technology.

Nguyen, H., Liu, W., & Chen, F. (2017). Discovering Congestion Propagation Patterns in Spatio-Temporal Traffic Data. *IEEE Transactions on Big Data, 3*(2), 169-180.

Nowacki, G. (2012). Development and Standardization of Intelligent Transport Systems. *International Journal on Marine Navigation and Safety of Sea Transportation*, 403-411.

NuoDB, Inc. (2018). *NuoDB*. Retrieved 2019, from https://www.nuodb.com/

OECD/ITF. (2015). *Big Data and Transport - Understanding and assessing options.* OECD.

Oliver, D. (2016). *Spatial Network Data: Concepts and Techniques for Summarization.* Redlands, CA, USA: Springer.

Oliver, D., Shekhar, S., Kang, J. M., Laubscher, R., Carlan, V., & Evans, M. R. (2012). Geo-referenced Time-series Summarization Using k-Full Trees: A Summary of Results. *IEEE 12th International Conference on Data Mining Workshops.* Brussels, Belgium.

Open Source GeoBI. (2010). *GeoKettle: Open Source Spatial ETL*. (Spatialytics) Retrieved 2018, from http://www.spatialytics.org/projects/geokettle/

Open Source Geospatial Foundation. (2001). *GeoServer*. (Open Source Geospatial Foundation) Retrieved from http://geoserver.org/

OpenJS Foundation. (2009). *NodeJS*. Retrieved from http://www.nodejs.org/

OPTIMUM Consortium. (2015). *OPTIMUM Project*. Retrieved from http://optimumproject.eu

OPTIMUM Consortium. (2016). *D2.4 - Data Harmonization - Initial Version.* European Commission. Retrieved from https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5beef9df2&appId=PPGMS

OPTIMUM Consortium. (2018). *D2.8 - Data Harmonization - Final version.* European Commission. Retrieved from https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5beefa598&appId=PPGMS

Oracle Corporation. (1979). *Oracle Database*. Retrieved from https://www.oracle.com/database/

Oracle Corporation. (1998). *Java Message Service (JMS)*. (Oracle Corporation) Retrieved from https://www.oracle.com/java/technologies/java-message-service.html

Oracle Corporation. (2019). *Oracle Spatial*. (Oracle Corporation) Retrieved from https://www.oracle.com/database/spatial/

Osborne, J. W. (2013). *Best Practices in Data Cleaning: A Complete Guide to Everything You Need to Do Before and After Collecting Your Data.* Thousand Oaks, CA, USA: SAGE Publications, Inc.

Pan, B., Demiryurek, U., Banaei-Kashani, F., & Shahabi, C. (2010). Spatiotemporal summarization of traffic data streams. *ACM SIGSPATIAL International Workshop on GeoStreaming.* San Jose, CA, USA.

Park, J., & Ram, S. (2004). Information systems interoperability: What lies beneath? *ACM Transactions on Information Systems, 22*(4), 595-632. doi:https://doi.org/10.1145/1028099.1028103

Paszke, A., Gross, S., Chintala, S., & Chanan, G. (2016). *PyTorch*. Retrieved from https://pytorch.org/

Pauly, A., & Schneider, M. (2017). Vague Spatial Data Types. In *Encyclopedia of GIS* (pp. 2393-2398). Cham, Switzerland: Springer International Publishing.

Pavlo, A., & Aslett, M. (2016). What's Really New with NewSQL? *ACM SIGMOD Record, 45*(2), 45-55. doi:10.1145/3003665.3003674

PCAST. (2014). *Big Data and Privacy: A Technological Perspective.* Washington: Executive Office of the President of the U.S.A.

Pearson, M. P. (2013). Researching Stonehenge: Theories Past and Present. *Archaeology International, 16*, 72-83. doi:10.5334/ai.1601

Pebesma, E. (2012). spacetime: Spatio-Temporal Data in R. *Journal of Statistical Software, 51*(7), 1-30.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, É. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*(85), 2825-2830.

Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems, 24*(3), 45-78.

Peixoto, J., & Moreira, A. (2013). Human Movement Analysis Using Heterogeneous Data Sources. *International Journal of Agricultural and Environmental Information Systems, 4*(3), 20.

Perrot, A., Bourqui, R., Hanusse, N., Lalanne, F., & Auber, D. (2015). Large interactive visualization of density functions on big data infrastructure. *IEEE 5th Symposium on Large Data Analysis and Visualization.* Chicago, IL, USA. doi:10.1109/LDAV.2015.7348077

pgRouting Community. (2007). *pgRouting Project*. Retrieved from https://pgrouting.org/

Plug, C., Xia, J., & Caulfield, C. (2011). Spatial and temporal visualisation techniques for crash analysis. *Accident Analysis & Prevention, 43*(6), 1937-1946.

Polson, N. G., & Sokolov, V. O. (2017). Deep learning for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies, 79*(June 2017), 1-17.

Portugal 2020, European Union. (2020). *Portugal 2020: Regional Development European Fund for Portugal*. Retrieved from https://www.portugal2020.pt/

Postel, J., & Reynolds, J. (1985). File Transfer Protocol. *RFC 959*. Retrieved from https://tools.ietf.org/html/rfc959

PostGIS Project Streering Committee. (n.d.). *PostGIS: Spatial and Geographic Objects for PostgreSQL*. Retrieved from https://postgis.net/

Project Jupyter. (2014). *Jupyter*. Retrieved from https://jupyter.org/

Prunayre, F.-X., Chartier, B., Coudert, M., Jacolin, Y., Eichar, J., & Lemoine, E. (2007). *Spatial Extension for Talend*. Retrieved from https://talend-spatial.github.io/

Pujari, A. K. (2001). *Data Mining Techniques.* Hyderabad, India: Universities Press.

Qiu, X., Ren, Y., Suganthan, P. N., & Amaratunga, G. A. (2017). Empirical Mode Decomposition based ensemble deep learning for load demand time series forecasting. *Applied Soft Computing, 54*(May 2017), 246-255.

QlikTech International AB. (1993). *Qlik ETL Solution*. Retrieved from https://www.qlik.com/us/etl

Quinn, S. D., & MacEachren, A. M. (2018). A geovisual analytics exploration of the OpenStreetMap crowd. *Cartography and Geograpic Information Science, 45*(2), 140-155.

Rancher Labs. (2014). *Rancher: Enterprise Kubernetes Management*. Retrieved from https://rancher.com/

Rao, K. V., Govardhan, A., & Rao, K. C. (2012). Spatiotemporal Data Mining: Issues, Tasks And Applications. *International Journal of Computer Science & Engineering Survey (IJCSES), 3*(1), 39-52. doi:10.5121/ijcses.2012.3104

RapidMiner, Inc. (2013). *RapidMiner Studio*. Retrieved 2018, from https://rapidminer.com/

Raza, A. (2012). Working with spatio-temporal data type. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XXXIX-B2*, 5-10.

RedisLabs. (2015). *Redis*. (RedisLabs) Retrieved from https://redis.io/

Reich, B. J., & Porter, M. D. (2015). Partially supervised spatiotemporal clustering for burglary crime series identification. *Journal of the Royal Statistical Society, Statistics in Society: Series A, 178*(2), 465-480.

Riveiro, M., Lebram, M., & Elmer, M. (2017). Anomaly Detection for Road Traffic: A VisualAnalytics Framework. *IEEE Transactions on Intelligent Transportation Systems, 18*(8), 2260-2270. doi:10.1109/TITS.2017.2675710

Rizzo, G., & Troncy, R. (2012). NERD: A Framework for Unifying Named Entity Recognition and Disambiguation Extraction Tools. *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics.* Avignon, France. doi:10.5555/2380921.2380936

Robinson, A. C., Peuquet, D. J., Pezanowski, S., Hardisty, F. A., & Swedberg, B. (2017). Design and evaluation of a geovisual analytics system for uncovering patterns in spatio-temporal event data. *Cartography and Geographic Information Science, 44*(3), 216-228.

Roche, D. (2019, October 12). *Geoparquet 0.0.3 Python Project*. Retrieved 2020, from https://pypi.org/project/geoparquet/

Rodrigue, J.-P. (2017). *The Geography of Transport Systems.* New York: Routledge.

Ronacher, A. (2010). *Flask: Web development, one drop at a time*. (Pallets) Retrieved from https://flask.palletsprojects.com/en/1.1.x/

Root, C., & Mostak, T. (2016). MapD: a GPU-powered big data analytics and visualization platform. *Special Interest Group on Computer Graphics and Interactive Techniques Conference (SIGGRAPH).* Anaheim, CA, USA. doi:10.1145/2897839.2927468

Rosa, A. M. (2017). *Análise de Fluxos em Tempo Real para Gestão de Dados de Tráfego.* Lisbon, Portugal: FCT-UNL. Retrieved from https://run.unl.pt/bitstream/10362/28226/1/Rosa_2017.pdf

Ruiz-Alarcon-Quintero, C. (2016). Harmonization of transport data sourcesaccordingto INSPIRE data specification on transport networks. *XII Conference on Transport Engineering, CIT 2016.* Valencia, Spain.

Ryu, S., Noh, J., & Kim, H. (2017). Deep Neural Network Based Demand Side Short Term Load Forecasting. *Energies, 10*(1), 3.

Sachdeva, V., & Chung, L. (2017). Handling Non-Functional Requirements For Big Data and IOT Projects in Scrum. *7th International Conference on Cloud Computing, Data Science & Engineering.* Noida, India.

Saeedmanesh, M., & Geroliminis, N. (2017). Dynamic clustering and propagation of congestion in heterogeneously congested urban traffic networks. *Transportation Research Procedia, 23*, 962-979.

Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. *International Conference on Collaboration Technologies and Systems (CTS).* San Diego, CA, USA.

Samper, J. J., Tomás, V. R., Soriano, F. R., & Pla Castells, M. (2013). Intelligent transport systems harmonisation assessment: use case of some Spanish intelligent transport systems services. *IET Intelligent Transport Systems*, 361-370.

Schmid, S., Gálicz, E., & Reinhardt, W. (2015). Performance investigation of selected SQL and NoSQL databases. *Association of Geographic Information Laboratories in Europe (AGILE).* Lisbon, Portugal.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks, 61*, 85-117. doi:10.1016/j.neunet.2014.09.003

Senožetnik, M., Bradeško, L., Kažic, B., Mladenic, D., & Šubic, T. (2016). Spatio-temporal clustering methods. *19th International Multiconference Information Society.* Ljubljana, Slovenia.

Shafranovich, Y. (2005). Common Format and MIME Type for Comma-Separated Values (CSV) Files. *RFC 4180*. Retrieved from https://tools.ietf.org/html/rfc4180

Shahid, N., Naqvi, I. H., & Bin Qaisar, S. (2015). Characteristics and classification of outlier detection techniques for wireless sensor networks in harsh environments: a survey. *Artificial Intelligence Review, 43*(2), 193-228.

Shao, W., Salim, F. D., Song, A., & Bouguettaya, A. (2016). Clustering Big Spatiotemporal-Interval Data. *IEEE Transactions on Big Data, 2*(3), 190-203. doi:10.1109/TBDATA.2016.2599923

Shekhar, S., Jiang, Z., Ali, R. Y., Eftelioglu, E., & Tang, X. (2015). Spatiotemporal Data Mining: A Computational Perspective. *ISPRS International Journal of Geo-Information, 4*, 2306-2338.

Shekhar, S., Zhang, P., & Huang, Y. (2010). Spatial Data Mining. In O. Mainon, & L. Rokach (Eds.), *Data Mining and Knowledge Discovery Handbook* (2 ed., pp. 837-854). New York, USA: Springer Science+Business Media.

Shi, Y., Deng, M., Yang, X., & Gong, J. (2018). Detecting anomalies in spatio-temporal flow data by constructing dynamic neighbourhoods. *Computers Environment and Urban Systems, 67*(January 2015), 80-96.

Shi, Z., & Pun-Cheng, L. S. (2019). Spatiotemporal Data Clustering: A Survey of Methods. *International Journal of Geo-Information, 8*(112).

Shrestha, A. (2014). *Visualizing Spatio-Temporal data.* Georgia State University.

Signore, R., Stegman, M. O., & Creamer, J. (1995). *The ODBC Solution: Open Database Connectivity in Distributed Environments.* New York, NY, USA: McGraw-Hill Inc.

Silva, J. P., & Santos, M. Y. (2010). Spatiotemporal database models and languages for moving objects: A review. *Iberian Conference on Information Systems and Technologies (CISTI).* Santiago de Compostela, Spain.

Silva, R. A. (2017). *Enhancing Exploratory Analysis across Multiple Levels of Detail of Spatiotemporal Events.* Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

Silveira, L. M., de Almeida, J. M., Marques-Neto, H. T., Sarraute, C., & Ziviani, A. (2016). MobHet: Predicting human mobility using heterogeneous data sources. *Computer Communications, 95*(1 December 2016), 54-68.

Song, W., Wang, L., & Zomaya, A. Y. (2017). Geographic spatiotemporal big data correlation analysis via the Hilbert–Huang transformation. *Journal of Computer and System Sciences, 89*(November 2017), 130-141.

Sonsilphong, S., Arch-int, N., Arch-int, S., & Pattarapongsin, C. (2016). A semantic interoperability approach to health-care data: Resolving data-level conflicts. *Expert Systems, 33*(6), 531-547. doi: https://doi.org/10.1111/exsy.12167

Sözer, A., Yazici, A., Oğuztüzün, H., & Taş, O. (2008). Modeling and querying fuzzy spatiotemporal databases. *Information Sciences, 178*(19), 3665-3682. doi:10.1016/j.ins.2008.05.034

Sriharsha, R. (2017). *Magellan: Geospatial Processing made easy*. (Databricks) Retrieved from https://github.com/harsha2010/magellan

Stanford NLP Group. (2003). *Stanford Log-linear Part-Of-Speech Tagger*. (Stanford NLP Group) Retrieved from https://nlp.stanford.edu/software/tagger.shtml

Steed, C. A. (2017). Interactive Data Visualization. In M. Chowdhury, A. Apon, & K. Dey (Eds.), *Data Analytics for intelligent Transportation Systems* (pp. 165-190). Elsevier.

Steiger, E., Westerholt, R., Resch, B., & Zipf, A. (2015). Twitter as an indicator for whereabouts of people? Correlating Twitter with UK census data. *Computers Environment and Urban Systems, 54*(November 2015), 255-265.

Strötgen, J., Zell, J., & Gertz, M. (2013). HeidelTime: Tuning English and Developing Spanish Resources for TempEval-3. *Proceedings of the Seventh International Workshop on Semantic Evaluation.* Atlanta, GA, USA.

SuperMap Software Co., Ltd. (1997). *SuperMap GIS*. (SuperMap Software Co., Ltd.) Retrieved from https://www.supermap.com/en-us/

Surkhovetskyy, G., Andrienko, N., Andrienko, G., & Fuchs, G. (2017). Data Abstraction for Visualizing Large Time Series. *Computer Graphics Forum, 36*(3). doi:10.1111/cgf.13237

Tableau Software. (2003). *Tableau: Business Intelligence and Analytics Software*. Retrieved from https://www.tableau.com/

Talend. (2006). *Talend Open Studio for Big data*. Retrieved 2019, from https://www.talend.com/products/talend-open-studio/

Tan, Q., Liu, Y., & Liu, J. (2020). Demystifying Deep Learning in Predictive Spatiotemporal Analytics: An Information-Theoretic Framework. *IEEE Transactions on Neural Networks and Learning Systems, Early Access*, 1-15. doi:10.1109/TNNLS.2020.3015215

Tang, L., Gao, J., Ren, C., Zhang, X., Yang, X., & Kan, Z. (2019). Detecting and Evaluating Urban Clusters with Spatiotemporal Big Data. *Sensors, 19*(3), 461. doi:10.3390/s19030461

Taylor, J. (2018). *CRISP-DM & Decision Management: Creating a decision-centric, repeatable approach.* Decision Management Solutions. Retrieved from http://www.decisionmanagementsolutions.com/wp-content/uploads/2018/07/CRISP-DM-and-Decision-Management-061718.pdf

The Apache Software Foundation. (2004). *Apache ActiveMQ*. (The Apache Software Foundation) Retrieved from https://activemq.apache.org/

The Apache Software Foundation. (2007). *Apache Camel*. (The Apache Software Foundation) Retrieved from https://camel.apache.org/

The Apache Software Foundation. (2007). *Apache HBase*. Retrieved from https://hbase.apache.org/

The Apache Software Foundation. (2008). *Apache Accumulo*. (The Apache Software Foundation) Retrieved from https://accumulo.apache.org/

The Apache Software Foundation. (2009). *Apache Flume*. Retrieved from https://flume.apache.org/

The Apache Software Foundation. (2011). *Apache Hive*. Retrieved 2018, from https://hive.apache.org/

The Apache Software Foundation. (2011). *Apache Ranger*. Retrieved from http://ranger.apache.org/

The Apache Software Foundation. (2011). *Apache Sentry*. Retrieved from http://sentry.apache.org/

The Apache Software Foundation. (2011). *Apache Sqoop*. Retrieved from http://sqoop.apache.org/

The Apache Software Foundation. (2012). *Apache Drill*. Retrieved from https://drill.apache.org/

The Apache Software Foundation. (2012). *Apache Mesos*. Retrieved from https://mesos.apache.org/

The Apache Software Foundation. (2014). *Apache Flink*. Retrieved from https://flink.apache.org/

The Apache Software Foundation. (2014). *Apache Mahout*. Retrieved 2018, from https://mahout.apache.org/

The Apache Software Foundation. (2015). *Apache SAMOA*. Retrieved 2019, from http://samoa.incubator.apache.org/

The Apache Software Foundation. (2015). *Apache Sedona*. (The Apache Software Foundation) Retrieved from http://sedona.apache.org/

The Apache Software Foundation. (2015). *Cloudera Impala*. (Cloudera, Inc) Retrieved 2018, from https://www.cloudera.com/documentation/enterprise/5-3-x/topics/impala_intro.html

The Apache Software Foundation. (2016). *Apache Cassandra*. Retrieved 2018, from http://cassandra.apache.org/

The Apache Software Foundation. (2016). *Apache Zeppelin*. Retrieved from https://zeppelin.apache.org

The Apache Software Foundation. (2017). *Apache Kafka*. Retrieved from http://kafka.apache.org/

The Apache Software Foundation. (2018). *Apache Hadoop*. Retrieved 2018, from https://hadoop.apache.org/

The Apache Software Foundation. (2018). *Apache Knox*. Retrieved from http://knox.apache.org/

The Apache Software Foundation. (2018). *Apache Parquet*. Retrieved from http://parquet.apache.org/

The Apache Software Foundation. (2018). *Apache Spark*. Retrieved 2018, from https://spark.apache.org/

The Apache Software Foundation. (2018). *Apache Spark MLlib*. Retrieved 2018, from https://spark.apache.org/mllib/

The Apache Software Foundation. (2018). *Apache Storm*. Retrieved 2018, from https://storm.apache.org/

The Apache Software Foundation. (2019). *Apache Ambari*. Retrieved from https://ambari.apache.org/

The Apache Software Foundation. (2019). *Apache Druid*. Retrieved 2019, from https://druid.apache.org/

The Apache Software Foundation. (2020). *Apache HAWQ*. Retrieved from http://hawq.apache.org/

The Apache Software Foundation. (2020). *Apache MADlib*. Retrieved from http://madlib.apache.org/

The Apache Software Foundation. (2020). *Apache ORC*. Retrieved from http://orc.apache.org/

The Apache Software Foundation. (2021). *Apache Sedona Documentation: Run GeoSpark via Zeppelin*. Retrieved from https://sedona.apache.org/archive/tutorial/zeppelin/

The Geographic JSON Working Group. (2016). *The GeoJSON Specification (RFC 7946)*. (The Geographic JSON Working Group) Retrieved from https://geojson.org/

The GeoMesa Project . (2013). *GeoMesa*. Retrieved from https://www.geomesa.org/

The HSQL Development Group. (2001). *HyperSQL - HSQLDB*. (The HSQL Development Group) Retrieved from http://hsqldb.org/

The Kubernetes Authors . (2014). *Kubernetes*. Retrieved from https://kubernetes.io/

The PosgreSQL Global Development Group. (1996). *PostgreSQL: The World's Most Advanced Open Source Relational Database*. Retrieved from https://www.postgresql.org/

The PostgreSQL Global Development Group. (2016, January 7). *PostgreSQL 9.5: UPSERT, Row Level Security, and Big Data.* Retrieved from PostgreSQL: https://www.postgresql.org/about/news/postgresql-95-upsert-row-level-security-and-big-data-1636/

The Presto Software Foundation . (2013). *Presto: SQL query engine for Big Data*. Retrieved from https://prestosql.io/

The R Foundation. (2000). *The R Project for Statistical Computing*. Retrieved 2018, from https://www.r-project.org/

The University of Waikato. (2005). *Weka 3: Machine Learning Software in Java*. Retrieved 2018, from https://www.cs.waikato.ac.nz/ml/weka/

The University of Waikato. (2010). *Moa Machine Learning for Streams*. Retrieved 2019, from https://moa.cms.waikato.ac.nz/

TIBCO Software Inc. (2007). *TIBCO Spotfire Analytics*. Retrieved from https://www.tibco.com/products/tibco-spotfire

TIBCO Software Inc. (2020). *Jaspersoft ETL*. Retrieved from https://www.jaspersoft.com/

Timescale, Inc. (2017). *TimeScaleDB*. Retrieved 2019, from https://www.timescale.com/

Tobler, W. R. (1970). A Computer Movie Simulating Urban Growth in the Detroit Region. *Economic Geography, 46*, 234-240.

Tomás, V. R., Castells, M. P., Samper, J. J., & Soriano, F. R. (2013). Intelligent transport systems harmonisation assessment: use case of some Spanish intelligent transport systems services. *IET Intelligent Transport Systems, 7*(3), 361-370. doi:10.1049/iet-its.2013.0008

TraefikLabs. (2016). *Traefik: The Cloud Native Application Proxy*. Retrieved from https://traefik.io/traefik/

United Nations, Department of Economic and Social Affairs, Population Division. (2014). *World Urbanization Prospects: The 2014 Revision, Highlights.* United Nations.

van der Veer, H., & Wiles, A. (2008). *Achieving Technical Interoperability -the ETSI Approach.* France: European Telecommunications Standards Institute.

Vatsavai, R. R., Chandola, V., Klasky, S., Ganduly, A., Stefanidis, A., & Shekhar, S. (2012). Spatiotemporal Data Mining in the Era of Big Spatial Data:Algorithms and Applications. *ACM SIGSPATIAL BIGSPATIAL.* Redondo Beach, CA, USA.

VMware, Inc. (2020). *GreenPlum Database*. Retrieved from https://greenplum.org/

VMware, Inc. (2007). *RabbitMQ*. Retrieved from https://www.rabbitmq.com/

VMware, Inc. (2020). *Spring Boot*. Retrieved from https://spring.io/projects/spring-boot

VoltDB, Inc. (2019). *VoltDB*. Retrieved 2019

Vonk, D. (2015, June 23). *Open issue: Using the spatial framework for hadoop with data stored in ORC files.* Retrieved from GitHub: https://github.com/Esri/spatial-framework-for-hadoop/issues/85

Wang, J., Gu, Q., Wu, J., Liu, G., & Xiong, Z. (2016). Traffic Speed Prediction and Congestion Source Exploration: A Deep Learning Method. *16th IEEE International Conference on Data Mining.* Barcelona, Spain.

Wang, J., Tang, J., Xu, Z., Wang, Y., Xue, G., Zhang, X., & Yang, D. (2017). Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach. *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications.* Atlanta, GA, USA.

Wang, S., Zhong, E., Cai, W., Zhou, Q., Lu, H., Gu, Y., . . . Long, L. (2018). A Visual Analytics Framework for Big Spatiotemporal Data. *Proceedings of the 2nd ACM SIGSPATIAL Workshop on Analytics for Local Events and News.* Seattle, WA, USA. doi:10.1145/3282866.3282869

Wang, S., Zhong, E., Zhou, Q., Cui, X., Lu, H., Yun, W., . . . Long, L. (2018). An Integrated Visual Analytics Framework for Spatiotemporal Data. *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* Seattle, WA, USA. doi:10.1145/3284566.3284574

Wang, S., Zhong, Y., & Wang, E. (2019). An integrated GIS platform architecture for spatio-temporal big data. *Future Generation Computer Systems, 94*(May 2019), 160-172.

Wang, T., Ke, H., Zheng, X., Wang, K., Sangaiah, A. K., & Liu, A. (2019). Big Data Cleaning Based on Mobile Edge Computing in Industrial Sensor-Cloud. *IEEE Transactions on Industrial Informatics, 16*(2), 1321-1329. doi:10.1109/TII.2019.2938861

Ward, J. S., & Barker, A. (2013). *Undefined By Data: A Survey of Big Data Definitions.* Cornell University Library.

Wei-Feng, L., Wei, C., & Jian, H. (2008). Research on a DATEX II based Dynamic Traffic Information Publish Platform. *2nd International Symposium on Intelligent Information Technology Application.* Shanghai, China. doi:10.1109/IITA.2008.81

Westerheim, H. (2014). Supporting Overall Interoperability in the Transport Sector By Means of a Framework the Case of the Its Station. *Norsk konferanse for organisasjoners bruk av IT, NOKOBIT.*

Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a Standard Process Model for DataMining. *4th international conference on the practical applications of knowledge discovery and data mining.* London, UK. doi:10.1.1.198.5133

Wong, P. C., Shen, H.-W., Johnson, C. R., Chen, C., & Ross, R. B. (2012). The Top 10 Challenges in Extreme-Scale Visual Analytics. *IEEE Computer Graphics and Applications, 32*(4), 63-67.

World Health Organization. (2014). *7 million premature deaths annually linked to air pollution*. (World Health Organization) Retrieved from http://www.who.int/mediacentre/news/releases/2014/air-pollution/en/

WSO2. (2005). *WSO2 Complex Event Processor (WSO2 CEP)*. (WSO2) Retrieved from https://wso2.com/products/complex-event-processor/

Wu, S., Morandini, L., & Sinnott, R. O. (2015). SMASH: A Cloud-based Architecture for Big Data Processing and Visualization of Traffic Data. *IEEE International Conference on Data Science and Data Intensive Systems.* Sydney, NSW, Australia.

Wu, X., Zurita-Milla, R., & Kraak, M.-J. (2015). Co-clustering geo-referenced time series: exploring spatio-temporal patterns in Dutch temperature data. *International Journal of Geographical Information Science, 29*(4), 624-642. doi:10.1080/13658816.2014.994520

Wu, X., Zurita-Milla, R., & Kraak, M.-J. (2015). Co-clustering geo-referenced time series: exploring spatio-temporal patterns in Dutch temperature data. *International Journal of Geographical Information Science, 29*(4), 624-642. doi:10.1080/13658816.2014.994520

Wu, X., Zurita-Milla, R., Verdiguier, E. I., & Kraak, M.-J. (2018). Triclustering Georeferenced Time Series for Analyzing Patterns of Intra-Annual Variability in Temperature. *Annals of the American Association of Geographers, 108*(1), 71-87.

Xiaofeng, M., Ding, Z., & Xu, J. (2012). *Moving Objects Management: Models, Techniques and Applications.* Springer-Verlag Berlin Heidelberg.

Xu, J., Deng, D., Demiryurek, U., Shahabi, C., & van der Schaar, M. (2015). Mining the Situation: Spatiotemporal Traffic Prediction With Big Data. *IEEE Journal of Selected Topics in Signal Processing, 9*(4), 702-715.

Yang, C., Clarke, K., Shekhar, S., & Tao, C. (2020). Big Spatiotemporal Data Analytics: a research andinnovation frontier. *International Journal of Geographical Information Science, 34*(6), 1075-1088. doi: https://doi.org/10.1080/13658816.2019.1698743

Yang, C., Sun, M., Liu, K., Huang, Q., Li, Z., Gui, Z., . . . Zhou, N. (2015). Contemporary Computing Technologiesfor Processing Big Spatiotemporal Data. In *Space-Time Integration in Geography and GIScience: Research Frontiers in the US and China* (pp. 327-351). Dordrecht, Netherlands: Springer .

Yang, C., Yu, M., Hu, F., Jiang, Y., & Li, Y. (2017). Utilizing Cloud Computing to address big geospatial data challenges. *Computers, Environment and Urban Systems, 61, Part B*(January 2017), 120-128.

Yu, J., Tahir, A., & Sarwat, M. (2019). GeoSparkViz in Action: A Data System with Built-in Support for Geospatial Visualization. *IEEE 35th International Conference on Data Engineering (ICDE).* Macao, China. doi:10.1109/ICDE.2019.00222

Yu, J., Wu, J., & Sarwat, M. (2015). GeoSpark: a cluster computing framework for processing large-scale spatial data. *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems.* New York, USA. doi:10.1145/2820783.2820860

Yu, J., Zhang, Z., & Sarwat, M. (2018). GeoSparkViz: a scalable geospatial data visualization framework in the apache spark ecosystem. *Proceedings of the 30th International Conference on Scientific and Statistical Database Management.* Bozen-Bolzano, Italy. doi:10.1145/3221269.3223040

Yue, P., & Tan, Z. (2018). GIS Databases and NoSQL Databases. In *Comprehensive Geographic Information Systems* (pp. 50-79). Amsterdam, Netherlands: Elsevier, Inc.

Zeitouni, K., Yeh, L., & Aufaure, M.-A. (2007). Join Indices as a Tool for Spatial Data Mining. In J. F. Roddick, & K. Hornsby, *Lecture Notes in Artificial Intelligence* (pp. 105-116). Springer.

Žejn, G., Šolc, T., Kožar, D., Samastur, M., Mrdjenovič, M., Čuhalev, J., . . . Kostelec, P. (2015). *OpenData.si*. Retrieved from https://opendata.si/

Zhang, F., Zheng, Y., Xu, D., Du, Z., Wang, Y., Liu, R., & Ye, X. (2016). Real-Time Spatial Queries for Moving Objects Using Storm Topology. *International Journal of Geo-Information, 5*(10), 178-197. doi:10.3390/ijgi5100178

Zhang, J., Zheng, Y., Qi, D., Li, R., & Yi, X. (2016). DNN-Based Prediction Model for Spatio-Temporal Data. *24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.* Burlingame, CA, USA.

Zhao, J., Qu, Q., Zhang, F., Xu, C., & Liu, S. (2017). Spatio-Temporal Analysis of Passenger Travel Patterns in Massive Smart Card Data. *IEEE Transactions on Intelligent Transportation Systems, 18*(11), 3135-3146.

Zhou, H., Li, M., & Gu, Z. (2020). Knowledge Fusion and Spatiotemporal Data Cleaning: A Review. *IEEE Fifth International Conference on Data Science in Cyberspace.* Hong Kong, China.

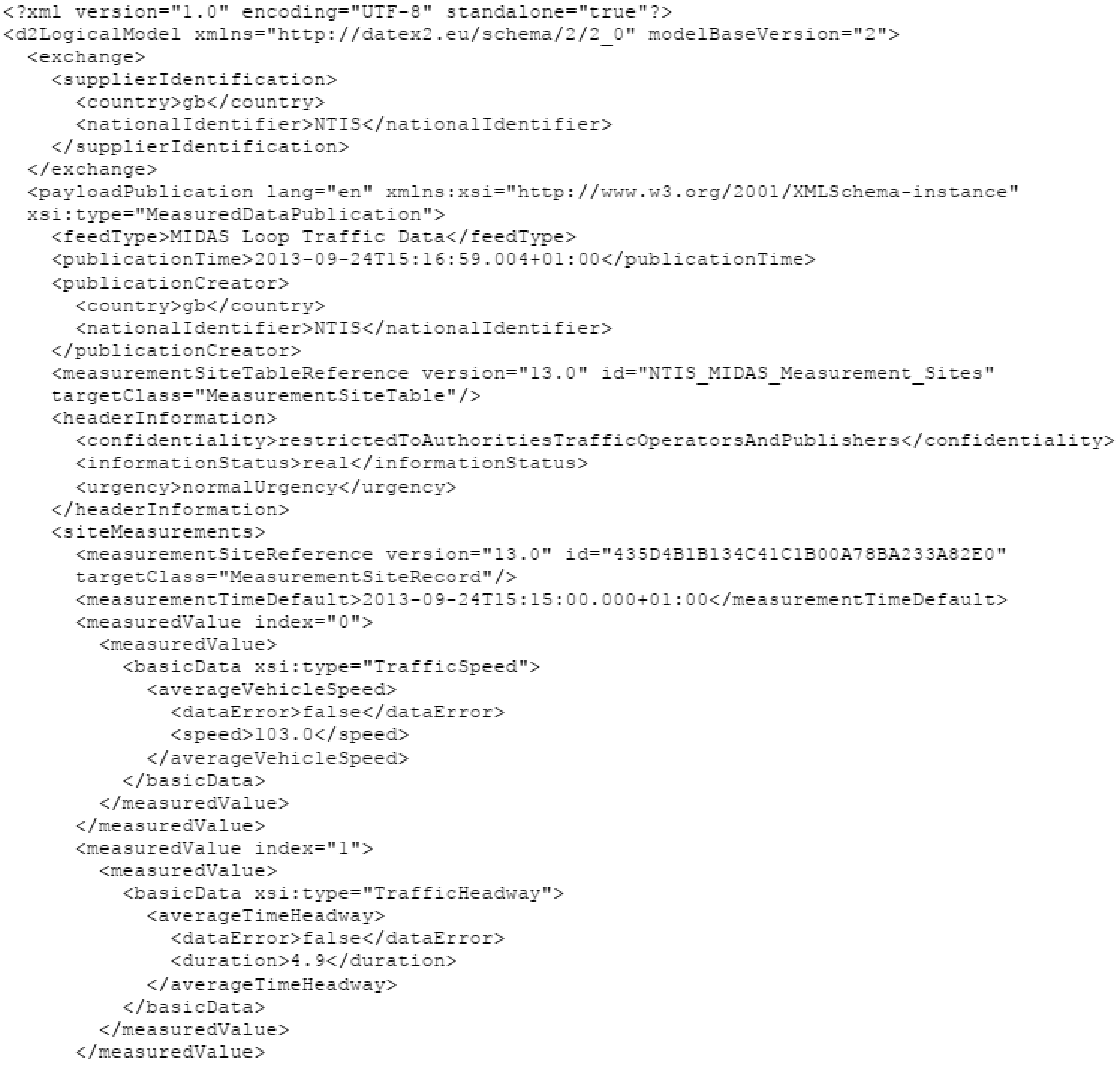
Zhou, X., & Lin, H. (2017). Local Sensitivity Analysis. In *Encyclopedia of GIS* (pp. 1130-1131). Cham, Switzerland: Springer International Publishing.

Zhu, J. Y., Zhang, C., Zhang, H., Zhi, S., Li, V. O., Han, J., & Zheng, Y. (2018). pg-Causality: Identifying Spatiotemporal Causal Pathways for Air Pollutants with Urban Big Data. *IEEE Transactions on Big Data, 4*(4), 571-585.

A

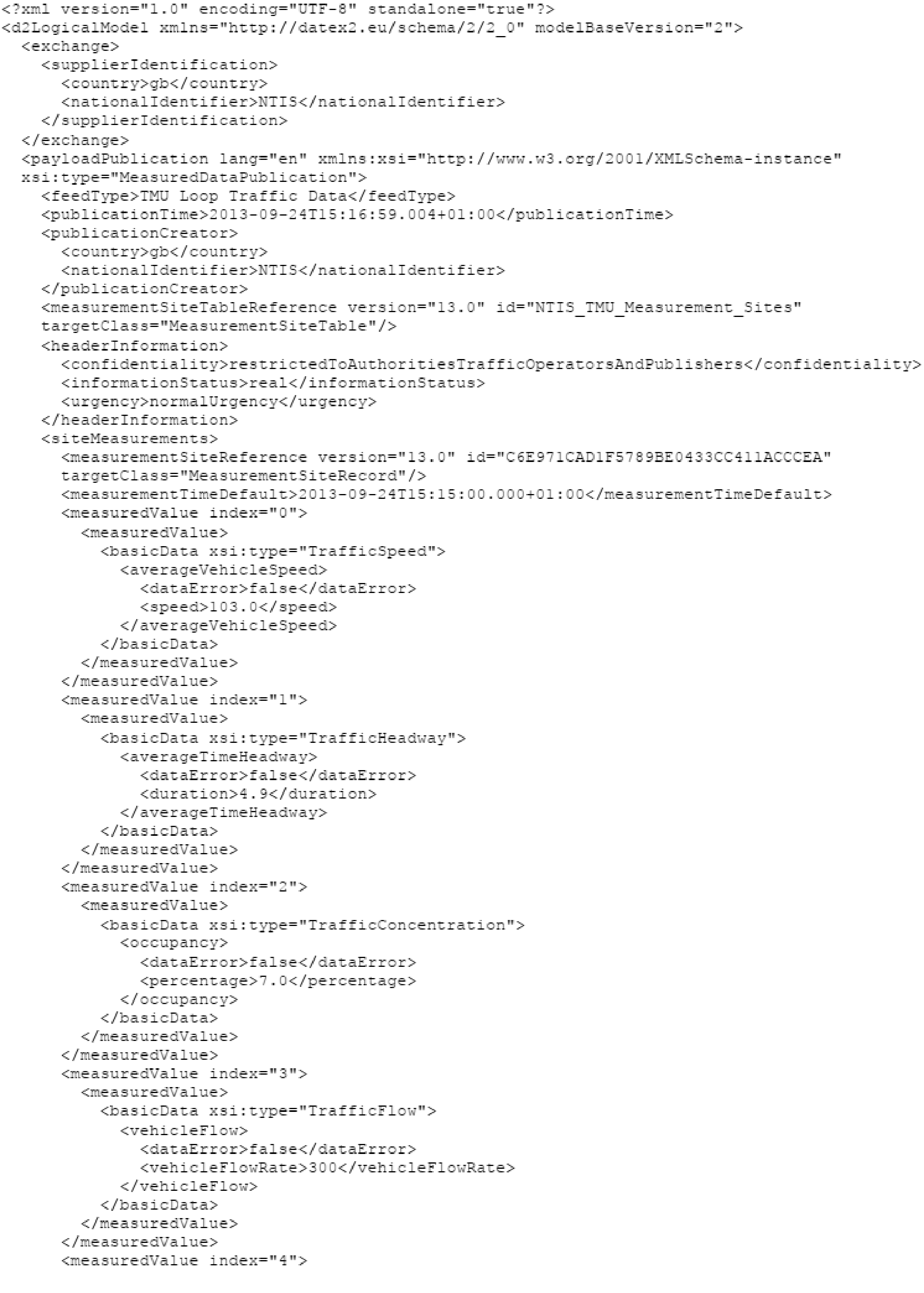
APPENDIXES

A.1. Example NTIS MIDAS Traffic Sensor Data DATEX II XML Format (Highways England, 2008)



```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<d2LogicalModel xmlns="http://datex2.eu/schema/2/2_0" modelBaseVersion="2">
  <exchange>
    <supplierIdentification>
      <country>gb</country>
      <nationalIdentifier>NTIS</nationalIdentifier>
    </supplierIdentification>
  </exchange>
  <payloadPublication lang="en" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="MeasuredDataPublication">
    <feedType>MIDAS Loop Traffic Data</feedType>
    <publicationTime>2013-09-24T15:16:59.004+01:00</publicationTime>
    <publicationCreator>
      <country>gb</country>
      <nationalIdentifier>NTIS</nationalIdentifier>
    </publicationCreator>
    <measurementSiteTableReference version="13.0" id="NTIS_MIDAS_Measurement_Sites"
    targetClass="MeasurementSiteTable"/>
    <headerInformation>
      <confidentiality>restrictedToAuthoritiesTrafficOperatorsAndPublishers</confidentiality>
      <informationStatus>real</informationStatus>
      <urgency>normalUrgency</urgency>
    </headerInformation>
    <siteMeasurements>
      <measurementSiteReference version="13.0" id="435D4B1B134C41C1B00A78BA233A82E0"
      targetClass="MeasurementSiteRecord"/>
      <measurementTimeDefault>2013-09-24T15:15:00.000+01:00</measurementTimeDefault>
      <measuredValue index="0">
        <measuredValue>
          <basicData xsi:type="TrafficSpeed">
            <averageVehicleSpeed>
              <dataError>false</dataError>
              <speed>103.0</speed>
            </averageVehicleSpeed>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="1">
        <measuredValue>
          <basicData xsi:type="TrafficHeadway">
            <averageTimeHeadway>
              <dataError>false</dataError>
              <duration>4.9</duration>
            </averageTimeHeadway>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="2">
        <measuredValue>
          <basicData xsi:type="TrafficConcentration">
            <occupancy>
              <dataError>false</dataError>
              <percentage>7.0</percentage>
            </occupancy>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="3">
        <measuredValue>
          <basicData xsi:type="TrafficFlow">
            <vehicleFlow>
              <dataError>false</dataError>
              <vehicleFlowRate>300</vehicleFlowRate>
            </vehicleFlow>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="4">
        <measuredValue>
          <basicData xsi:type="TrafficFlow">
            <vehicleFlow>
              <dataError>false</dataError>
              <vehicleFlowRate>120</vehicleFlowRate>
            </vehicleFlow>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="5">
        <measuredValue>
          <basicData xsi:type="TrafficFlow">
            <vehicleFlow>
              <dataError>false</dataError>
              <vehicleFlowRate>180</vehicleFlowRate>
            </vehicleFlow>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="6">
        <measuredValue>
          <basicData xsi:type="TrafficFlow">
            <vehicleFlow>
              <dataError>false</dataError>
              <vehicleFlowRate>120</vehicleFlowRate>
            </vehicleFlow>
          </basicData>
        </measuredValue>
      </measuredValue>

      ....more measured values for each monitored lane

    </siteMeasurements>
  </payloadPublication>
</d2LogicalModel>
```

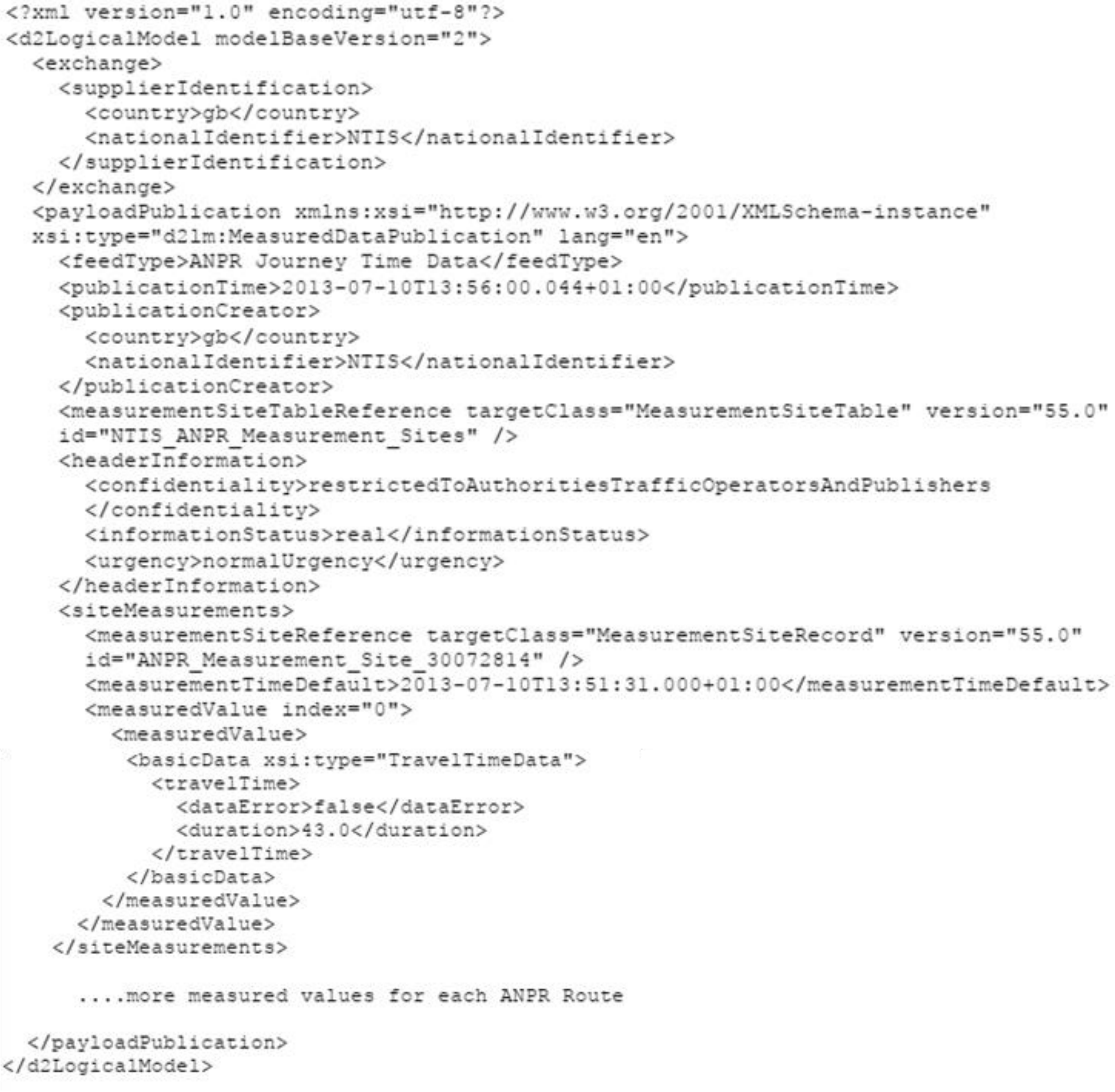
A.2. Example NTIS TMU Traffic Sensor Data DATEX II XML Format (Highways England, 2008)



```xml
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<d2LogicalModel xmlns="http://datex2.eu/schema/2/2_0" modelBaseVersion="2">
  <exchange>
    <supplierIdentification>
      <country>gb</country>
      <nationalIdentifier>NTIS</nationalIdentifier>
    </supplierIdentification>
  </exchange>
  <payloadPublication lang="en" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="MeasuredDataPublication">
    <feedType>TMU Loop Traffic Data</feedType>
    <publicationTime>2013-09-24T15:16:59.004+01:00</publicationTime>
    <publicationCreator>
      <country>gb</country>
      <nationalIdentifier>NTIS</nationalIdentifier>
    </publicationCreator>
    <measurementSiteTableReference version="13.0" id="NTIS_TMU_Measurement_Sites"
    targetClass="MeasurementSiteTable"/>
    <headerInformation>
      <confidentiality>restrictedToAuthoritiesTrafficOperatorsAndPublishers</confidentiality>
      <informationStatus>real</informationStatus>
      <urgency>normalUrgency</urgency>
    </headerInformation>
    <siteMeasurements>
      <measurementSiteReference version="13.0" id="C6E971CAD1F5789BE0433CC411ACCCEA"
      targetClass="MeasurementSiteRecord"/>
      <measurementTimeDefault>2013-09-24T15:15:00.000+01:00</measurementTimeDefault>
      <measuredValue index="0">
        <measuredValue>
          <basicData xsi:type="TrafficSpeed">
            <averageVehicleSpeed>
              <dataError>false</dataError>
              <speed>103.0</speed>
            </averageVehicleSpeed>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="1">
        <measuredValue>
          <basicData xsi:type="TrafficHeadway">
            <averageTimeHeadway>
              <dataError>false</dataError>
              <duration>4.9</duration>
            </averageTimeHeadway>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="2">
        <measuredValue>
          <basicData xsi:type="TrafficConcentration">
            <occupancy>
              <dataError>false</dataError>
              <percentage>7.0</percentage>
            </occupancy>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="3">
        <measuredValue>
          <basicData xsi:type="TrafficFlow">
            <vehicleFlow>
              <dataError>false</dataError>
              <vehicleFlowRate>300</vehicleFlowRate>
            </vehicleFlow>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="4">
```

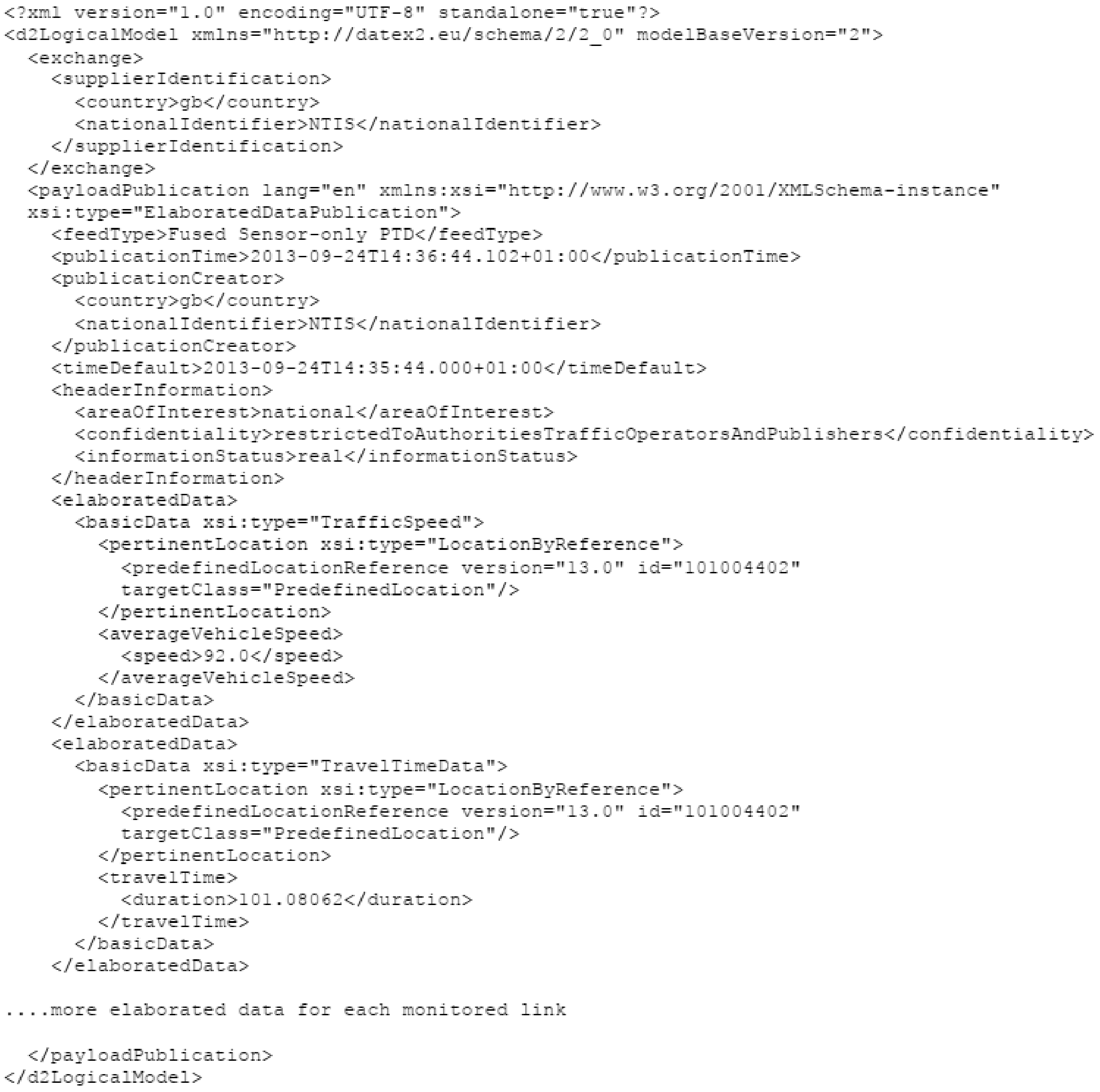
```
        <measuredValue>
          <basicData xsi:type="TrafficFlow">
            <vehicleFlow>
              <dataError>false</dataError>
              <vehicleFlowRate>120</vehicleFlowRate>
            </vehicleFlow>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="5">
        <measuredValue>
          <basicData xsi:type="TrafficFlow">
            <vehicleFlow>
              <dataError>false</dataError>
              <vehicleFlowRate>180</vehicleFlowRate>
            </vehicleFlow>
          </basicData>
        </measuredValue>
      </measuredValue>
      <measuredValue index="6">
        <measuredValue>
          <basicData xsi:type="TrafficFlow">
            <vehicleFlow>
              <dataError>false</dataError>
              <vehicleFlowRate>120</vehicleFlowRate>
            </vehicleFlow>
          </basicData>
        </measuredValue>
      </measuredValue>

      ....more measured values for each monitored lane

    </siteMeasurements>
  </payloadPublication>
</d2LogicalModel>
```

A.3. Example NTIS ANPR Traffic Sensor Data DATEX II XML Format (Highways England, 2008)



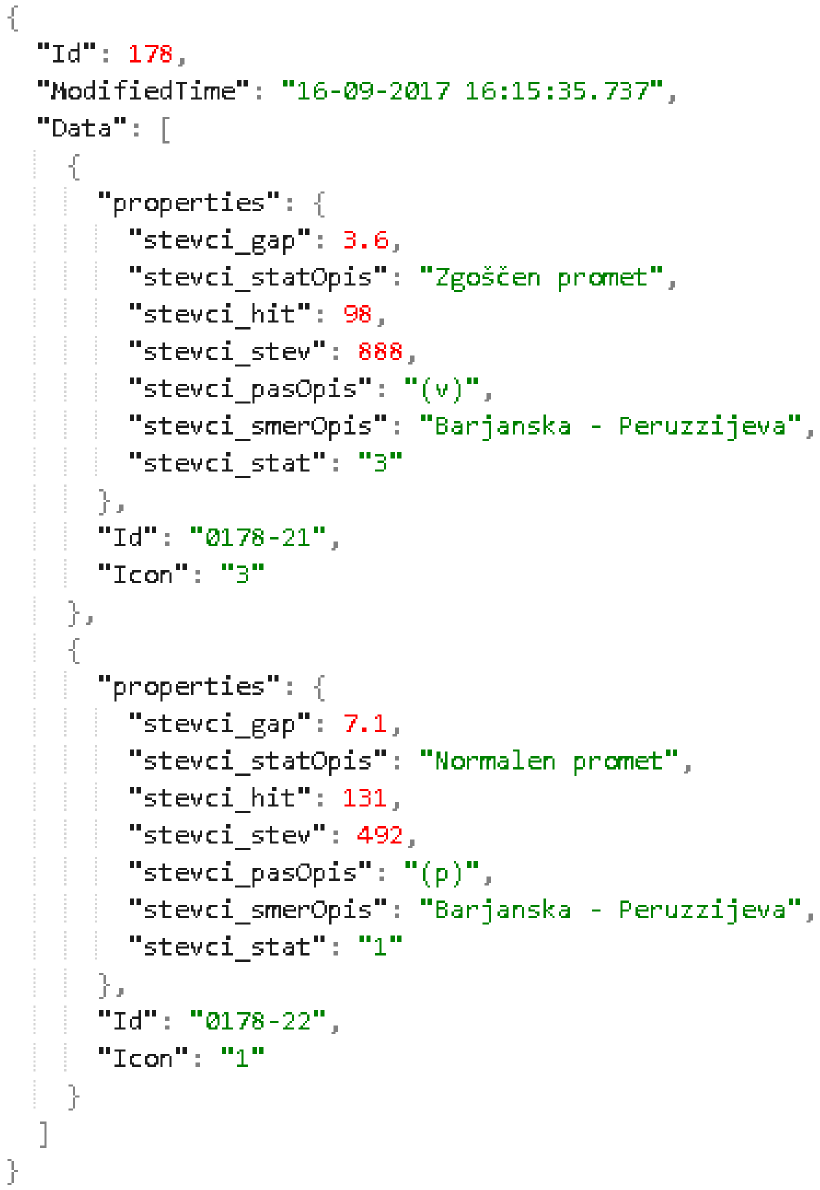
```
<?xml version="1.0" encoding="utf-8"?>
<d2LogicalModel modelBaseVersion="2">
  <exchange>
    <supplierIdentification>
      <country>gb</country>
      <nationalIdentifier>NTIS</nationalIdentifier>
    </supplierIdentification>
  </exchange>
  <payloadPublication xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="d2lm:MeasuredDataPublication" lang="en">
    <feedType>ANPR Journey Time Data</feedType>
    <publicationTime>2013-07-10T13:56:00.044+01:00</publicationTime>
    <publicationCreator>
      <country>gb</country>
      <nationalIdentifier>NTIS</nationalIdentifier>
    </publicationCreator>
    <measurementSiteTableReference targetClass="MeasurementSiteTable" version="55.0"
    id="NTIS_ANPR_Measurement_Sites" />
    <headerInformation>
      <confidentiality>restrictedToAuthoritiesTrafficOperatorsAndPublishers
      </confidentiality>
      <informationStatus>real</informationStatus>
      <urgency>normalUrgency</urgency>
    </headerInformation>
    <siteMeasurements>
      <measurementSiteReference targetClass="MeasurementSiteRecord" version="55.0"
      id="ANPR_Measurement_Site_30072814" />
      <measurementTimeDefault>2013-07-10T13:51:31.000+01:00</measurementTimeDefault>
      <measuredValue index="0">
        <measuredValue>
          <basicData xsi:type="TravelTimeData">
            <travelTime>
              <dataError>false</dataError>
              <duration>43.0</duration>
            </travelTime>
          </basicData>
        </measuredValue>
      </measuredValue>
    </siteMeasurements>

      ....more measured values for each ANPR Route

  </payloadPublication>
</d2LogicalModel>
```

A.4. Example NTIS Fused Traffic Sensor Data DATEX II XML Format (Highways England, 2008)



```
<?xml version="1.0" encoding="UTF-8" standalone="true"?>
<d2LogicalModel xmlns="http://datex2.eu/schema/2/2_0" modelBaseVersion="2">
  <exchange>
    <supplierIdentification>
      <country>gb</country>
      <nationalIdentifier>NTIS</nationalIdentifier>
    </supplierIdentification>
  </exchange>
  <payloadPublication lang="en" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:type="ElaboratedDataPublication">
    <feedType>Fused Sensor-only PTD</feedType>
    <publicationTime>2013-09-24T14:36:44.102+01:00</publicationTime>
    <publicationCreator>
      <country>gb</country>
      <nationalIdentifier>NTIS</nationalIdentifier>
    </publicationCreator>
    <timeDefault>2013-09-24T14:35:44.000+01:00</timeDefault>
    <headerInformation>
      <areaOfInterest>national</areaOfInterest>
      <confidentiality>restrictedToAuthoritiesTrafficOperatorsAndPublishers</confidentiality>
      <informationStatus>real</informationStatus>
    </headerInformation>
    <elaboratedData>
      <basicData xsi:type="TrafficSpeed">
        <pertinentLocation xsi:type="LocationByReference">
          <predefinedLocationReference version="13.0" id="101004402"
          targetClass="PredefinedLocation"/>
        </pertinentLocation>
        <averageVehicleSpeed>
          <speed>92.0</speed>
        </averageVehicleSpeed>
      </basicData>
    </elaboratedData>
    <elaboratedData>
      <basicData xsi:type="TravelTimeData">
        <pertinentLocation xsi:type="LocationByReference">
          <predefinedLocationReference version="13.0" id="101004402"
          targetClass="PredefinedLocation"/>
        </pertinentLocation>
        <travelTime>
          <duration>101.08062</duration>
        </travelTime>
      </basicData>
    </elaboratedData>

....more elaborated data for each monitored link

  </payloadPublication>
</d2LogicalModel>
```

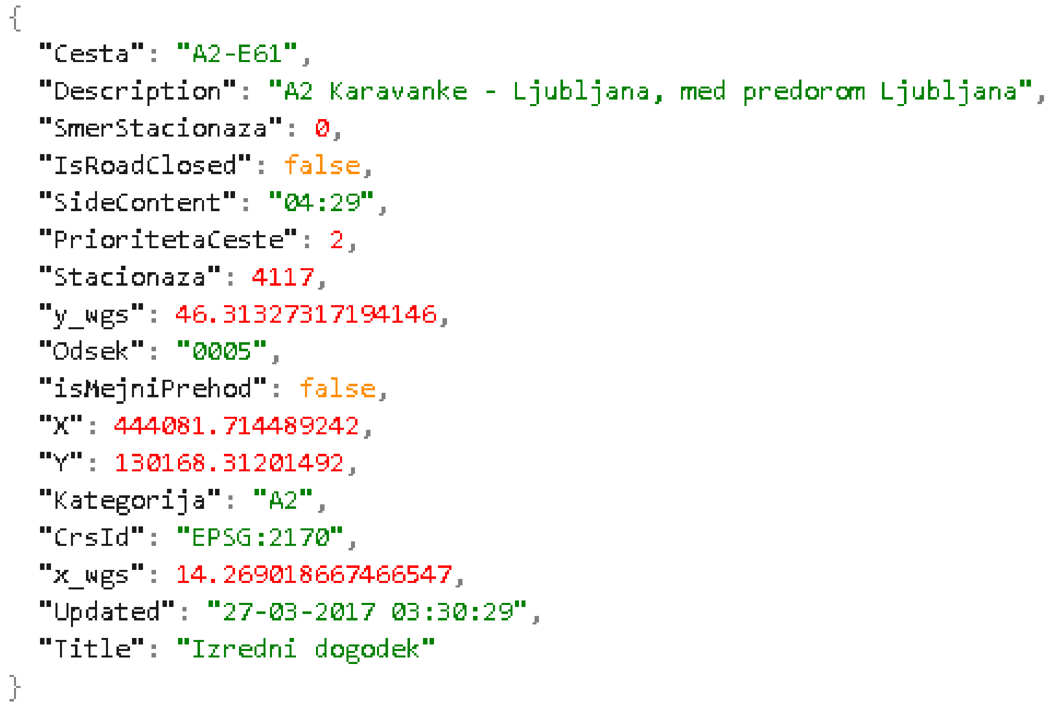
A.5. Example Slovenian Traffic Sensor Data



```
{
  "Id": 178,
  "ModifiedTime": "16-09-2017 16:15:35.737",
  "Data": [
    {
      "properties": {
        "stevci_gap": 3.6,
        "stevci_statOpis": "Zgoščen promet",
        "stevci_hit": 98,
        "stevci_stev": 888,
        "stevci_pasOpis": "(v)",
        "stevci_smerOpis": "Barjanska - Peruzzijeva",
        "stevci_stat": "3"
      },
      "Id": "0178-21",
      "Icon": "3"
    },
    {
      "properties": {
        "stevci_gap": 7.1,
        "stevci_statOpis": "Normalen promet",
        "stevci_hit": 131,
        "stevci_stev": 492,
        "stevci_pasOpis": "(p)",
        "stevci_smerOpis": "Barjanska - Peruzzijeva",
        "stevci_stat": "1"
      },
      "Id": "0178-22",
      "Icon": "1"
    }
  ]
}
```

Parameter	Description
stevci_gap	Gap time between vehicles (in seconds)
stevci_statOpis	Traffic status (e.g normal, heavy)
stevci_hit	Average speed (in Km/h)
stevci_stev	Occupancy
stevci_pasOpis	Bearing, direction
stevci_smerOpis	Start and end locations for the road stretch where the sensor is located
stevci_stat	Numerical traffic status

A.6. Example Slovenian Traffic Event Data

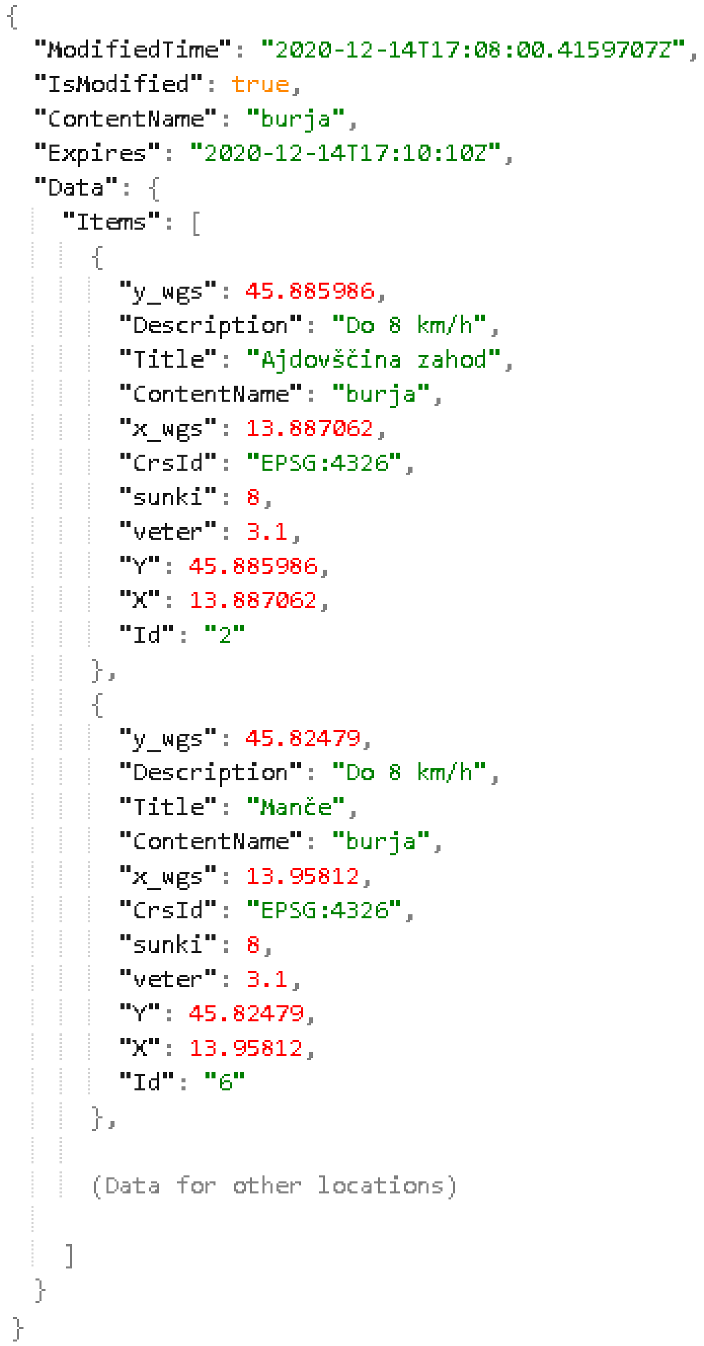


```
{
  "Cesta": "A2-E61",
  "Description": "A2 Karavanke - Ljubljana, med predorom Ljubljana",
  "SmerStacionaza": 0,
  "IsRoadClosed": false,
  "SideContent": "04:29",
  "PrioritetaCeste": 2,
  "Stacionaza": 4117,
  "y_wgs": 46.31327317194146,
  "Odsek": "0005",
  "isMejniPrehod": false,
  "X": 444081.714489242,
  "Y": 130168.31201492,
  "Kategorija": "A2",
  "CrsId": "EPSG:2170",
  "x_wgs": 14.269018667466547,
  "Updated": "27-03-2017 03:30:29",
  "Title": "Izredni dogodek"
}
```

Parameter	Description
Cesta	Road name
Description	Event description
SmerStacionaza	Bearing, direction
IsRoadClosed	True if the road is closed after the event occurred; False otherwise
SideContent	N.D.
PrioritetaCeste	Road priority
Stacionaza	Road kilometre mark
x_wgs, y_wgs	Latitude and longitude expressed in the World Geodetic System (WGS) coordinate system
Odsek	Road section
isMejniPrehod	True if it is a border cross; False otherwise
X, Y	Latitude and longitude

Kategorija	Road category
CrsId	EPSG ID for the X and Y parameters (MGI/Slovenia Grid coordinate system)
Updated	Date and time of last update
Title	Type of event

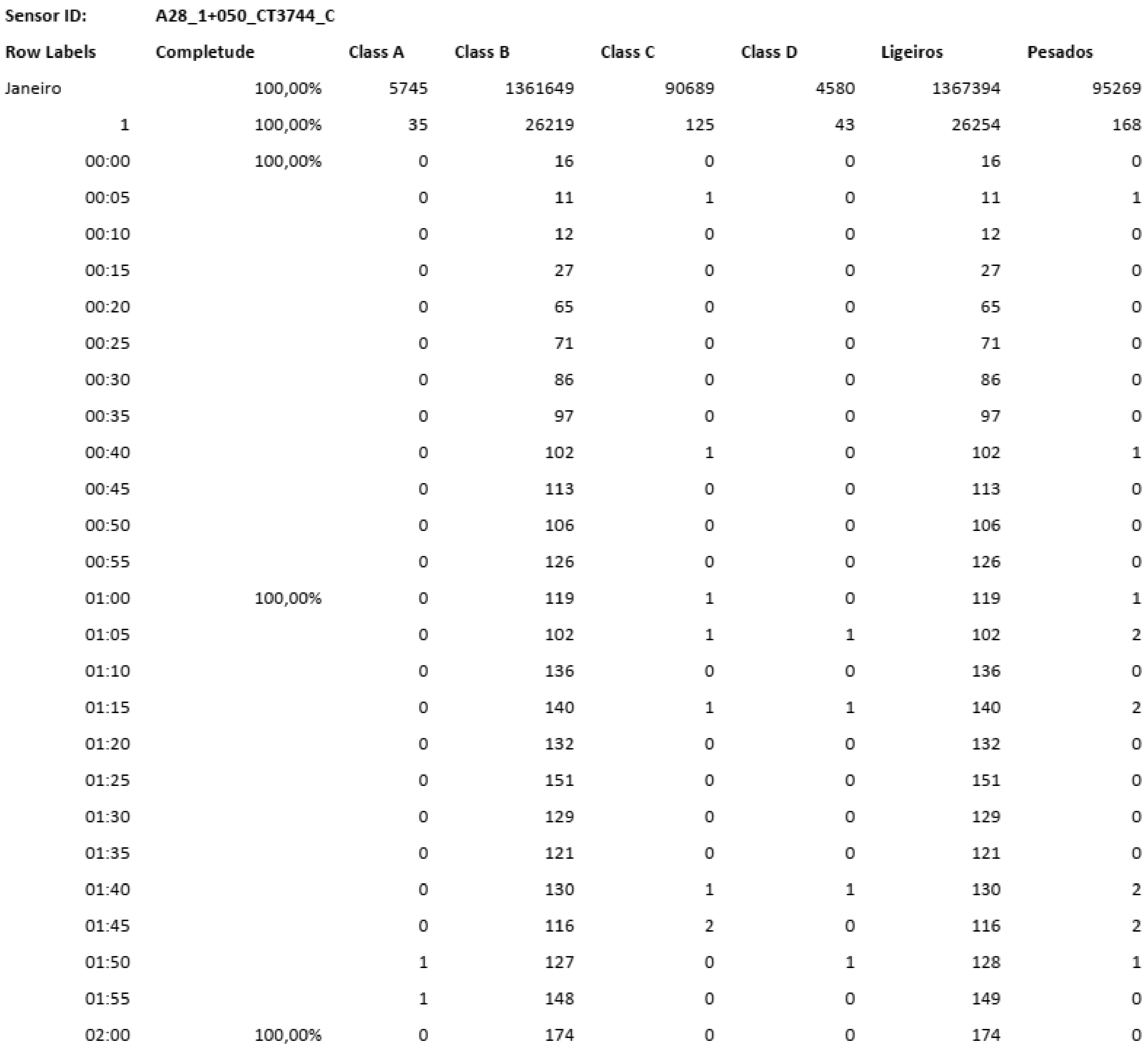
A.7. Example Slovenian Wind Conditions Data



```
{
  "ModifiedTime": "2020-12-14T17:08:00.4159707Z",
  "IsModified": true,
  "ContentName": "burja",
  "Expires": "2020-12-14T17:10:10Z",
  "Data": {
    "Items": [
      {
        "y_wgs": 45.885986,
        "Description": "Do 8 km/h",
        "Title": "Ajdovščina zahod",
        "ContentName": "burja",
        "x_wgs": 13.887062,
        "CrsId": "EPSG:4326",
        "sunki": 8,
        "veter": 3.1,
        "Y": 45.885986,
        "X": 13.887062,
        "Id": "2"
      },
      {
        "y_wgs": 45.82479,
        "Description": "Do 8 km/h",
        "Title": "Manče",
        "ContentName": "burja",
        "x_wgs": 13.95812,
        "CrsId": "EPSG:4326",
        "sunki": 8,
        "veter": 3.1,
        "Y": 45.82479,
        "X": 13.95812,
        "Id": "6"
      },

      (Data for other locations)

    ]
  }
}
```

Parameter	Description
x_wgs, y_wgs	Latitude and longitude expressed in the World Geodetic System (WGS) coordinate system
Description	Text description of the wind conditions (in Km/h)
Title	Name of the location
Content-Name	Name for the content data (*burja* means wind)
CrsId	EPSG ID for the X and Y parameters (WGS coordinate system)
sunki	Gust speed (in Km/h)
veter	Wind speed (in Km/h)
X, Y	Latitude and longitude

A.8. Example IP Traffic Counter Data in CSV Format

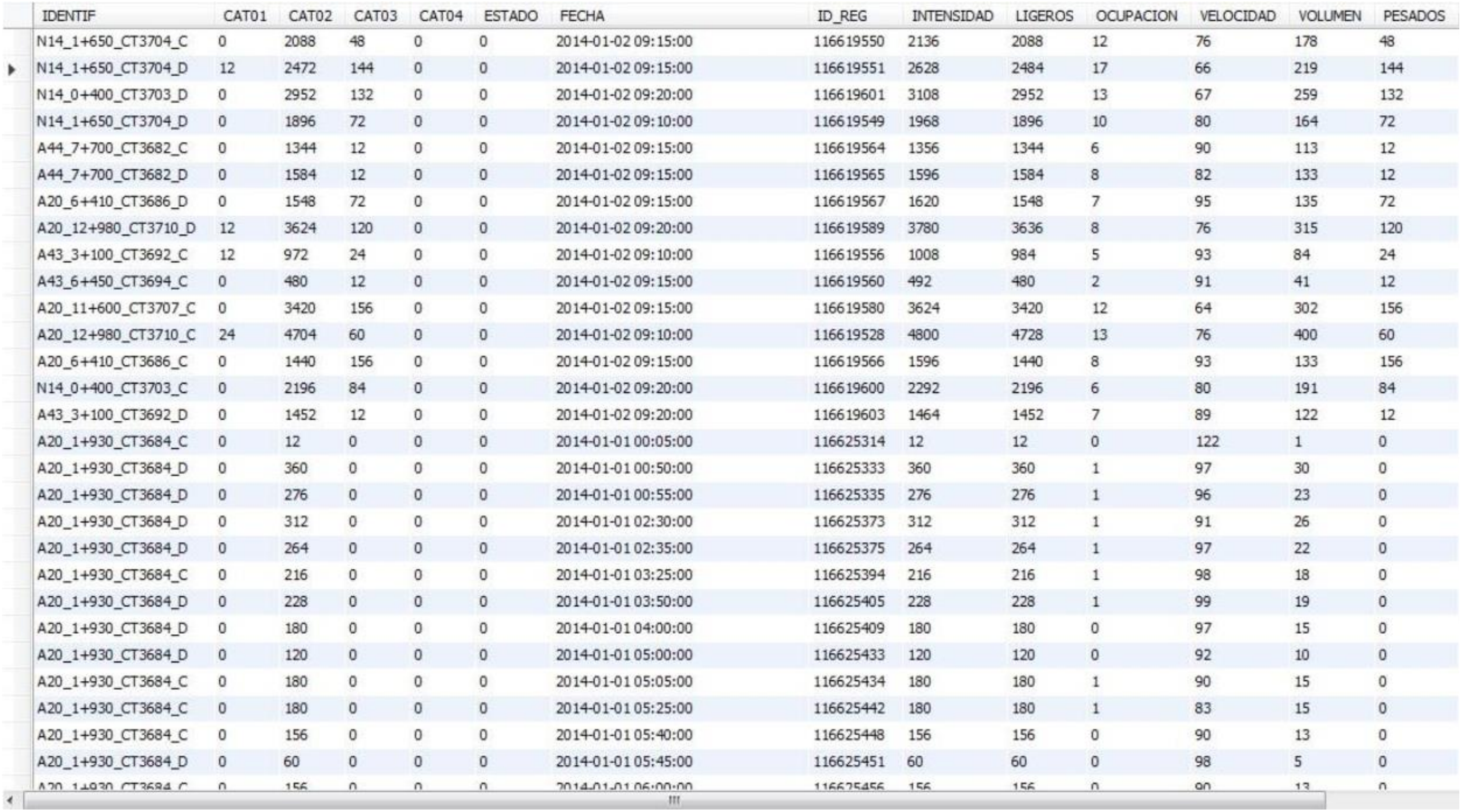


Sensor ID:	**A28_1+050_CT3744_C**						
Row Labels	**Completude**	**Class A**	**Class B**	**Class C**	**Class D**	**Ligeiros**	**Pesados**
Janeiro	100,00%	5745	1361649	90689	4580	1367394	95269
1	100,00%	35	26219	125	43	26254	168
00:00	100,00%	0	16	0	0	16	0
00:05		0	11	1	0	11	1
00:10		0	12	0	0	12	0
00:15		0	27	0	0	27	0
00:20		0	65	0	0	65	0
00:25		0	71	0	0	71	0
00:30		0	86	0	0	86	0
00:35		0	97	0	0	97	0
00:40		0	102	1	0	102	1
00:45		0	113	0	0	113	0
00:50		0	106	0	0	106	0
00:55		0	126	0	0	126	0
01:00	100,00%	0	119	1	0	119	1
01:05		0	102	1	1	102	2
01:10		0	136	0	0	136	0
01:15		0	140	1	1	140	2
01:20		0	132	0	0	132	0
01:25		0	151	0	0	151	0
01:30		0	129	0	0	129	0
01:35		0	121	0	0	121	0
01:40		0	130	1	1	130	2
01:45		0	116	2	0	116	2
01:50		1	127	0	1	128	1
01:55		1	148	0	0	149	0
02:00	100,00%	0	174	0	0	174	0

Parameter	Description
SensorID	The unique ID for the sensor
Row Labels	This column presents the temporal range of each row. The first row, "Janeiro", corresponds to the aggregated values for the entire month of January. The second row represents aggregated values for the 1st of January. The next rows correspond to five-minute time intervals starting at mid-night and ending at 23:55
Completude	The percentage of completeness (real record number vs. expected record number)
Class A	Total number of class 1 vehicles (height of front axis <1,10m)
Class B	Total number of class 2 vehicles (height of front axis >=1,10m)
Class C	Total number of class 3 vehicles (3 axis)
Class D	Total number of class 4 vehicles (more than 4 axis)
Ligeiros	Total number of light vehicles
Pesados	Total number of heavy vehicles

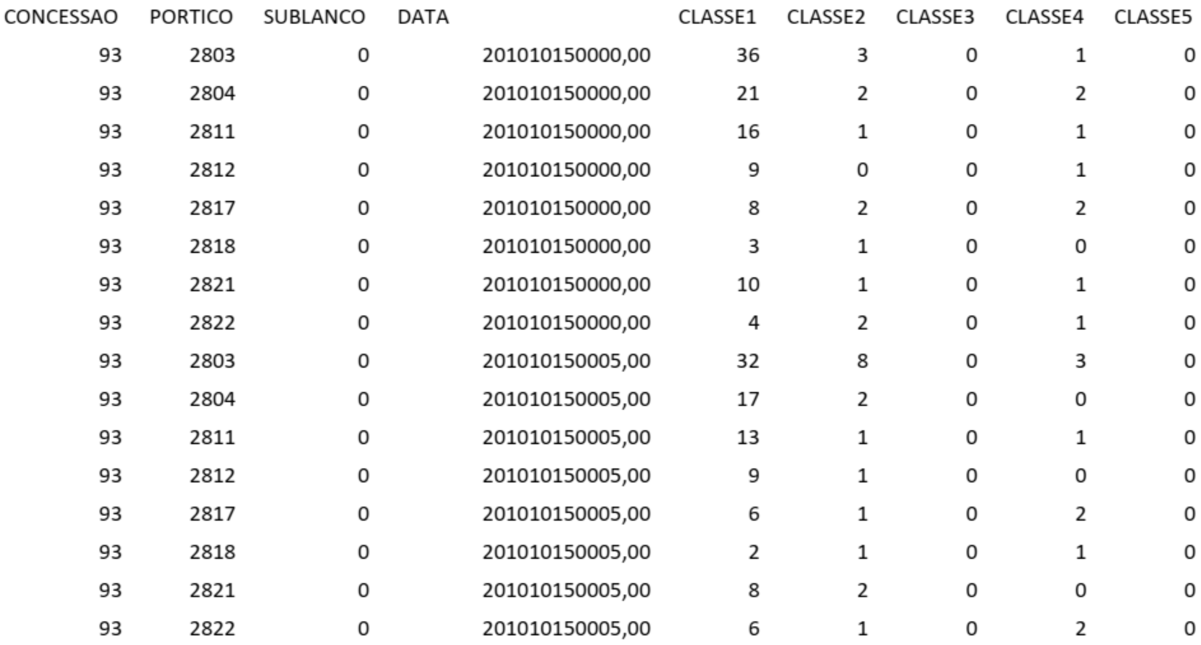
A.9. Example IP Traffic Counter Data Database Dump

Parameter	Description
IDENTIF	The unique ID for the sensor
CAT01	Total number of class 1 vehicles (height of front axis <1,10m)
CAT02	Total number of class 2 vehicles (height of front axis >=1,10m)
CAT03	Total number of class 3 vehicles (3 axis)
CAT04	Total number of class 4 vehicles (more than 4 axis)
ESTADO	Road state ("0": open road; "1": closed road)
FECHA	Date and time of the sensor reading
ID_REG	Unique ID for the sensor reading
INTEN-SIDAD	Traffic intensity
LIGEROS	Total number of light vehicles
OCUPACION	Highway occupation percentage
VELOCIDAD	Average speed
VOLUMEN	Traffic volume
PESADOS	Total number of heavy vehicles



IDENTIF	CAT01	CAT02	CAT03	CAT04	ESTADO	FECHA	ID_REG	INTENSIDAD	LIGEROS	OCUPACION	VELOCIDAD	VOLUMEN	PESADOS
N14_1+650_CT3704_C	0	2088	48	0	0	2014-01-02 09:15:00	116619550	2136	2088	12	76	178	48
N14_1+650_CT3704_D	12	2472	144	0	0	2014-01-02 09:15:00	116619551	2628	2484	17	66	219	144
N14_0+400_CT3703_D	0	2952	132	0	0	2014-01-02 09:20:00	116619601	3108	2952	13	67	259	132
N14_1+650_CT3704_D	0	1896	72	0	0	2014-01-02 09:10:00	116619549	1968	1896	10	80	164	72
A44_7+700_CT3682_C	0	1344	12	0	0	2014-01-02 09:15:00	116619564	1356	1344	6	90	113	12
A44_7+700_CT3682_D	0	1584	12	0	0	2014-01-02 09:15:00	116619565	1596	1584	8	82	133	12
A20_6+410_CT3686_D	0	1548	72	0	0	2014-01-02 09:15:00	116619567	1620	1548	7	95	135	72
A20_12+980_CT3710_D	12	3624	120	0	0	2014-01-02 09:20:00	116619589	3780	3636	8	76	315	120
A43_3+100_CT3692_C	12	972	24	0	0	2014-01-02 09:10:00	116619556	1008	984	5	93	84	24
A43_6+450_CT3694_C	0	480	12	0	0	2014-01-02 09:15:00	116619560	492	480	2	91	41	12
A20_11+600_CT3707_C	0	3420	156	0	0	2014-01-02 09:15:00	116619580	3624	3420	12	64	302	156
A20_12+980_CT3710_C	24	4704	60	0	0	2014-01-02 09:10:00	116619528	4800	4728	13	76	400	60
A20_6+410_CT3686_C	0	1440	156	0	0	2014-01-02 09:15:00	116619566	1596	1440	8	93	133	156
N14_0+400_CT3703_C	0	2196	84	0	0	2014-01-02 09:20:00	116619600	2292	2196	6	80	191	84
A43_3+100_CT3692_D	0	1452	12	0	0	2014-01-02 09:20:00	116619603	1464	1452	7	89	122	12
A20_1+930_CT3684_C	0	12	0	0	0	2014-01-01 00:05:00	116625314	12	12	0	122	1	0
A20_1+930_CT3684_D	0	360	0	0	0	2014-01-01 00:50:00	116625333	360	360	1	97	30	0
A20_1+930_CT3684_D	0	276	0	0	0	2014-01-01 00:55:00	116625335	276	276	1	96	23	0
A20_1+930_CT3684_D	0	312	0	0	0	2014-01-01 02:30:00	116625373	312	312	1	91	26	0
A20_1+930_CT3684_D	0	264	0	0	0	2014-01-01 02:35:00	116625375	264	264	1	97	22	0
A20_1+930_CT3684_C	0	216	0	0	0	2014-01-01 03:25:00	116625394	216	216	1	98	18	0
A20_1+930_CT3684_D	0	228	0	0	0	2014-01-01 03:50:00	116625405	228	228	1	99	19	0
A20_1+930_CT3684_D	0	180	0	0	0	2014-01-01 04:00:00	116625409	180	180	0	97	15	0
A20_1+930_CT3684_D	0	120	0	0	0	2014-01-01 05:00:00	116625433	120	120	0	92	10	0
A20_1+930_CT3684_C	0	180	0	0	0	2014-01-01 05:05:00	116625434	180	180	1	90	15	0
A20_1+930_CT3684_C	0	180	0	0	0	2014-01-01 05:25:00	116625442	180	180	1	83	15	0
A20_1+930_CT3684_C	0	156	0	0	0	2014-01-01 05:40:00	116625448	156	156	0	90	13	0
A20_1+930_CT3684_D	0	60	0	0	0	2014-01-01 05:45:00	116625451	60	60	0	98	5	0
A20_1+930_CT3684_C	0	156	0	0	0	2014-01-01 06:00:00	116625456	156	156	0	90	13	0

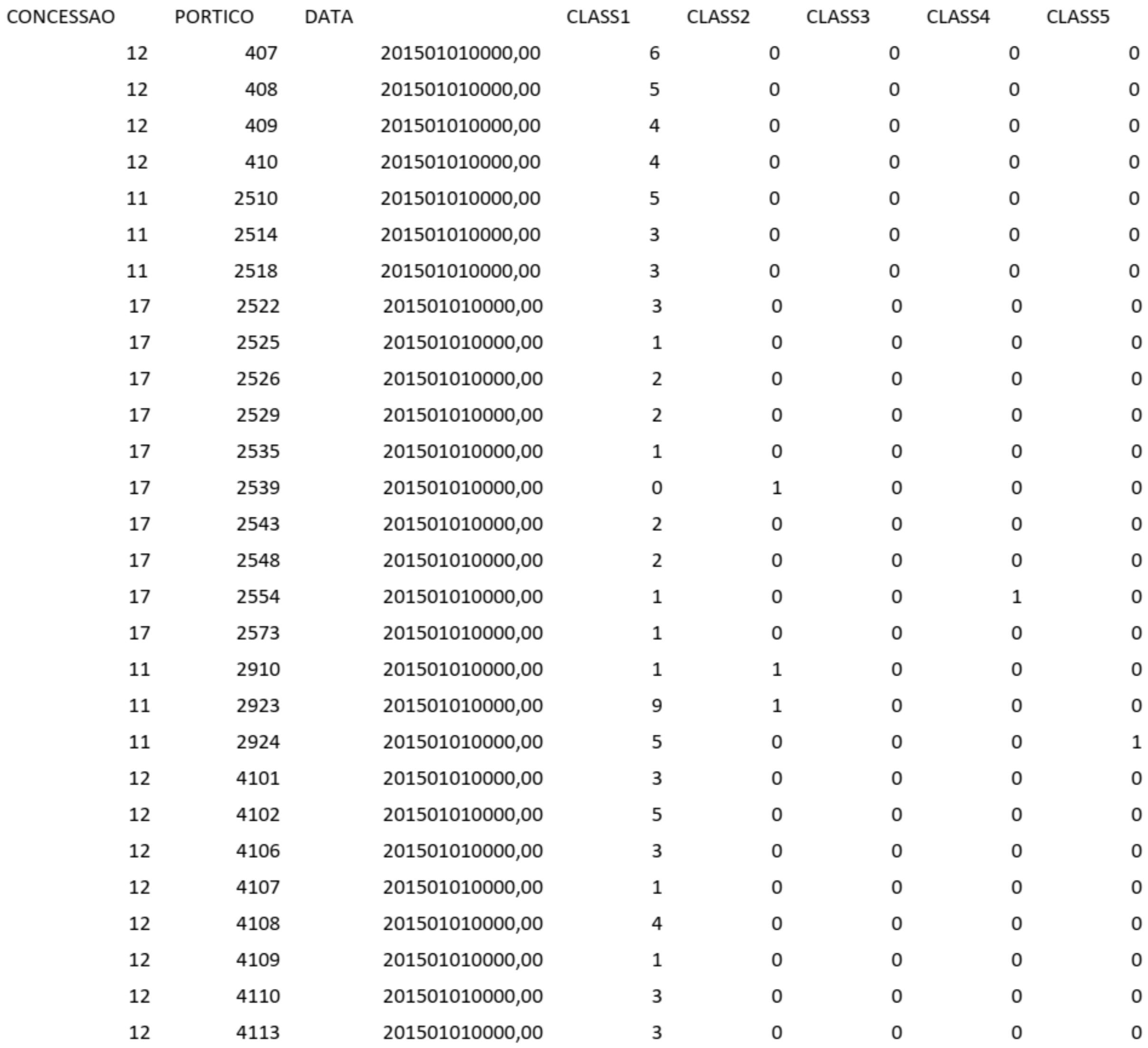
A.10. Example Electronic Toll Sensor Data from Via Livre



CONCESSAO	PORTICO	SUBLANCO	DATA	CLASSE1	CLASSE2	CLASSE3	CLASSE4	CLASSE5
93	2803	0	201010150000,00	36	3	0	1	0
93	2804	0	201010150000,00	21	2	0	2	0
93	2811	0	201010150000,00	16	1	0	1	0
93	2812	0	201010150000,00	9	0	0	1	0
93	2817	0	201010150000,00	8	2	0	2	0
93	2818	0	201010150000,00	3	1	0	0	0
93	2821	0	201010150000,00	10	1	0	1	0
93	2822	0	201010150000,00	4	2	0	1	0
93	2803	0	201010150005,00	32	8	0	3	0
93	2804	0	201010150005,00	17	2	0	0	0
93	2811	0	201010150005,00	13	1	0	1	0
93	2812	0	201010150005,00	9	1	0	0	0
93	2817	0	201010150005,00	6	1	0	2	0
93	2818	0	201010150005,00	2	1	0	1	0
93	2821	0	201010150005,00	8	2	0	0	0
93	2822	0	201010150005,00	6	1	0	2	0

Parameter	Description
CONCESSAO	The unique ID for the sensor
PORTICO	Toll gate ID
SUBLANCO	Highway section
DATA	Date and time for the sensor reading
CLASSE1	Total number of class 1 vehicles (height of front axis <1,10m)
CLASSE2	Total number of class 2 vehicles (height of front axis >=1,10m)
CLASSE3	Total number of class 3 vehicles (3 axis)
CLASSE4	Total number of class 4 vehicles (more than 4 axis)
CLASSE5	Total number of motorcycles with toll charging systems

A.11. Example Electronic Toll Sensor Data from Ascendi

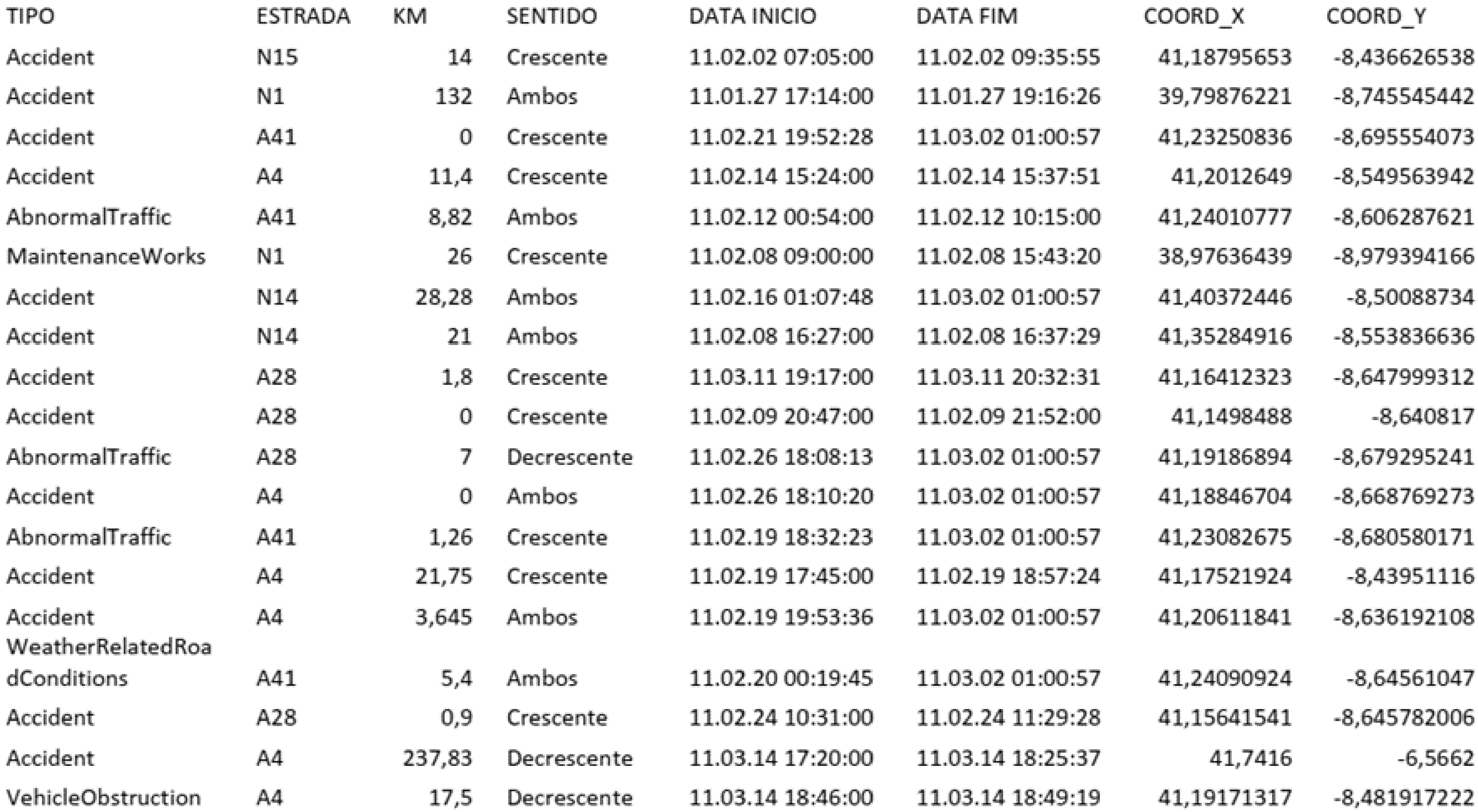


CONCESSAO	PORTICO	DATA	CLASS1	CLASS2	CLASS3	CLASS4	CLASS5
12	407	201501010000,00	6	0	0	0	0
12	408	201501010000,00	5	0	0	0	0
12	409	201501010000,00	4	0	0	0	0
12	410	201501010000,00	4	0	0	0	0
11	2510	201501010000,00	5	0	0	0	0
11	2514	201501010000,00	3	0	0	0	0
11	2518	201501010000,00	3	0	0	0	0
17	2522	201501010000,00	3	0	0	0	0
17	2525	201501010000,00	1	0	0	0	0
17	2526	201501010000,00	2	0	0	0	0
17	2529	201501010000,00	2	0	0	0	0
17	2535	201501010000,00	1	0	0	0	0
17	2539	201501010000,00	0	1	0	0	0
17	2543	201501010000,00	2	0	0	0	0
17	2548	201501010000,00	2	0	0	0	0
17	2554	201501010000,00	1	0	0	1	0
17	2573	201501010000,00	1	0	0	0	0
11	2910	201501010000,00	1	1	0	0	0
11	2923	201501010000,00	9	1	0	0	0
11	2924	201501010000,00	5	0	0	0	1
12	4101	201501010000,00	3	0	0	0	0
12	4102	201501010000,00	5	0	0	0	0
12	4106	201501010000,00	3	0	0	0	0
12	4107	201501010000,00	1	0	0	0	0
12	4108	201501010000,00	4	0	0	0	0
12	4109	201501010000,00	1	0	0	0	0
12	4110	201501010000,00	3	0	0	0	0
12	4113	201501010000,00	3	0	0	0	0

Parameter	Description
CONCESSAO	The unique ID for the sensor
PORTICO	Toll gate ID
DATA	Date and time for the sensor reading
CLASSE1	Total number of class 1 vehicles (height of front axis <1,10m)
CLASSE2	Total number of class 2 vehicles (height of front axis >=1,10m)
CLASSE3	Total number of class 3 vehicles (3 axis)
CLASSE4	Total number of class 4 vehicles (more than 4 axis)
CLASSE5	Total number of motorcycles with toll charging systems

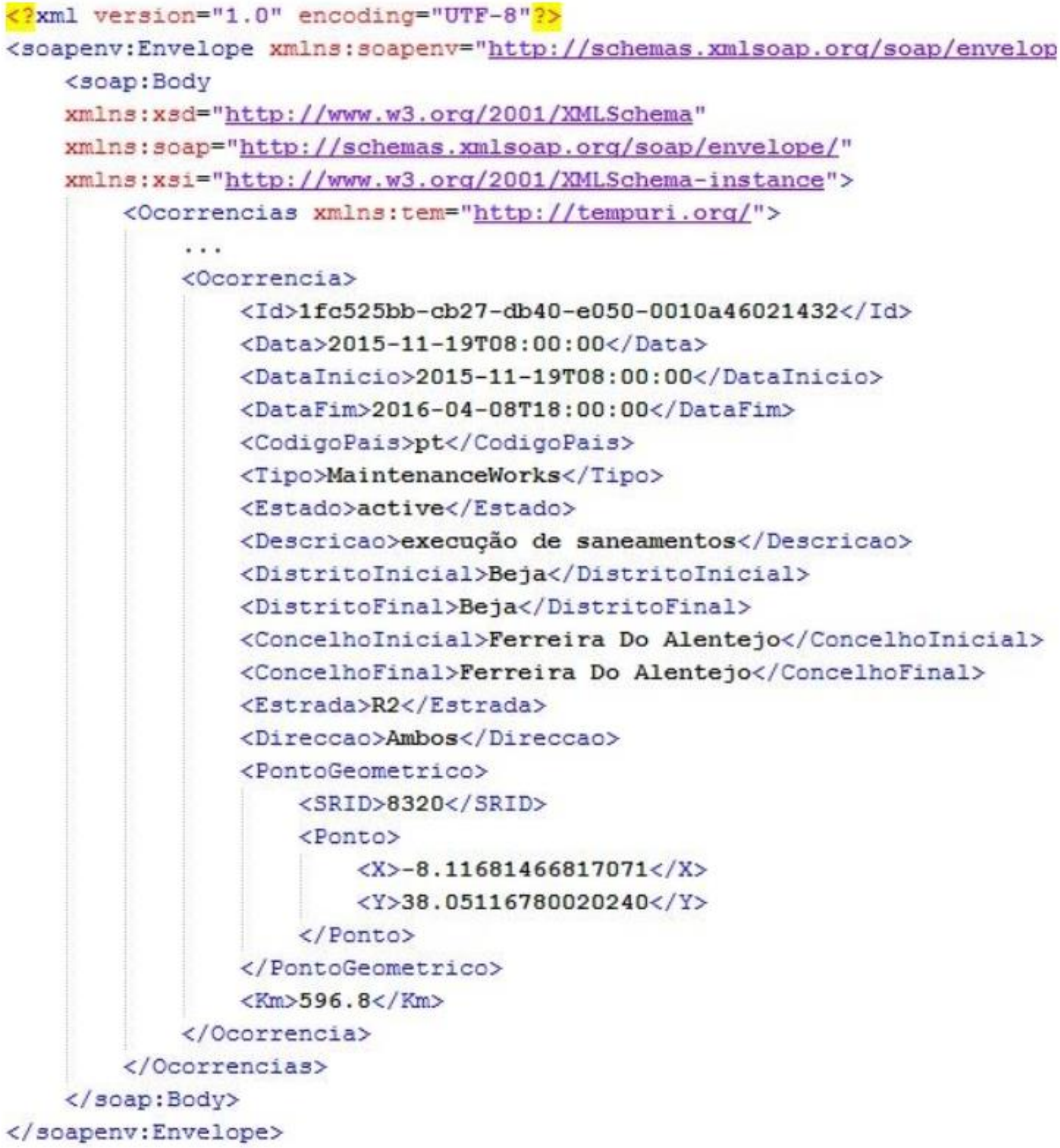
A.12. Example IP Batch Traffic Event Data Database SQL Dump

Parameter	Description
TIPO	DATEX II-based event type
ESTRADA	Road name
KM	Kilometre on which the event happened/started
SENTIDO	Direction ("Crescente": from lowest to highest road kilometre; "Decrescente": from highest to lowest kilometre; "Ambos": both directions)
DATA INICIO	Event starting date and time
DATA FIM	Event ending date and time
COORD_X	Latitude expressed in the World Geodetic System (WGS) coordinate system
COORD_Y	Longitude expressed in the World Geodetic System (WGS) coordinate system



TIPO	ESTRADA	KM	SENTIDO	DATA INICIO	DATA FIM	COORD_X	COORD_Y
Accident	N15	14	Crescente	11.02.02 07:05:00	11.02.02 09:35:55	41,18795653	-8,436626538
Accident	N1	132	Ambos	11.01.27 17:14:00	11.01.27 19:16:26	39,79876221	-8,745545442
Accident	A41	0	Crescente	11.02.21 19:52:28	11.03.02 01:00:57	41,23250836	-8,695554073
Accident	A4	11,4	Crescente	11.02.14 15:24:00	11.02.14 15:37:51	41,2012649	-8,549563942
AbnormalTraffic	A41	8,82	Ambos	11.02.12 00:54:00	11.02.12 10:15:00	41,24010777	-8,606287621
MaintenanceWorks	N1	26	Crescente	11.02.08 09:00:00	11.02.08 15:43:20	38,97636439	-8,979394166
Accident	N14	28,28	Ambos	11.02.16 01:07:48	11.03.02 01:00:57	41,40372446	-8,50088734
Accident	N14	21	Ambos	11.02.08 16:27:00	11.02.08 16:37:29	41,35284916	-8,553836636
Accident	A28	1,8	Crescente	11.03.11 19:17:00	11.03.11 20:32:31	41,16412323	-8,647999312
Accident	A28	0	Crescente	11.02.09 20:47:00	11.02.09 21:52:00	41,1498488	-8,640817
AbnormalTraffic	A28	7	Decrescente	11.02.26 18:08:13	11.03.02 01:00:57	41,19186894	-8,679295241
Accident	A4	0	Ambos	11.02.26 18:10:20	11.03.02 01:00:57	41,18846704	-8,668769273
AbnormalTraffic	A41	1,26	Crescente	11.02.19 18:32:23	11.03.02 01:00:57	41,23082675	-8,680580171
Accident	A4	21,75	Crescente	11.02.19 17:45:00	11.02.19 18:57:24	41,17521924	-8,43951116
Accident	A4	3,645	Ambos	11.02.19 19:53:36	11.03.02 01:00:57	41,20611841	-8,636192108
WeatherRelatedRoadConditions	A41	5,4	Ambos	11.02.20 00:19:45	11.03.02 01:00:57	41,24090924	-8,64561047
Accident	A28	0,9	Crescente	11.02.24 10:31:00	11.02.24 11:29:28	41,15641541	-8,645782006
Accident	A4	237,83	Decrescente	11.03.14 17:20:00	11.03.14 18:25:37	41,7416	-6,5662
VehicleObstruction	A4	17,5	Decrescente	11.03.14 18:46:00	11.03.14 18:49:19	41,19171317	-8,481917222

A.13. Example IP Real-Time Traffic Event Data in XML Format

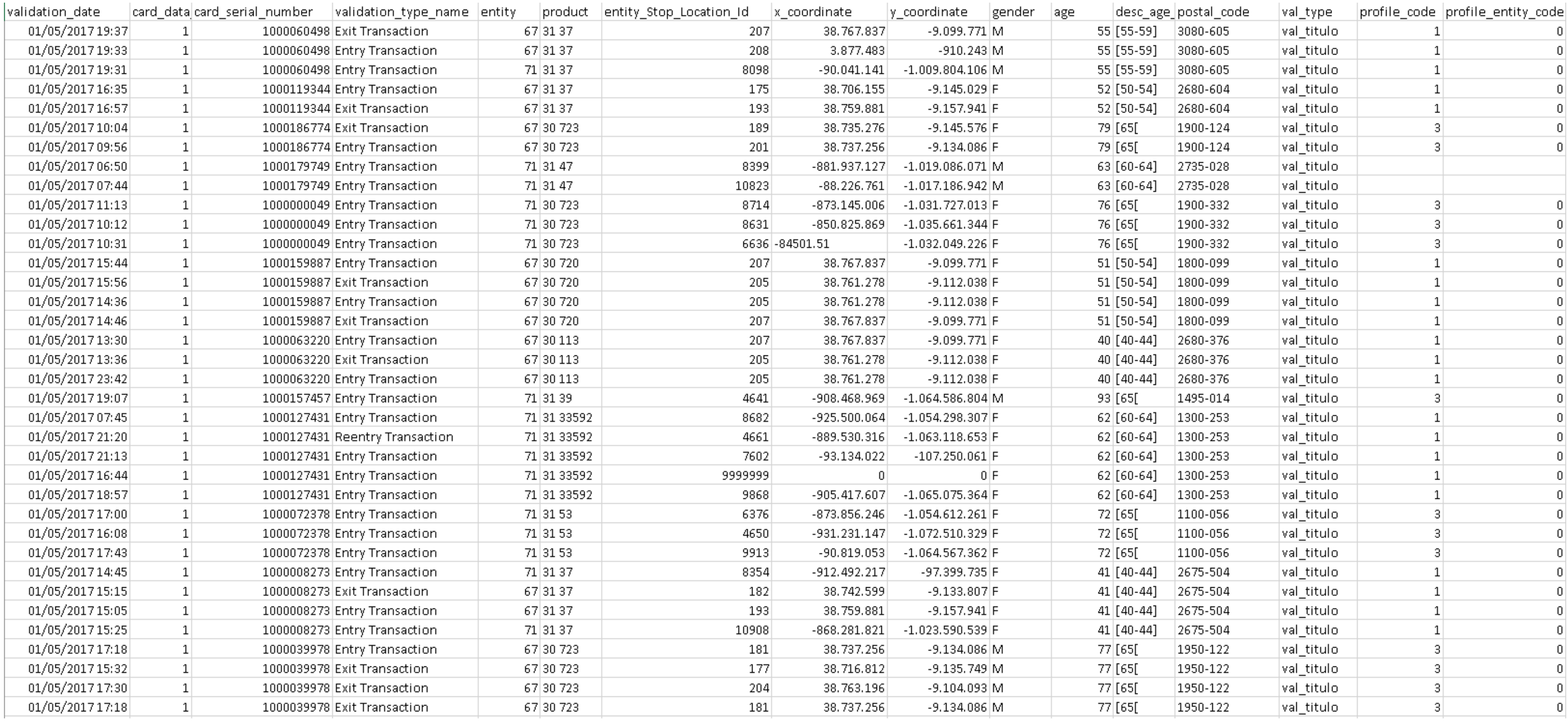


```xml
<?xml version="1.0" encoding="UTF-8"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelop
    <soap:Body
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <Ocorrencias xmlns:tem="http://tempuri.org/">
            ...
            <Ocorrencia>
                <Id>1fc525bb-cb27-db40-e050-0010a46021432</Id>
                <Data>2015-11-19T08:00:00</Data>
                <DataInicio>2015-11-19T08:00:00</DataInicio>
                <DataFim>2016-04-08T18:00:00</DataFim>
                <CodigoPais>pt</CodigoPais>
                <Tipo>MaintenanceWorks</Tipo>
                <Estado>active</Estado>
                <Descricao>execução de saneamentos</Descricao>
                <DistritoInicial>Beja</DistritoInicial>
                <DistritoFinal>Beja</DistritoFinal>
                <ConcelhoInicial>Ferreira Do Alentejo</ConcelhoInicial>
                <ConcelhoFinal>Ferreira Do Alentejo</ConcelhoFinal>
                <Estrada>R2</Estrada>
                <Direccao>Ambos</Direccao>
                <PontoGeometrico>
                    <SRID>8320</SRID>
                    <Ponto>
                        <X>-8.11681466817071</X>
                        <Y>38.05116780020240</Y>
                    </Ponto>
                </PontoGeometrico>
                <Km>596.8</Km>
            </Ocorrencia>
        </Ocorrencias>
    </soap:Body>
</soapenv:Envelope>
```

Parameter	Description
Id	Unique ID for the event
Data	Event date and time
DataInicio	Event starting date and time
DataFim	Event ending date and time
CodigoPais	Country code
Tipo	DATEX II-based event type
Estado	Event status
Descricao	Event description
DistritoInicial	Township in which the event started
DistritoFinal	Township in which the event ended
ConcelhoInicial	Municipality in which the event started
ConcelhoFinal	Municipality in which the event ended
Estrada	Road name
Direcao	Direction ("Crescente": from lowest to highest road kilometre; "Decrescente": from highest to lowest kilometre; "Ambos": both directions)
X	Latitude expressed in the World Geodetic System (WGS) coordinate system
Y	Longitude expressed in the World Geodetic System (WGS) coordinate system
Km	Kilometre on which the event happened/started

A.14. Example OTLIS Ticket Validation Data in Excel Format

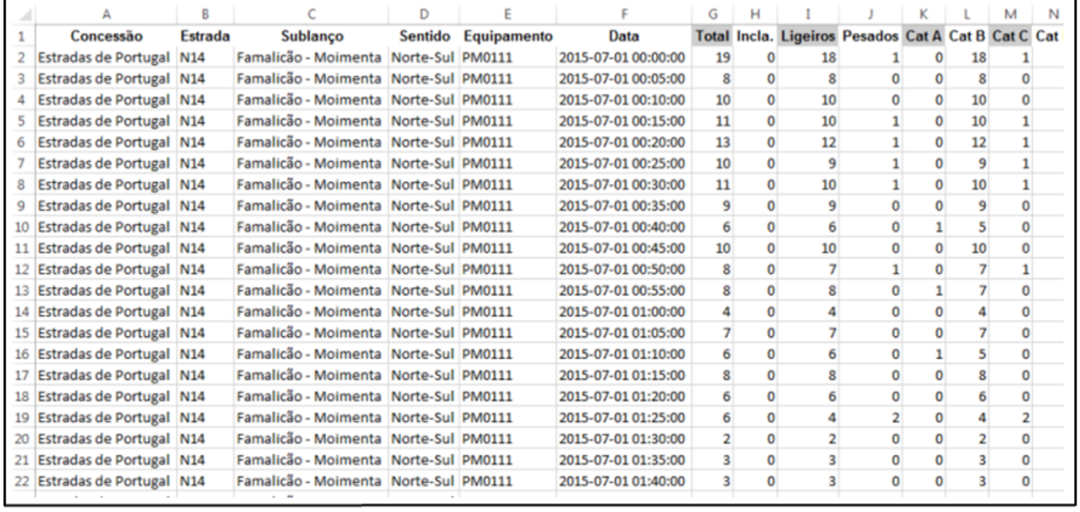
Parameter	Description
validation_date	Validation date
card_data	Flag parameter indicating that the record has ticket/card data
card_serial_number	Unique serial number for the validated card
validation_type_name	Type of validation (Entry, Exit, Reentry)
entity	Entity code for the public transport operator
product	Type of card product (e.g. elderly, children, tourism)
entity_stop_location_id	Unique ID for the stop/station location
x_coordinate	Latitude expressed in the World Geodetic System (WGS) coordinate system
y_coordinate	Longitude expressed in the World Geodetic System (WGS) coordinate system
gender	Card holder's gender
age	Card holder's age
desc_age	Card holder's age group
postal_code	Card holder's residency postal code
val_type	Validation type (profile, ticket)
profile_code	Card holder's profile code (for profile validations)
profile_entity_code	Card's holder's profile entity code (for profile validations, e.g., child, third age, retired, military, student, etc.)



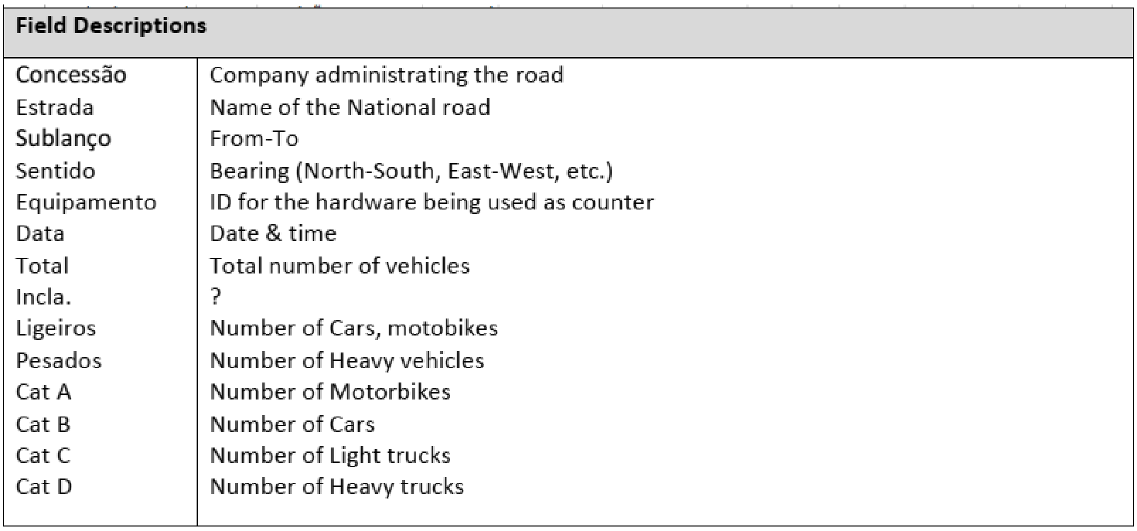
validation_date	card_data	card_serial_number	validation_type_name	entity	product	entity_Stop_Location_Id	x_coordinate	y_coordinate	gender	age	desc_age_	postal_code	val_type	profile_code	profile_entity_code
01/05/2017 19:37	1	1000060498	Exit Transaction	67	31 37	207	38.767.837	-9.099.771	M	55	[55-59]	3080-605	val_titulo	1	0
01/05/2017 19:33	1	1000060498	Entry Transaction	67	31 37	208	3.877.483	-910.243	M	55	[55-59]	3080-605	val_titulo	1	0
01/05/2017 19:31	1	1000060498	Entry Transaction	71	31 37	8098	-90.041.141	-1.009.804.106	M	55	[55-59]	3080-605	val_titulo	1	0
01/05/2017 16:35	1	1000119344	Entry Transaction	67	31 37	175	38.706.155	-9.145.029	F	52	[50-54]	2680-604	val_titulo	1	0
01/05/2017 16:57	1	1000119344	Exit Transaction	67	31 37	193	38.759.881	-9.157.941	F	52	[50-54]	2680-604	val_titulo	1	0
01/05/2017 10:04	1	1000186774	Exit Transaction	67	30 723	189	38.735.276	-9.145.576	F	79	[65[	1900-124	val_titulo	3	0
01/05/2017 09:56	1	1000186774	Entry Transaction	67	30 723	201	38.737.256	-9.134.086	F	79	[65[	1900-124	val_titulo	3	0
01/05/2017 06:50	1	1000179749	Entry Transaction	71	31 47	8399	-881.937.127	-1.019.086.071	M	63	[60-64]	2735-028	val_titulo		
01/05/2017 07:44	1	1000179749	Entry Transaction	71	31 47	10823	-88.226.761	-1.017.186.942	M	63	[60-64]	2735-028	val_titulo		
01/05/2017 11:13	1	1000000049	Entry Transaction	71	30 723	8714	-873.145.006	-1.031.727.013	F	76	[65[	1900-332	val_titulo	3	0
01/05/2017 10:12	1	1000000049	Entry Transaction	71	30 723	8631	-850.825.869	-1.035.661.344	F	76	[65[	1900-332	val_titulo	3	0
01/05/2017 10:31	1	1000000049	Entry Transaction	71	30 723	6636	-84501.51	-1.032.049.226	F	76	[65[	1900-332	val_titulo	3	0
01/05/2017 15:44	1	1000159887	Entry Transaction	67	30 720	207	38.767.837	-9.099.771	F	51	[50-54]	1800-099	val_titulo	1	0
01/05/2017 15:56	1	1000159887	Exit Transaction	67	30 720	205	38.761.278	-9.112.038	F	51	[50-54]	1800-099	val_titulo	1	0
01/05/2017 14:36	1	1000159887	Entry Transaction	67	30 720	205	38.761.278	-9.112.038	F	51	[50-54]	1800-099	val_titulo	1	0
01/05/2017 14:46	1	1000159887	Exit Transaction	67	30 720	207	38.767.837	-9.099.771	F	51	[50-54]	1800-099	val_titulo	1	0
01/05/2017 13:30	1	1000063220	Entry Transaction	67	30 113	207	38.767.837	-9.099.771	F	40	[40-44]	2680-376	val_titulo	1	0
01/05/2017 13:36	1	1000063220	Exit Transaction	67	30 113	205	38.761.278	-9.112.038	F	40	[40-44]	2680-376	val_titulo	1	0
01/05/2017 23:42	1	1000063220	Entry Transaction	67	30 113	205	38.761.278	-9.112.038	F	40	[40-44]	2680-376	val_titulo	1	0
01/05/2017 19:07	1	1000157457	Entry Transaction	71	31 39	4641	-908.468.969	-1.064.586.804	M	93	[65[	1495-014	val_titulo	3	0
01/05/2017 07:45	1	1000127431	Entry Transaction	71	31 33592	8682	-925.500.064	-1.054.298.307	F	62	[60-64]	1300-253	val_titulo	1	0
01/05/2017 21:20	1	1000127431	Reentry Transaction	71	31 33592	4661	-889.530.316	-1.063.118.653	F	62	[60-64]	1300-253	val_titulo	1	0
01/05/2017 21:13	1	1000127431	Entry Transaction	71	31 33592	7602	-93.134.022	-107.250.061	F	62	[60-64]	1300-253	val_titulo	1	0
01/05/2017 16:44	1	1000127431	Entry Transaction	71	31 33592	9999999	0	0	F	62	[60-64]	1300-253	val_titulo	1	0
01/05/2017 18:57	1	1000127431	Entry Transaction	71	31 33592	9868	-905.417.607	-1.065.075.364	F	62	[60-64]	1300-253	val_titulo	1	0
01/05/2017 17:00	1	1000072378	Entry Transaction	71	31 53	6376	-873.856.246	-1.054.612.261	F	72	[65[	1100-056	val_titulo	3	0
01/05/2017 16:08	1	1000072378	Entry Transaction	71	31 53	4650	-931.231.147	-1.072.510.329	F	72	[65[	1100-056	val_titulo	3	0
01/05/2017 17:43	1	1000072378	Entry Transaction	71	31 53	9913	-90.819.053	-1.064.567.362	F	72	[65[	1100-056	val_titulo	3	0
01/05/2017 14:45	1	1000008273	Entry Transaction	71	31 37	8354	-912.492.217	-97.399.735	F	41	[40-44]	2675-504	val_titulo	1	0
01/05/2017 15:15	1	1000008273	Exit Transaction	67	31 37	182	38.742.599	-9.133.807	F	41	[40-44]	2675-504	val_titulo	1	0
01/05/2017 15:05	1	1000008273	Entry Transaction	67	31 37	193	38.759.881	-9.157.941	F	41	[40-44]	2675-504	val_titulo	1	0
01/05/2017 15:25	1	1000008273	Entry Transaction	71	31 37	10908	-868.281.821	-1.023.590.539	F	41	[40-44]	2675-504	val_titulo	1	0
01/05/2017 17:18	1	1000039978	Entry Transaction	67	30 723	181	38.737.256	-9.134.086	M	77	[65[	1950-122	val_titulo	3	0
01/05/2017 15:32	1	1000039978	Exit Transaction	67	30 723	177	38.716.812	-9.135.749	M	77	[65[	1950-122	val_titulo	3	0
01/05/2017 17:30	1	1000039978	Exit Transaction	67	30 723	204	38.763.196	-9.104.093	M	77	[65[	1950-122	val_titulo	3	0
01/05/2017 17:18	1	1000039978	Exit Transaction	67	30 723	181	38.737.256	-9.134.086	M	77	[65[	1950-122	val_titulo	3	0

A.15. Example Data Source Exploration Report for Data Source DTC_POR_RoadVolume

Overview	
Data source acronym	relatorio_Volume_[Road]_[From]-[To]_5MIn
Document version	V0.3
Provider	
Data provider	Infrastruturas de Portugal
Data provider URI	N.A.
Ownership	Infrastruturas de Portugal
Data administrator (if different)	Infrastruturas de Portugal
Permission status	Granted
Partner	
Pilot case	Portuguese
Responsible person	N.A.
Possible scenario coverage	
N.A.	
Details	
Data source description	
Counter Data on National Roads (5 minute granularity)	
Data type (standard if possible)	Excel Sheets (.xslx)
Standard	N.A.
Direct data URI	N.A.
Data Size	N.A.
Sample size	~18000 records for one segment of one road, between 01/07/2015 and 31/07/2015
Data lifetime	Years
Availability	Available
Data collection frequency	5 minutes (or 1 minute, depending on the hardware used)
Data quality	Excellent
Raw data sample	
N.A.	
Print screen (if possible)	



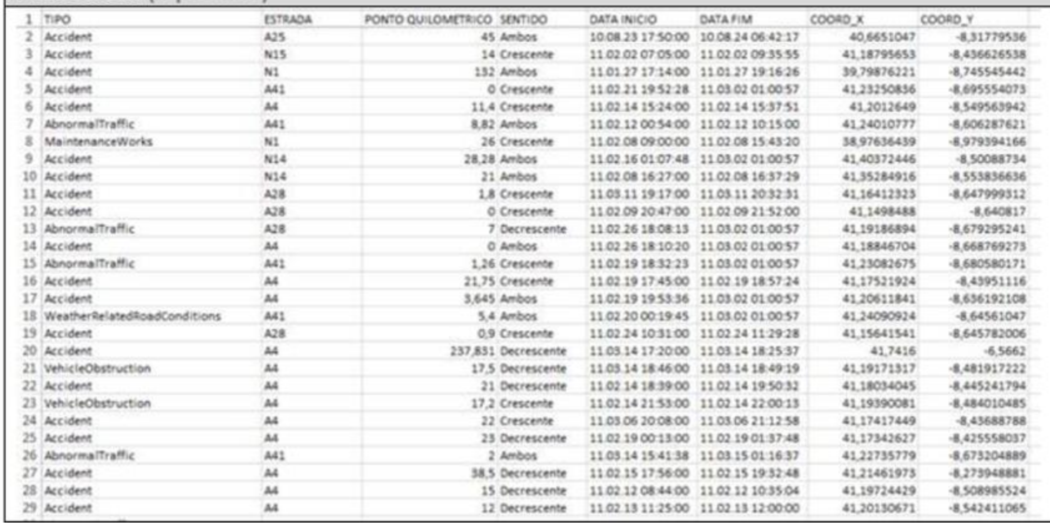
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Concessão	Estrada	Sublanço	Sentido	Equipamento	Data	Total	Incla.	Ligeiros	Pesados	Cat A	Cat B	Cat C	Cat
2	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:00:00	19	0	18	1	0	18	1	
3	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:05:00	8	0	8	0	0	8	0	
4	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:10:00	10	0	10	0	0	10	0	
5	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:15:00	11	0	10	1	0	10	1	
6	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:20:00	13	0	12	1	0	12	1	
7	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:25:00	10	0	9	1	0	9	1	
8	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:30:00	11	0	10	1	0	10	1	
9	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:35:00	9	0	9	0	0	9	0	
10	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:40:00	6	0	6	0	1	5	0	
11	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:45:00	10	0	10	0	0	10	0	
12	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:50:00	8	0	7	1	0	7	1	
13	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 00:55:00	8	0	8	0	1	7	0	
14	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 01:00:00	4	0	4	0	0	4	0	
15	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 01:05:00	7	0	7	0	0	7	0	
16	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 01:10:00	6	0	6	0	1	5	0	
17	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 01:15:00	8	0	8	0	0	8	0	
18	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 01:20:00	6	0	6	0	0	6	0	
19	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 01:25:00	6	0	4	2	0	4	2	
20	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 01:30:00	2	0	2	0	0	2	0	
21	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 01:35:00	3	0	3	0	0	3	0	
22	Estradas de Portugal	N14	Famalicão - Moimenta	Norte-Sul	PM0111	2015-07-01 01:40:00	3	0	3	0	0	3	0	



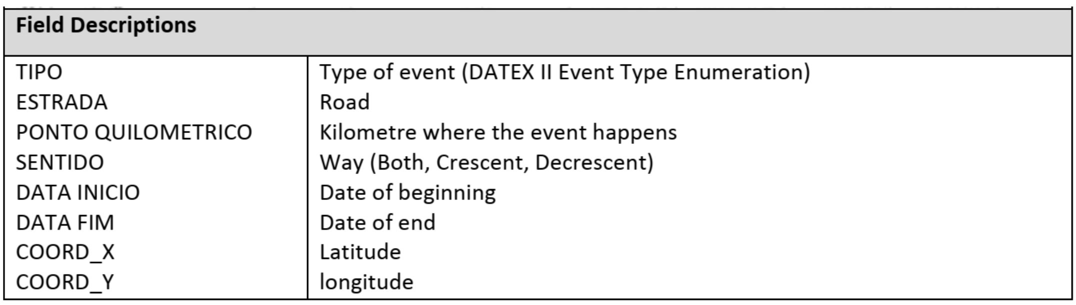
Field Descriptions	
Concessão	Company administrating the road
Estrada	Name of the National road
Sublanço	From-To
Sentido	Bearing (North-South, East-West, etc.)
Equipamento	ID for the hardware being used as counter
Data	Date & time
Total	Total number of vehicles
Incla.	?
Ligeiros	Number of Cars, motobikes
Pesados	Number of Heavy vehicles
Cat A	Number of Motorbikes
Cat B	Number of Cars
Cat C	Number of Light trucks
Cat D	Number of Heavy trucks

A.16. Example Data Source Exploration Report for Data Source DTC_POR_TrafficEventsDB

Overview	
Data source acronym	Traffic Events Database
Document version	V0.1
Provider	
Data provider	Infrastruturas de Portugal
Data provider URI	N.A.
Ownership	Infrastruturas de Portugal
Data administrator (if different)	Infrastruturas de Portugal
Permission status	Granted
Partner	
Pilot case	Portuguese
Responsible person	N.A.
Possible scenario coverage	
N.A.	
Details	
Data source description	
Traffic Events Database	
Data type	XLSX (Possible SQL)
Standard	DATEXII (Event Types)
Direct data URI	N.A.
Data Size	N.A.
Sample size	3224 records for Pilot network roads and 7838 records for complementary roads
Data lifetime	Years
Availability	Available
Data collection frequency	On Demand
Data quality	Great
Raw data sample	
N.A.	
Print screen (if possible)	



1	TIPO	ESTRADA	PONTO QUILOMETRICO	SENTIDO	DATA INICIO	DATA FIM	COORD_X	COORD_Y
2	Accident	A25	45	Ambos	10.08.23 17:50:00	10.08.24 06:42:17	40,6651047	-8,31779536
3	Accident	N15	14	Crescente	11.02.02 07:05:00	11.02.02 09:35:55	41,18795653	-8,436626538
4	Accident	N1	132	Ambos	11.01.27 17:14:00	11.01.27 19:16:26	39,79876221	-8,745545442
5	Accident	A41	0	Crescente	11.02.21 19:52:28	11.03.02 01:00:57	41,23250836	-8,695554073
6	Accident	A4	11,4	Crescente	11.02.14 15:24:00	11.02.14 15:37:51	41,2012649	-8,549563942
7	AbnormalTraffic	A41	8,82	Ambos	11.02.12 00:54:00	11.02.12 10:15:00	41,24010777	-8,606287621
8	MaintenanceWorks	N1	26	Crescente	11.02.08 09:00:00	11.02.08 15:43:20	38,97636439	-8,979394166
9	Accident	N14	28,28	Ambos	11.02.16 01:07:48	11.03.02 01:00:57	41,40372446	-8,50088734
10	Accident	N14	21	Ambos	11.02.08 16:27:00	11.02.08 16:37:29	41,35284916	-8,553836636
11	Accident	A28	1,8	Crescente	11.03.11 19:17:00	11.03.11 20:32:31	41,16412323	-8,647999312
12	Accident	A28	0	Crescente	11.02.09 20:47:00	11.02.09 21:52:00	41,1498488	-8,640817
13	AbnormalTraffic	A28	7	Decrescente	11.02.26 18:08:13	11.03.02 01:00:57	41,19186894	-8,679295241
14	Accident	A4	0	Ambos	11.02.26 18:10:20	11.03.02 01:00:57	41,18846704	-8,668769273
15	AbnormalTraffic	A41	1,26	Crescente	11.02.19 18:32:23	11.03.02 01:00:57	41,23082675	-8,680580171
16	Accident	A4	21,75	Crescente	11.02.19 17:45:00	11.02.19 18:57:24	41,17521924	-8,43951116
17	Accident	A4	3,645	Ambos	11.02.19 19:53:36	11.03.02 01:00:57	41,20611841	-8,636192108
18	WeatherRelatedRoadConditions	A41	5,4	Ambos	11.02.20 00:19:45	11.03.02 01:00:57	41,24090924	-8,64561047
19	Accident	A28	0,9	Crescente	11.02.24 10:31:00	11.02.24 11:29:28	41,15641541	-8,645782006
20	Accident	A4	237,831	Decrescente	11.03.14 17:20:00	11.03.14 18:25:37	41,7416	-6,5662
21	VehicleObstruction	A4	17,5	Decrescente	11.03.14 18:46:00	11.03.14 18:49:19	41,19171317	-8,481917222
22	Accident	A4	21	Decrescente	11.02.14 18:39:00	11.02.14 19:50:32	41,18034045	-8,445241794
23	VehicleObstruction	A4	17,2	Crescente	11.02.14 21:53:00	11.02.14 22:00:13	41,19390081	-8,484010485
24	Accident	A4	22	Crescente	11.03.06 20:08:00	11.03.06 21:12:58	41,17417449	-8,43688788
25	Accident	A4	23	Decrescente	11.02.19 00:13:00	11.02.19 01:37:48	41,17342627	-8,425558037
26	AbnormalTraffic	A41	2	Ambos	11.03.14 15:41:38	11.03.15 01:16:37	41,22735779	-8,673204889
27	Accident	A4	38,5	Decrescente	11.02.15 17:56:00	11.02.15 19:32:48	41,21461973	-8,273948881
28	Accident	A4	15	Decrescente	11.02.12 08:44:00	11.02.12 10:35:04	41,19724429	-8,508985524
29	Accident	A4	12	Decrescente	11.02.13 11:25:00	11.02.13 12:00:00	41,20130671	-8,542411065



Field Descriptions	
TIPO	Type of event (DATEX II Event Type Enumeration)
ESTRADA	Road
PONTO QUILOMETRICO	Kilometre where the event happens
SENTIDO	Way (Both, Crescent, Decrescent)
DATA INICIO	Date of beginning
DATA FIM	Date of end
COORD_X	Latitude
COORD_Y	longitude

A.17. Common Data Parameters between all Data Sources Divided by Structure

DATA STRUCTURE	PARAMETER	TYPE	DATA SOURCES
ROAD SENSOR VALUE	valueId	Integer	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	sensorId	String	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	dateTime	Timestamp	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	numberOfClass1Vehicles	Integer	1, 2, 3
	numberOfClass2Vehicles	Integer	1, 2, 3
	numberOfClass3Vehicles	Integer	1, 2, 3
	numberOfClass4Vehicles	Integer	1, 2, 3
	numberOfClass5Vehicles	Integer	1, 2, 3
	totalNumberOfVehicles	Integer	1, 2, 3, 38
	gapBetweenVehicles	Float	38, 44
	totalPerHour	Integer	38
	occupancy	Integer	13, 38, 44

	averageSpeed	Integer Float	9, 12, 38, 44
	occupancyTrend	StringEnum (rising, falling, equal)	13
	occupancyStatus	StringEnum (colors)	13
	flow	Float/Integer	11, 44
	flowTrend	StringEnum (rising, falling, equal)	11
	flowStatus	StringEnum (colors)	11
	congestion	Float/Integer	10
	congestionTrend	StringEnum (rising, falling, equal)	10
	congestionStatus	StringEnum (colors)	10
	averageSpeedTrend	StringEnum (rising, falling, equal)	9, 12
	averageSpeedStatus	StringEnum (colors)	9, 12
	travelTime	Float/Integer	12, 14, 45
	travelTimeTrend	StringEnum (rising, falling, equal)	12, 14
	travelTimeStatus	StringEnum (colors)	12, 14
ROAD SENSOR METADATA	sensorId	String	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	roadType	String Enum (highway, nationalRoad)	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	road	String	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	concession	String	1, 2, 3

	section	String	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	bearing	Integer (Angle) String Enum	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	latitude	Double	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	longitude	Double	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	country	String	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	frequency	Integer	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	sensorType	String	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
	source	String	1, 2, 3, 9, 10, 11, 12, 13, 14, 38, 44, 45
TRAFFIC EVENT	eventId	Integer	5, 34, 41, 43
	road	String	5, 34, 41, 43
	section	String	5, 34, 41, 43
	type	String	5, 34, 41, 43
	bearing	String	5, 34
	latitude	Double	5, 34, 41, 43
	longitude	Double	5, 34, 41, 43
	country	String	5, 34, 41, 43
	startDateTime	Timestamp	5, 34, 41, 43
	endDateTime	Timestamp	5, 34, 41
	description	String	34
	source	String	5, 34, 41, 43
	version	String	5, 34, 41, 43
	severity	String	34

A.18. Mapping Common Data Parameters to DATEX II Model Parameters

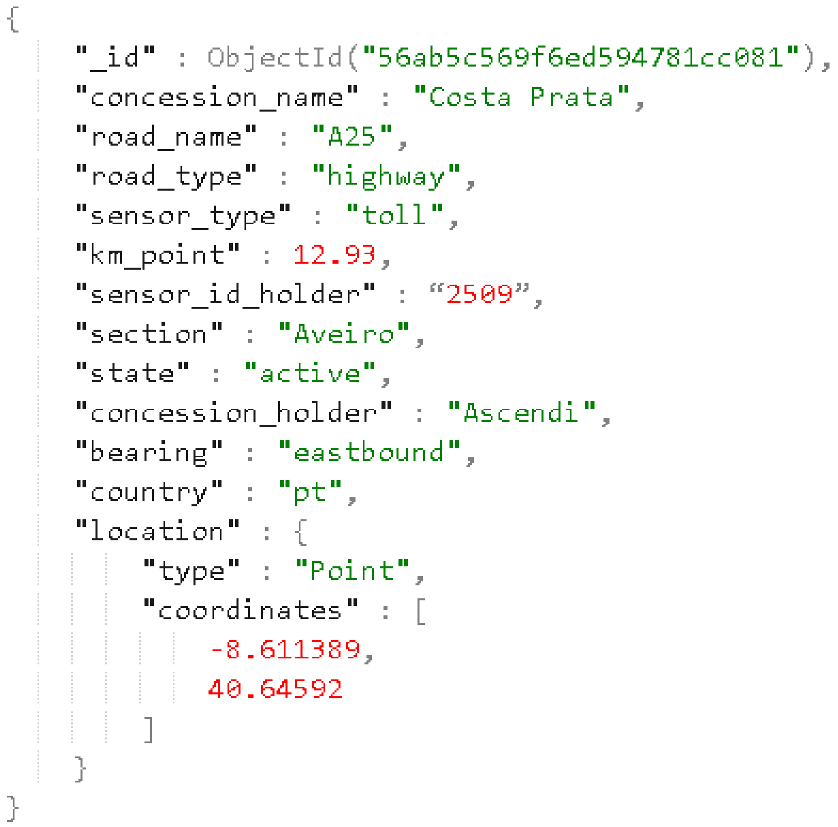
DATA STRUCTURE	PARAMETER	STANDARD CORRESPONDENCE
ROAD SENSOR VALUE	valueId	*Not applicable (N.A.)*
	sensorId	datex:SiteMeasurements(measurementSiteReference)
	dateTime	datex:DateTimeValue(dateTime)*
	numberOfClass1Vehicles	*Needs "Level B" extension*
	numberOfClass2Vehicles	*Needs "Level B" extension*
	numberOfClass3Vehicles	*Needs "Level B" extension*
	numberOfClass4Vehicles	*Needs "Level B" extension*
	numberOfClass5Vehicles	*Needs "Level B" extension*
	totalNumberOfVehicles	datex:TrafficFlow(vehicleFlowValue)
	gapBetweenVehicles	datex:TrafficHeadway(averageDistanceHeadway)
	totalPerHour	datex:TrafficFlow(vehicleFlowValue)
	occupancy	datex:TrafficConcentration (occupancy)
	averageSpeed	datex:TrafficSpeed(averageVehicleSpeed)
		datex:MeasuredOrDerivedDataTypeEnum(trafficSpeed)
	occupancyTrend	*N.A.*
	occupancyStatus	*N.A.*
	flow	datex:TrafficFlow(vehicleFlowValue)
	flowTrend	*N.A.*
	flowStatus	*N.A.*
	congestion	datex:TrafficConcentration (concentration)
	congestionTrend	*N.A.*

	congestionStatus	*N.A.*
	averageSpeedTrend	*N.A.*
	averageSpeedStatus	*N.A.*
	travelTime	datex:TravelTimeData(travelTime)
	travelTimeTrend	datex:TravelTimeData(travelTimeTrendType)
	travelTimeStatus	*N.A.*
	average_speed_lane1	datex:TrafficSpeed(averageVehicleSpeed)
	traffic_headway_lane1	datex:TrafficHeadway(averageDistanceHeadway)
	traffic_concentration_lane1	datex:TrafficConcentration (occupancy)
	flow_lane1_uk_vehicle_class1	*Needs "Level B" extension*
	flow_lane1_uk_vehicle_class2	*Needs "Level B" extension*
	flow_lane1_uk_vehicle_class3	*Needs "Level B" extension*
	flow_lane1_uk_vehicle_class4	*Needs "Level B" extension*
	flow_lane1	datex:TrafficFlow(vehicleFlowValue)
	average_speed_lane2	datex:TrafficSpeed(averageVehicleSpeed)
	traffic_headway_lane2	datex:TrafficHeadway(averageDistanceHeadway)
	traffic_concentration_lane2	datex:TrafficConcentration (occupancy)
	flow_lane2_uk_vehicle_class1	*Needs "Level B" extension*
	flow_lane2_uk_vehicle_class2	*Needs "Level B" extension*
	flow_lane2_uk_vehicle_class3	*Needs "Level B" extension*
	flow_lane2_uk_vehicle_class4	*Needs "Level B" extension*

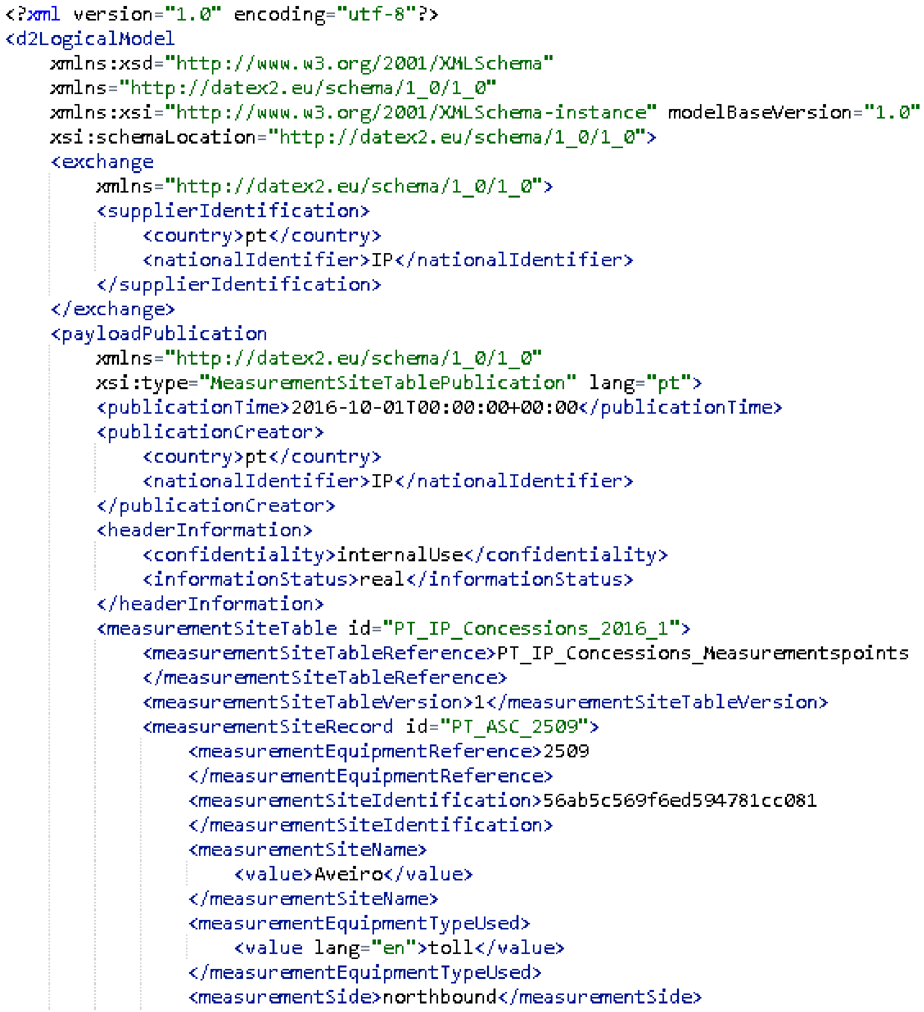
	flow_lane2	datex:TrafficFlow(vehicleFlowValue)
	average_speed_lane3	datex:TrafficSpeed(averageVehicleSpeed)
	traffic_headway_lane3	datex:TrafficHeadway(averageDistanceHeadway)
	traffic_concentration_lane3	datex:TrafficConcentration (occupancy)
	flow_lane3_uk_vehicle_class1	*Needs "Level B" extension*
	flow_lane3_uk_vehicle_class2	*Needs "Level B" extension*
	flow_lane3_uk_vehicle_class3	*Needs "Level B" extension*
	flow_lane3_uk_vehicle_class4	*Needs "Level B" extension*
	flow_lane3	datex:TrafficFlow(vehicleFlowValue)
	average_speed_lane4	datex:TrafficSpeed(averageVehicleSpeed)
	traffic_headway_lane4	datex:TrafficHeadway(averageDistanceHeadway)
	traffic_concentration_lane4	datex:TrafficConcentration (occupancy)
	flow_lane4_uk_vehicle_class1	*Needs "Level B" extension*
	flow_lane4_uk_vehicle_class2	*Needs "Level B" extension*
	flow_lane4_uk_vehicle_class3	*Needs "Level B" extension*
	flow_lane4_uk_vehicle_class4	*Needs "Level B" extension*
	flow_lane4	datex:TrafficFlow(vehicleFlowValue)
ROAD SENSOR METADATA	sensorId	datex:measurementSiteRecord(id)
	roadType	datex:measurementSiteRecord(affectedCarriagewayAndLanes)
	lane	datex:measurementSiteRecord(affectedCarriagewayAndLanes)

	road	*N.A.*
	concession	datex:nationalIdentifier
	section	*N.A.*
	bearing	datex:DirectionBearingValue(directionBearing)
		datex:Direction
	latitude	datex:PointCoordinates(latitude)
	longitude	datex:PointCoordinates(longitude)
	country	datex:CountryEnum
	frequency	datex:measurementSpecificCharacteristics(period)
	sensorType	*N.A.*
	source	*N.A.*
TRAFFIC EVENT	eventId	datex:SituationRecord(id)
	road	*N.A.*
	section	*N.A.*
	type	datex:SituationRecord(type)
	bearing	datex:DirectionBearingValue(directionBearing)
	latitude	datex:Location(locationForDisplay(latitude))
	longitude	datex:Location(locationForDisplay(longitude))
	country	datex:CountryEnum
	startDateTime	datex:Validity(validityTimeSpecification(validityTimeSpecification(overallStartTime)))
	endDateTime	datex:Validity(validityTimeSpecification(validityTimeSpecification(overallEndTime)))
	description	*N.A.*
	source	datex:SituationRecord(source)
	version	datex:SituationRecord(version)
	severity	datex:SeverityEnum

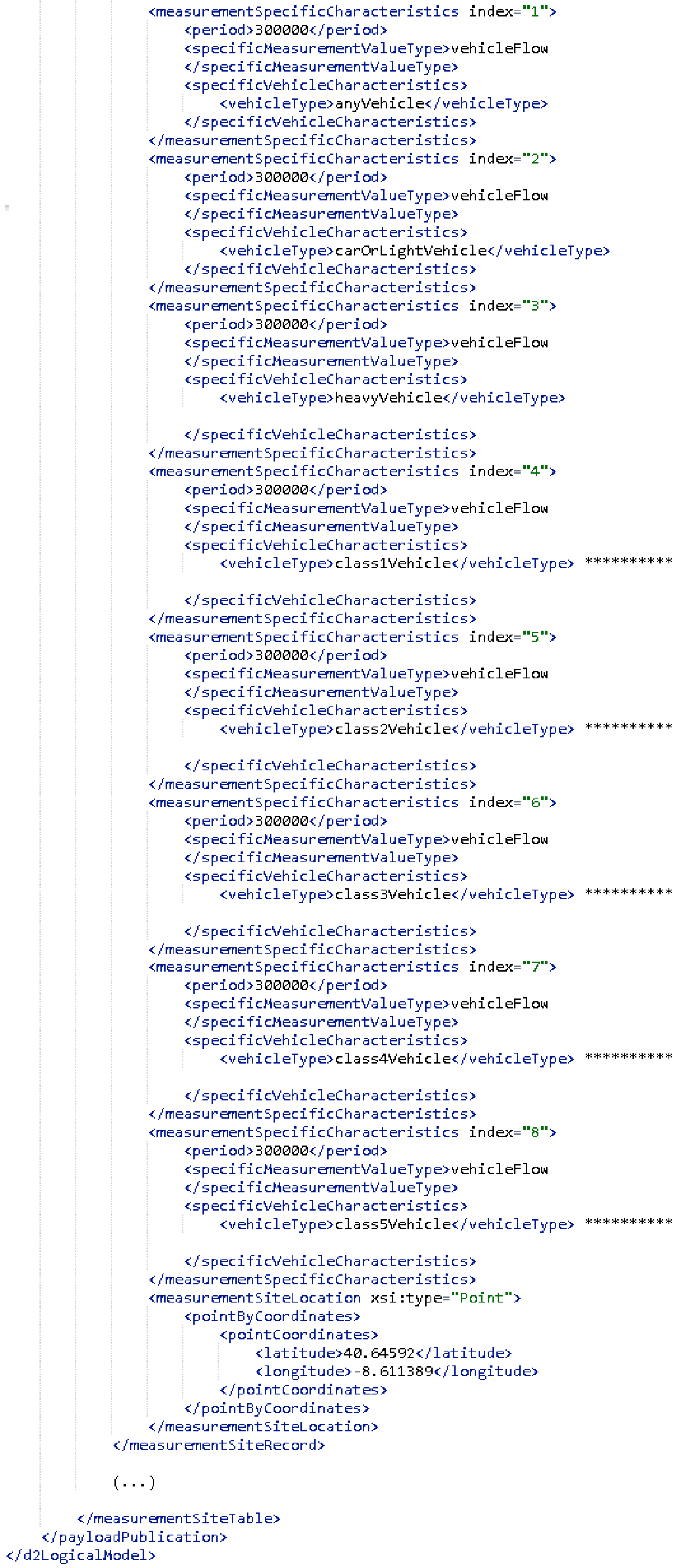
A.19. Traffic Sensor Metadata Harmonized Schema & Corresponding DATEX II Model Example for a Specific Toll Sensor (Portugal)



```
{
    "_id" : ObjectId("56ab5c569f6ed594781cc081"),
    "concession_name" : "Costa Prata",
    "road_name" : "A25",
    "road_type" : "highway",
    "sensor_type" : "toll",
    "km_point" : 12.93,
    "sensor_id_holder" : “2509”,
    "section" : "Aveiro",
    "state" : "active",
    "concession_holder" : "Ascendi",
    "bearing" : "eastbound",
    "country" : "pt",
    "location" : {
        "type" : "Point",
        "coordinates" : [
            -8.611389,
            40.64592
        ]
    }
}
```

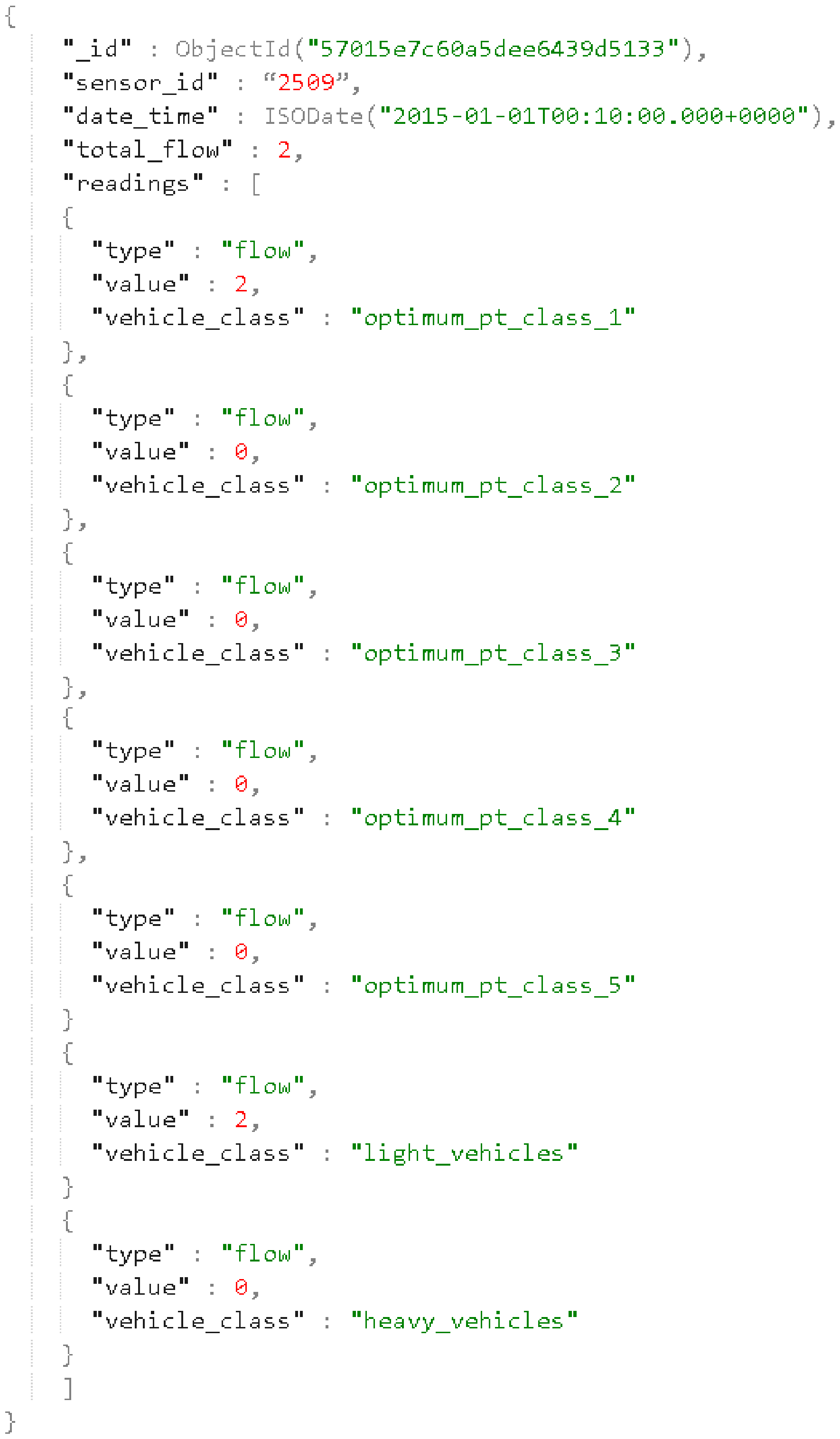


```xml
<?xml version="1.0" encoding="utf-8"?>
<d2LogicalModel
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns="http://datex2.eu/schema/1_0/1_0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" modelBaseVersion="1.0"
    xsi:schemaLocation="http://datex2.eu/schema/1_0/1_0">
    <exchange
        xmlns="http://datex2.eu/schema/1_0/1_0">
        <supplierIdentification>
            <country>pt</country>
            <nationalIdentifier>IP</nationalIdentifier>
        </supplierIdentification>
    </exchange>
    <payloadPublication
        xmlns="http://datex2.eu/schema/1_0/1_0"
        xsi:type="MeasurementSiteTablePublication" lang="pt">
        <publicationTime>2016-10-01T00:00:00+00:00</publicationTime>
        <publicationCreator>
            <country>pt</country>
            <nationalIdentifier>IP</nationalIdentifier>
        </publicationCreator>
        <headerInformation>
            <confidentiality>internalUse</confidentiality>
            <informationStatus>real</informationStatus>
        </headerInformation>
        <measurementSiteTable id="PT_IP_Concessions_2016_1">
            <measurementSiteTableReference>PT_IP_Concessions_Measurementspoints
            </measurementSiteTableReference>
            <measurementSiteTableVersion>1</measurementSiteTableVersion>
            <measurementSiteRecord id="PT_ASC_2509">
                <measurementEquipmentReference>2509
                </measurementEquipmentReference>
                <measurementSiteIdentification>56ab5c569f6ed594781cc081
                </measurementSiteIdentification>
                <measurementSiteName>
                    <value>Aveiro</value>
                </measurementSiteName>
                <measurementEquipmentTypeUsed>
                    <value lang="en">toll</value>
                </measurementEquipmentTypeUsed>
                <measurementSide>northbound</measurementSide>
```

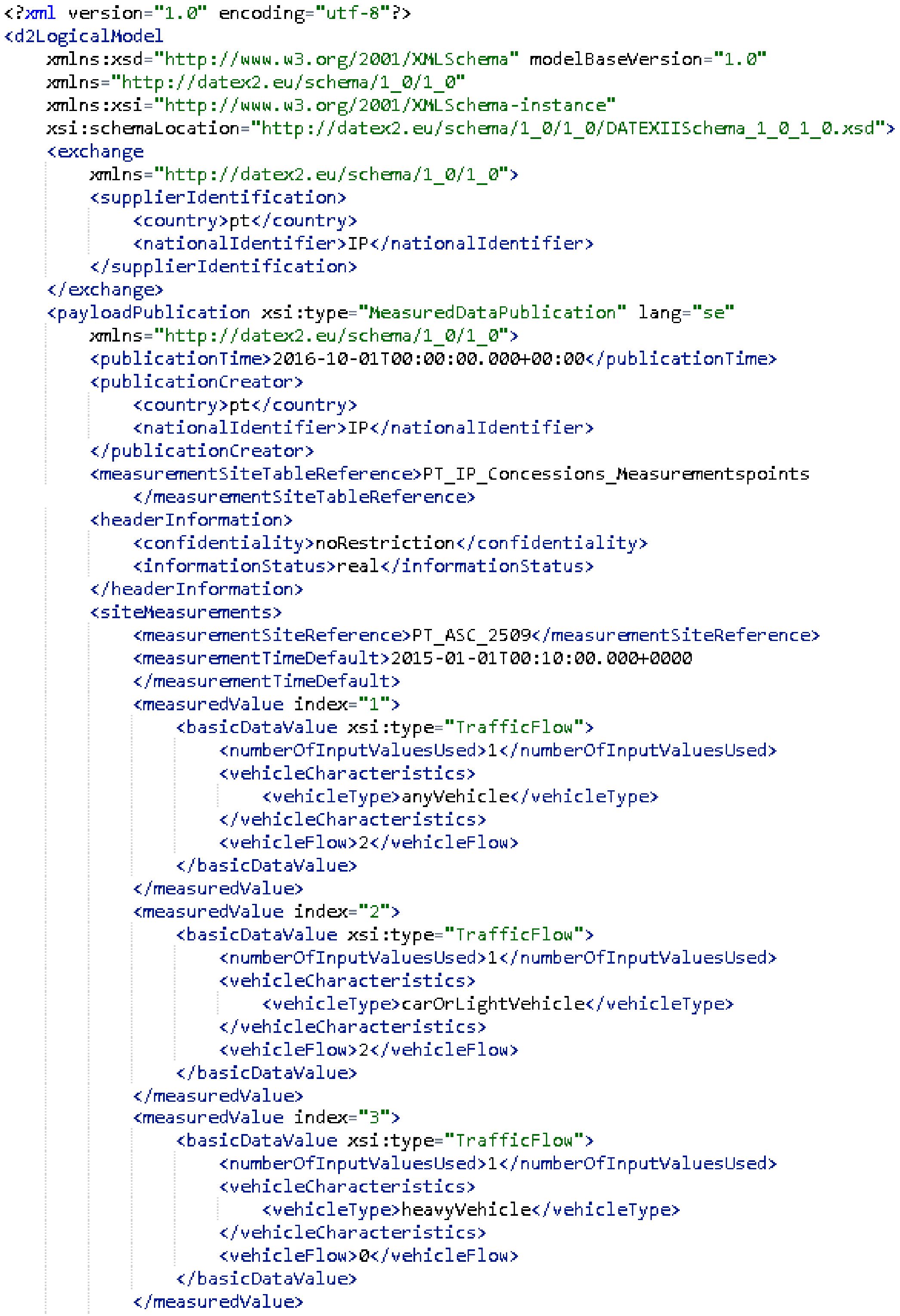


```
                <measurementSpecificCharacteristics index="1">
                    <period>300000</period>
                    <specificMeasurementValueType>vehicleFlow
                    </specificMeasurementValueType>
                    <specificVehicleCharacteristics>
                        <vehicleType>anyVehicle</vehicleType>
                    </specificVehicleCharacteristics>
                </measurementSpecificCharacteristics>
                <measurementSpecificCharacteristics index="2">
                    <period>300000</period>
                    <specificMeasurementValueType>vehicleFlow
                    </specificMeasurementValueType>
                    <specificVehicleCharacteristics>
                        <vehicleType>carOrLightVehicle</vehicleType>
                    </specificVehicleCharacteristics>
                </measurementSpecificCharacteristics>
                <measurementSpecificCharacteristics index="3">
                    <period>300000</period>
                    <specificMeasurementValueType>vehicleFlow
                    </specificMeasurementValueType>
                    <specificVehicleCharacteristics>
                        <vehicleType>heavyVehicle</vehicleType>

                    </specificVehicleCharacteristics>
                </measurementSpecificCharacteristics>
                <measurementSpecificCharacteristics index="4">
                    <period>300000</period>
                    <specificMeasurementValueType>vehicleFlow
                    </specificMeasurementValueType>
                    <specificVehicleCharacteristics>
                        <vehicleType>class1Vehicle</vehicleType> **********

                    </specificVehicleCharacteristics>
                </measurementSpecificCharacteristics>
                <measurementSpecificCharacteristics index="5">
                    <period>300000</period>
                    <specificMeasurementValueType>vehicleFlow
                    </specificMeasurementValueType>
                    <specificVehicleCharacteristics>
                        <vehicleType>class2Vehicle</vehicleType> **********

                    </specificVehicleCharacteristics>
                </measurementSpecificCharacteristics>
                <measurementSpecificCharacteristics index="6">
                    <period>300000</period>
                    <specificMeasurementValueType>vehicleFlow
                    </specificMeasurementValueType>
                    <specificVehicleCharacteristics>
                        <vehicleType>class3Vehicle</vehicleType> **********

                    </specificVehicleCharacteristics>
                </measurementSpecificCharacteristics>
                <measurementSpecificCharacteristics index="7">
                    <period>300000</period>
                    <specificMeasurementValueType>vehicleFlow
                    </specificMeasurementValueType>
                    <specificVehicleCharacteristics>
                        <vehicleType>class4Vehicle</vehicleType> **********

                    </specificVehicleCharacteristics>
                </measurementSpecificCharacteristics>
                <measurementSpecificCharacteristics index="8">
                    <period>300000</period>
                    <specificMeasurementValueType>vehicleFlow
                    </specificMeasurementValueType>
                    <specificVehicleCharacteristics>
                        <vehicleType>class5Vehicle</vehicleType> **********

                    </specificVehicleCharacteristics>
                </measurementSpecificCharacteristics>
                <measurementSiteLocation xsi:type="Point">
                    <pointByCoordinates>
                        <pointCoordinates>
                            <latitude>40.64592</latitude>
                            <longitude>-8.611389</longitude>
                        </pointCoordinates>
                    </pointByCoordinates>
                </measurementSiteLocation>
            </measurementSiteRecord>

            (...)

        </measurementSiteTable>
    </payloadPublication>
</d2LogicalModel>
```

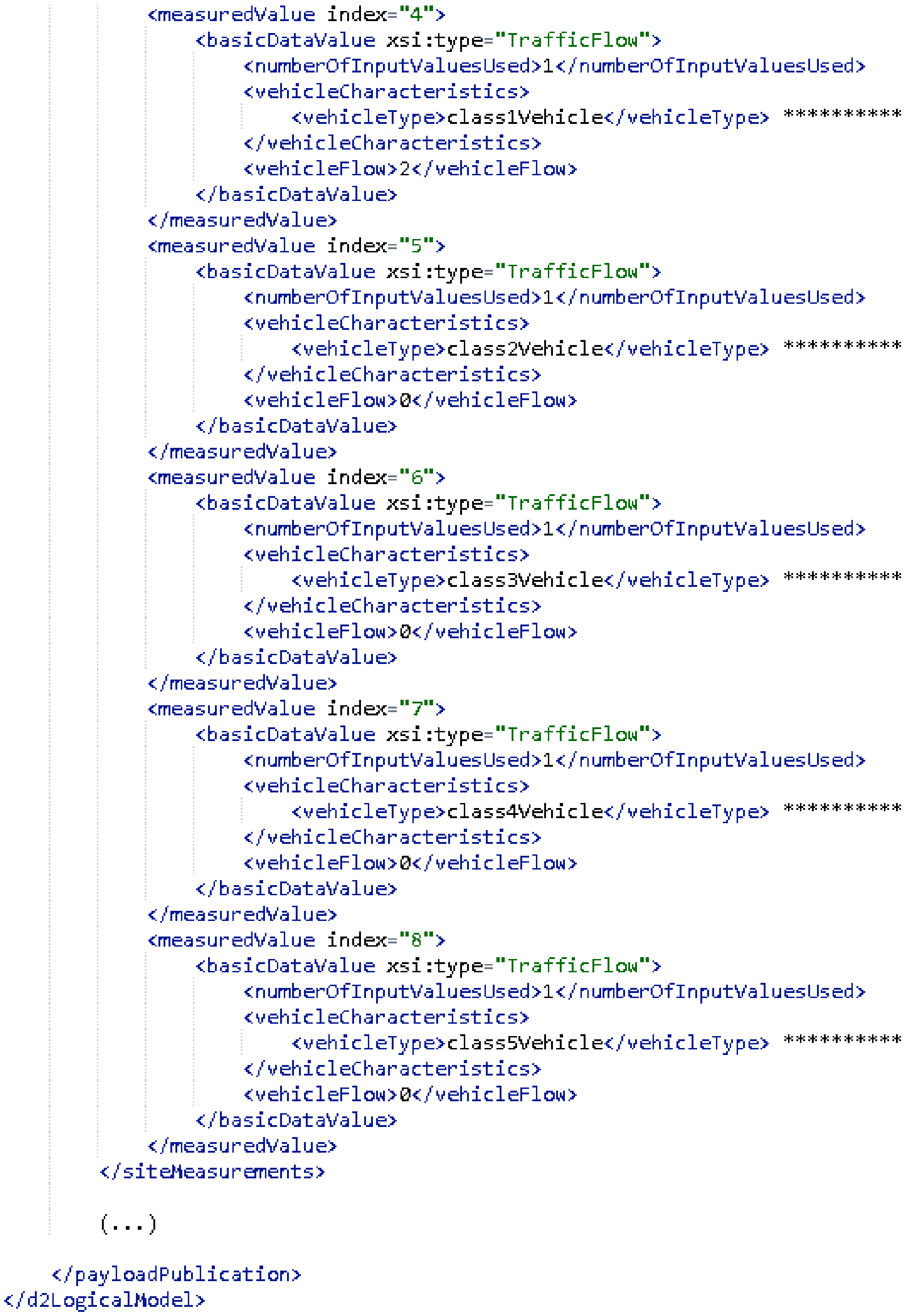
A.20. Traffic Sensor Reading Harmonized Schema & Corresponding DATEX II Model Example for a Specific Toll Sensor (Portugal)



```
{
    "_id" : ObjectId("57015e7c60a5dee6439d5133"),
    "sensor_id" : “2509”,
    "date_time" : ISODate("2015-01-01T00:10:00.000+0000"),
    "total_flow" : 2,
    "readings" : [
    {
      "type" : "flow",
      "value" : 2,
      "vehicle_class" : "optimum_pt_class_1"
    },
    {
      "type" : "flow",
      "value" : 0,
      "vehicle_class" : "optimum_pt_class_2"
    },
    {
      "type" : "flow",
      "value" : 0,
      "vehicle_class" : "optimum_pt_class_3"
    },
    {
      "type" : "flow",
      "value" : 0,
      "vehicle_class" : "optimum_pt_class_4"
    },
    {
      "type" : "flow",
      "value" : 0,
      "vehicle_class" : "optimum_pt_class_5"
    }
    {
      "type" : "flow",
      "value" : 2,
      "vehicle_class" : "light_vehicles"
    }
    {
      "type" : "flow",
      "value" : 0,
      "vehicle_class" : "heavy_vehicles"
    }
    ]
}
```

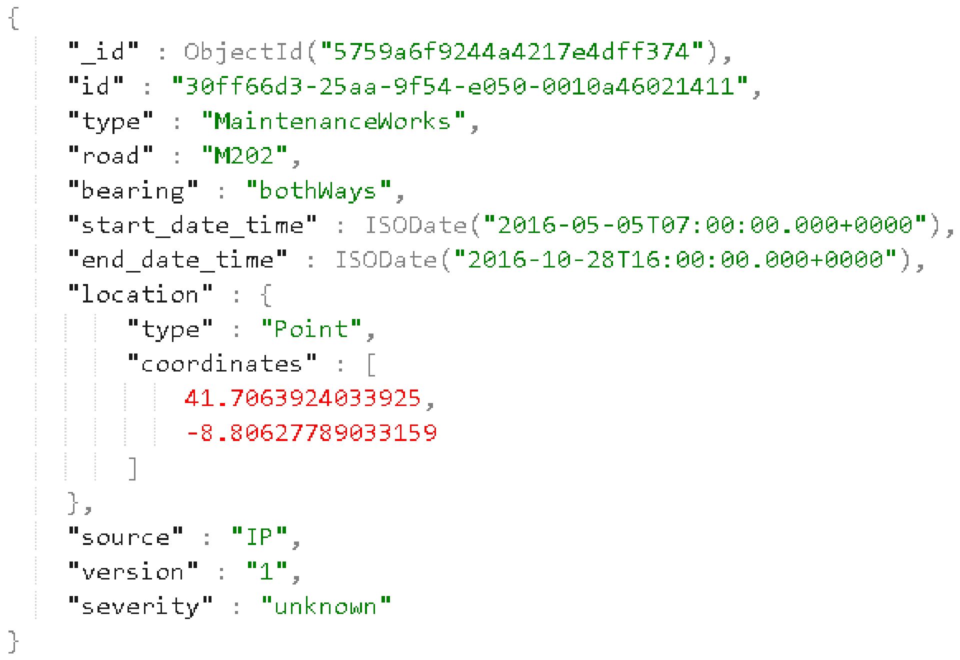


```
<?xml version="1.0" encoding="utf-8"?>
<d2LogicalModel
    xmlns:xsd="http://www.w3.org/2001/XMLSchema" modelBaseVersion="1.0"
    xmlns="http://datex2.eu/schema/1_0/1_0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://datex2.eu/schema/1_0/1_0/DATEXIISchema_1_0_1_0.xsd">
    <exchange
        xmlns="http://datex2.eu/schema/1_0/1_0">
        <supplierIdentification>
            <country>pt</country>
            <nationalIdentifier>IP</nationalIdentifier>
        </supplierIdentification>
    </exchange>
    <payloadPublication xsi:type="MeasuredDataPublication" lang="se"
        xmlns="http://datex2.eu/schema/1_0/1_0">
        <publicationTime>2016-10-01T00:00:00.000+00:00</publicationTime>
        <publicationCreator>
            <country>pt</country>
            <nationalIdentifier>IP</nationalIdentifier>
        </publicationCreator>
        <measurementSiteTableReference>PT_IP_Concessions_Measurementspoints
            </measurementSiteTableReference>
        <headerInformation>
            <confidentiality>noRestriction</confidentiality>
            <informationStatus>real</informationStatus>
        </headerInformation>
        <siteMeasurements>
            <measurementSiteReference>PT_ASC_2509</measurementSiteReference>
            <measurementTimeDefault>2015-01-01T00:10:00.000+0000
            </measurementTimeDefault>
            <measuredValue index="1">
                <basicDataValue xsi:type="TrafficFlow">
                    <numberOfInputValuesUsed>1</numberOfInputValuesUsed>
                    <vehicleCharacteristics>
                        <vehicleType>anyVehicle</vehicleType>
                    </vehicleCharacteristics>
                    <vehicleFlow>2</vehicleFlow>
                </basicDataValue>
            </measuredValue>
            <measuredValue index="2">
                <basicDataValue xsi:type="TrafficFlow">
                    <numberOfInputValuesUsed>1</numberOfInputValuesUsed>
                    <vehicleCharacteristics>
                        <vehicleType>carOrLightVehicle</vehicleType>
                    </vehicleCharacteristics>
                    <vehicleFlow>2</vehicleFlow>
                </basicDataValue>
            </measuredValue>
            <measuredValue index="3">
                <basicDataValue xsi:type="TrafficFlow">
                    <numberOfInputValuesUsed>1</numberOfInputValuesUsed>
                    <vehicleCharacteristics>
                        <vehicleType>heavyVehicle</vehicleType>
                    </vehicleCharacteristics>
                    <vehicleFlow>0</vehicleFlow>
                </basicDataValue>
            </measuredValue>
```

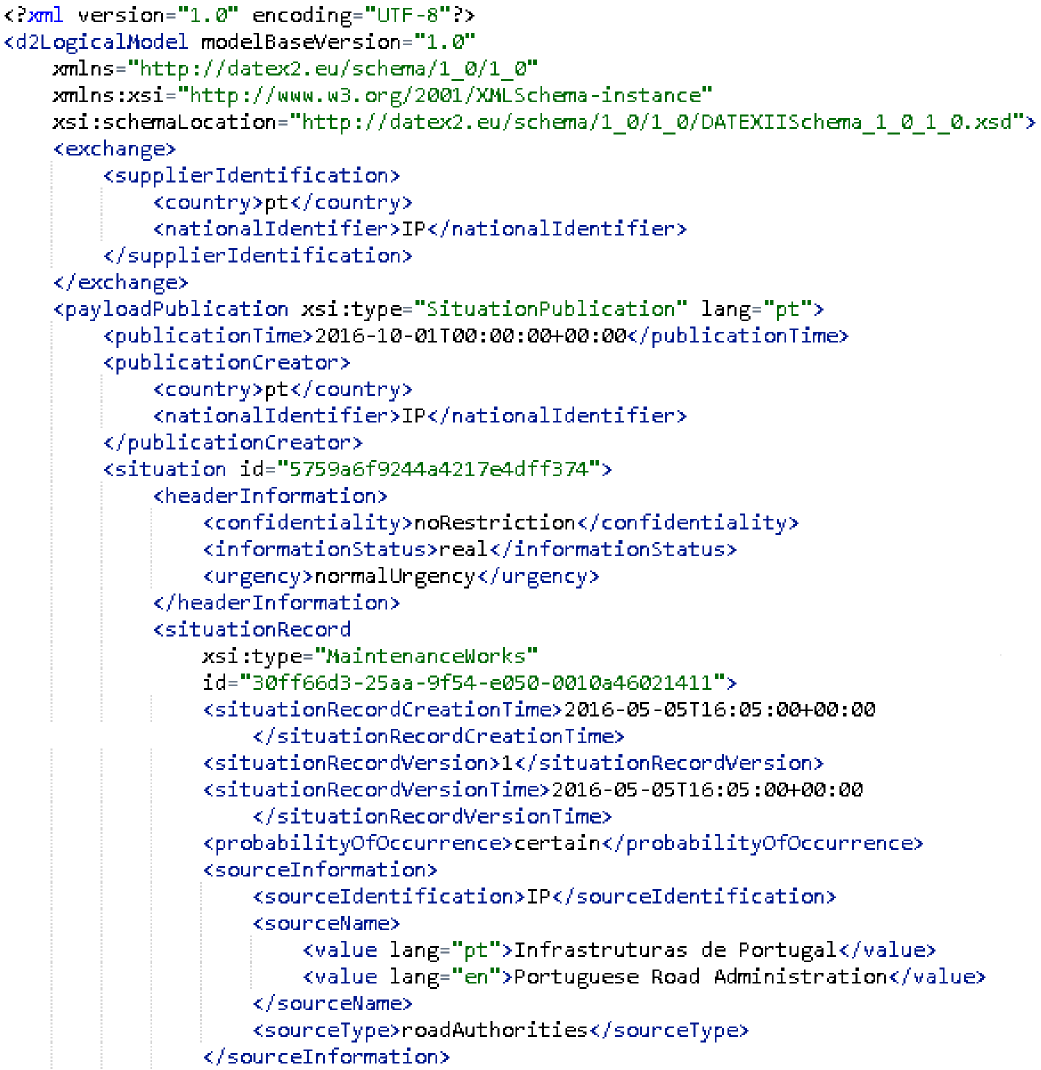


```
            <measuredValue index="4">
                <basicDataValue xsi:type="TrafficFlow">
                    <numberOfInputValuesUsed>1</numberOfInputValuesUsed>
                    <vehicleCharacteristics>
                        <vehicleType>class1Vehicle</vehicleType> **********
                    </vehicleCharacteristics>
                    <vehicleFlow>2</vehicleFlow>
                </basicDataValue>
            </measuredValue>
            <measuredValue index="5">
                <basicDataValue xsi:type="TrafficFlow">
                    <numberOfInputValuesUsed>1</numberOfInputValuesUsed>
                    <vehicleCharacteristics>
                        <vehicleType>class2Vehicle</vehicleType> **********
                    </vehicleCharacteristics>
                    <vehicleFlow>0</vehicleFlow>
                </basicDataValue>
            </measuredValue>
            <measuredValue index="6">
                <basicDataValue xsi:type="TrafficFlow">
                    <numberOfInputValuesUsed>1</numberOfInputValuesUsed>
                    <vehicleCharacteristics>
                        <vehicleType>class3Vehicle</vehicleType> **********
                    </vehicleCharacteristics>
                    <vehicleFlow>0</vehicleFlow>
                </basicDataValue>
            </measuredValue>
            <measuredValue index="7">
                <basicDataValue xsi:type="TrafficFlow">
                    <numberOfInputValuesUsed>1</numberOfInputValuesUsed>
                    <vehicleCharacteristics>
                        <vehicleType>class4Vehicle</vehicleType> **********
                    </vehicleCharacteristics>
                    <vehicleFlow>0</vehicleFlow>
                </basicDataValue>
            </measuredValue>
            <measuredValue index="8">
                <basicDataValue xsi:type="TrafficFlow">
                    <numberOfInputValuesUsed>1</numberOfInputValuesUsed>
                    <vehicleCharacteristics>
                        <vehicleType>class5Vehicle</vehicleType> **********
                    </vehicleCharacteristics>
                    <vehicleFlow>0</vehicleFlow>
                </basicDataValue>
            </measuredValue>
        </siteMeasurements>

        (...)

    </payloadPublication>
</d2LogicalModel>
```

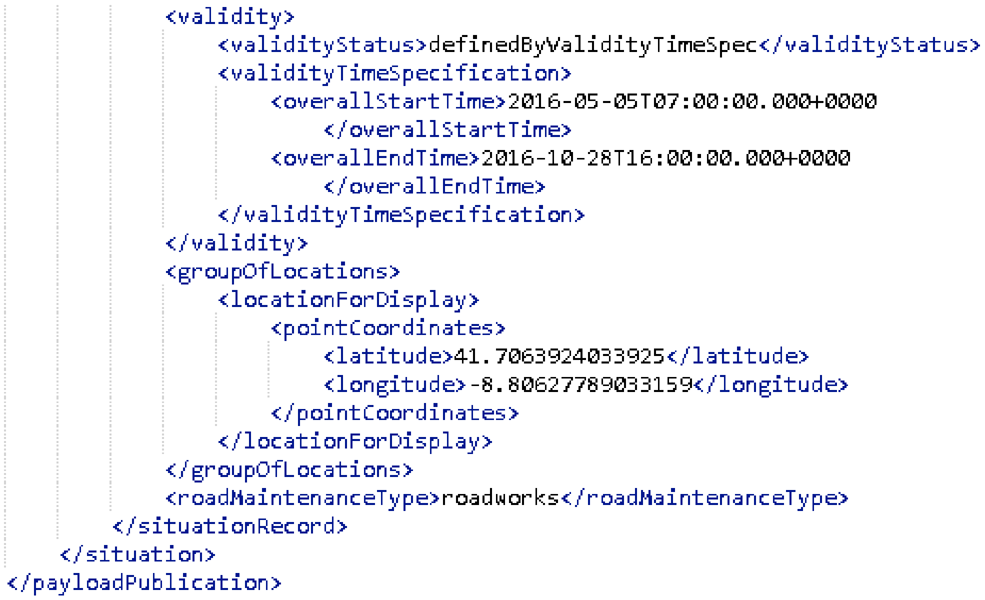
A.21. Traffic Event Harmonized Schema & Corresponding DATEX II Model Example for a Specific Event (Portugal)



```
{
    "_id" : ObjectId("5759a6f9244a4217e4dff374"),
    "id" : "30ff66d3-25aa-9f54-e050-0010a46021411",
    "type" : "MaintenanceWorks",
    "road" : "M202",
    "bearing" : "bothWays",
    "start_date_time" : ISODate("2016-05-05T07:00:00.000+0000"),
    "end_date_time" : ISODate("2016-10-28T16:00:00.000+0000"),
    "location" : {
        "type" : "Point",
        "coordinates" : [
            41.7063924033925,
            -8.80627789033159
        ]
    },
    "source" : "IP",
    "version" : "1",
    "severity" : "unknown"
}
```

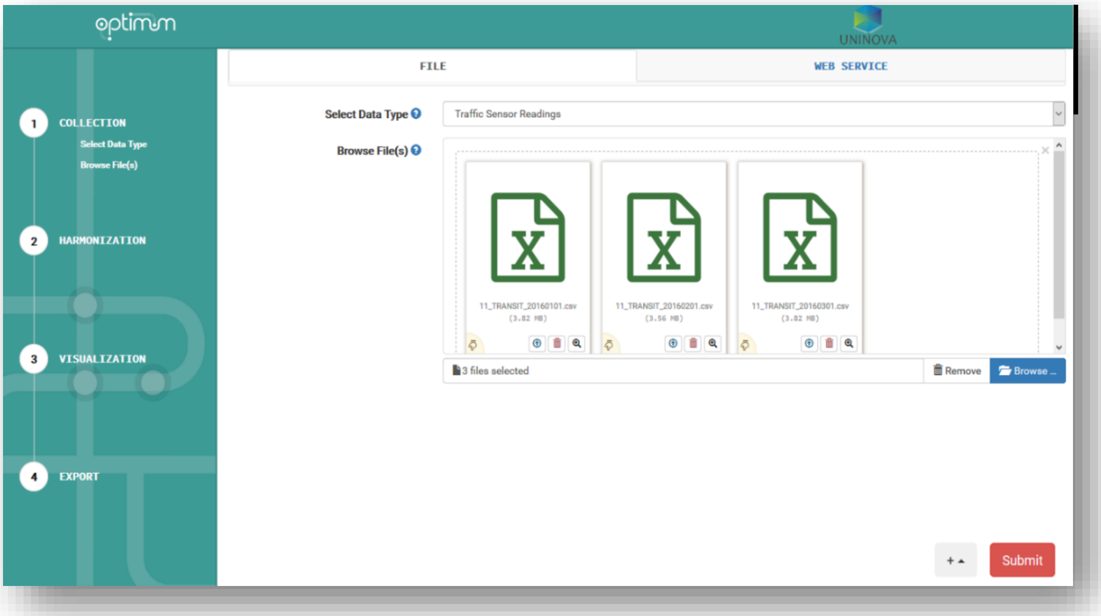


```
<?xml version="1.0" encoding="UTF-8"?>
<d2LogicalModel modelBaseVersion="1.0"
    xmlns="http://datex2.eu/schema/1_0/1_0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://datex2.eu/schema/1_0/1_0/DATEXIISchema_1_0_1_0.xsd">
    <exchange>
        <supplierIdentification>
            <country>pt</country>
            <nationalIdentifier>IP</nationalIdentifier>
        </supplierIdentification>
    </exchange>
    <payloadPublication xsi:type="SituationPublication" lang="pt">
        <publicationTime>2016-10-01T00:00:00+00:00</publicationTime>
        <publicationCreator>
            <country>pt</country>
            <nationalIdentifier>IP</nationalIdentifier>
        </publicationCreator>
        <situation id="5759a6f9244a4217e4dff374">
            <headerInformation>
                <confidentiality>noRestriction</confidentiality>
                <informationStatus>real</informationStatus>
                <urgency>normalUrgency</urgency>
            </headerInformation>
            <situationRecord
                xsi:type="MaintenanceWorks"
                id="30ff66d3-25aa-9f54-e050-0010a46021411">
                <situationRecordCreationTime>2016-05-05T16:05:00+00:00
                    </situationRecordCreationTime>
                <situationRecordVersion>1</situationRecordVersion>
                <situationRecordVersionTime>2016-05-05T16:05:00+00:00
                    </situationRecordVersionTime>
                <probabilityOfOccurrence>certain</probabilityOfOccurrence>
                <sourceInformation>
                    <sourceIdentification>IP</sourceIdentification>
                    <sourceName>
                        <value lang="pt">Infrastruturas de Portugal</value>
                        <value lang="en">Portuguese Road Administration</value>
                    </sourceName>
                    <sourceType>roadAuthorities</sourceType>
                </sourceInformation>
```



```
                <validity>
                    <validityStatus>definedByValidityTimeSpec</validityStatus>
                    <validityTimeSpecification>
                        <overallStartTime>2016-05-05T07:00:00.000+0000
                            </overallStartTime>
                        <overallEndTime>2016-10-28T16:00:00.000+0000
                            </overallEndTime>
                    </validityTimeSpecification>
                </validity>
                <groupOfLocations>
                    <locationForDisplay>
                        <pointCoordinates>
                            <latitude>41.7063924033925</latitude>
                            <longitude>-8.80627789033159</longitude>
                        </pointCoordinates>
                    </locationForDisplay>
                </groupOfLocations>
                <roadMaintenanceType>roadworks</roadMaintenanceType>
            </situationRecord>
        </situation>
    </payloadPublication>
```

A.22. Big Data Harmonization Web Application: Views



Select Data Type: Traffic Sensor Readings

Browse File(s):

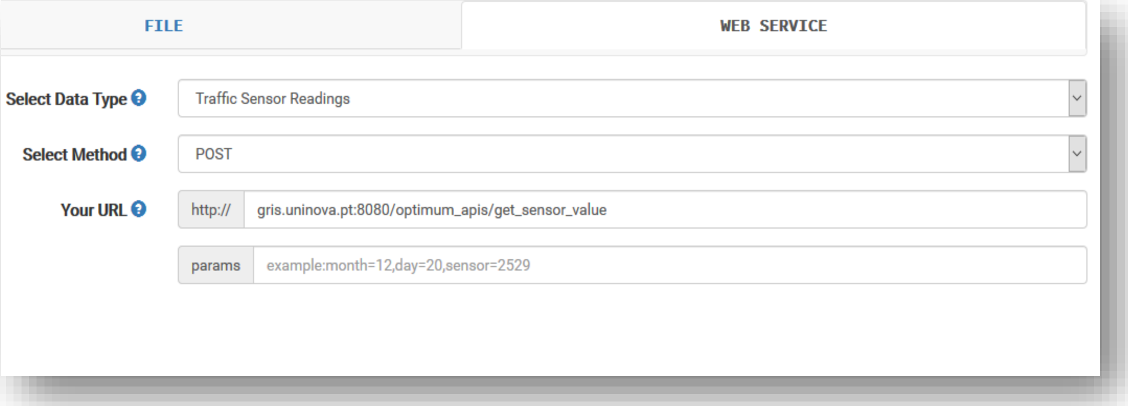
11_TRANSIT_20160101.csv (3.82 MB)

11_TRANSIT_20160201.csv (3.56 MB)

11_TRANSIT_20160301.csv (3.82 MB)

3 files selected

View 1 — Data collection from files



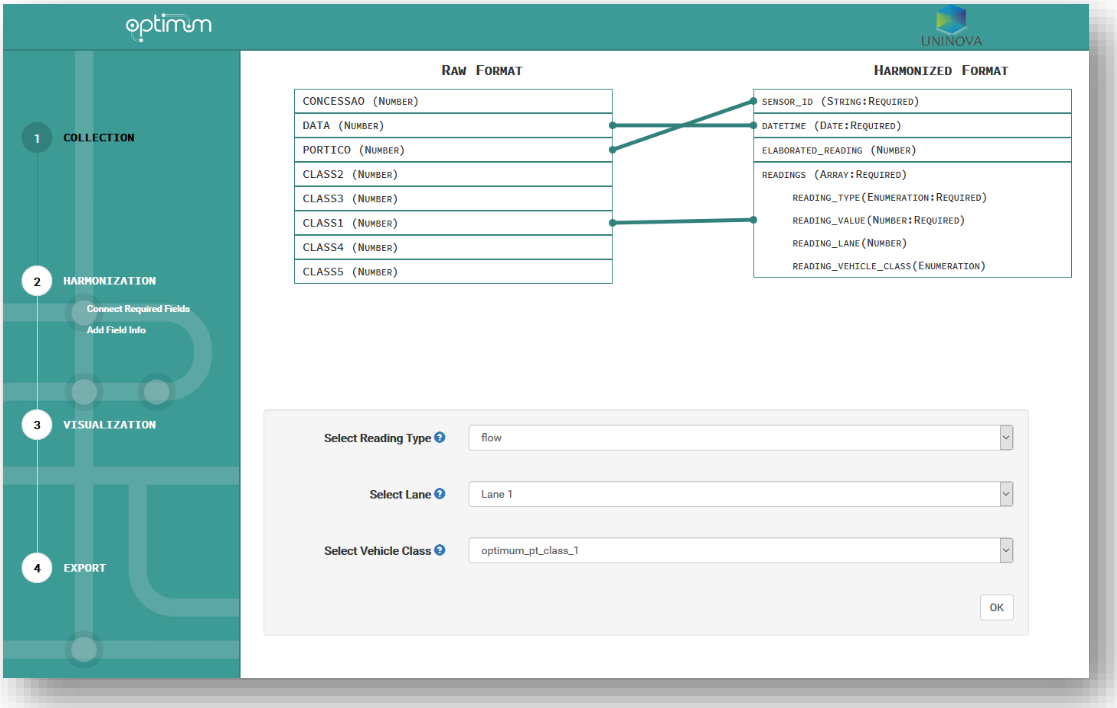
Select Data Type: Traffic Sensor Readings

Select Method: POST

Your URL: http:// gris.uninova.pt:8080/optimum_apis/get_sensor_value

params: example:month=12,day=20,sensor=2529

View 2 — Data collection from Web services and pub-sub mechanisms



RAW FORMAT

- CONCESSAO (NUMBER)
- DATA (NUMBER)
- PORTICO (NUMBER)
- CLASS2 (NUMBER)
- CLASS3 (NUMBER)
- CLASS1 (NUMBER)
- CLASS4 (NUMBER)
- CLASS5 (NUMBER)

HARMONIZED FORMAT

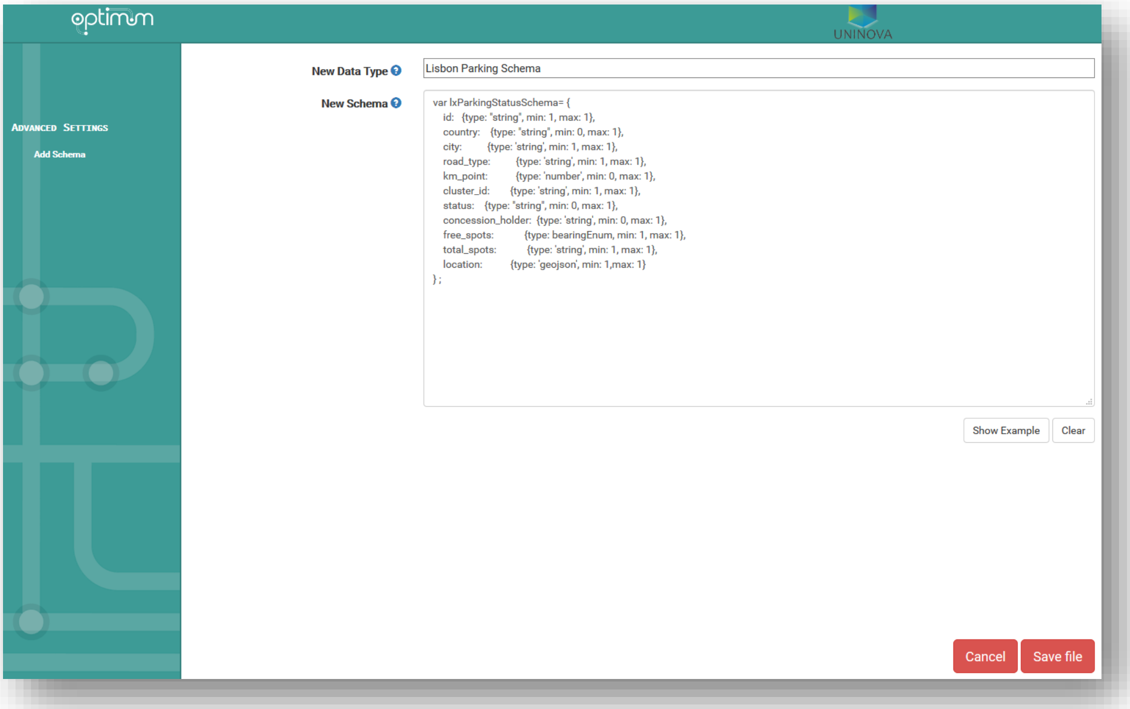
- SENSOR_ID (STRING:REQUIRED)
- DATETIME (DATE:REQUIRED)
- ELABORATED_READING (NUMBER)
- READINGS (ARRAY:REQUIRED)
  - READING_TYPE(ENUMERATION:REQUIRED)
  - READING_VALUE(NUMBER:REQUIRED)
  - READING_LANE(NUMBER)
  - READING_VEHICLE_CLASS(ENUMERATION)

Select Reading Type: flow

Select Lane: Lane 1

Select Vehicle Class: optimum_pt_class_1

View 3 — Mapping raw data parameters to harmonized schemas

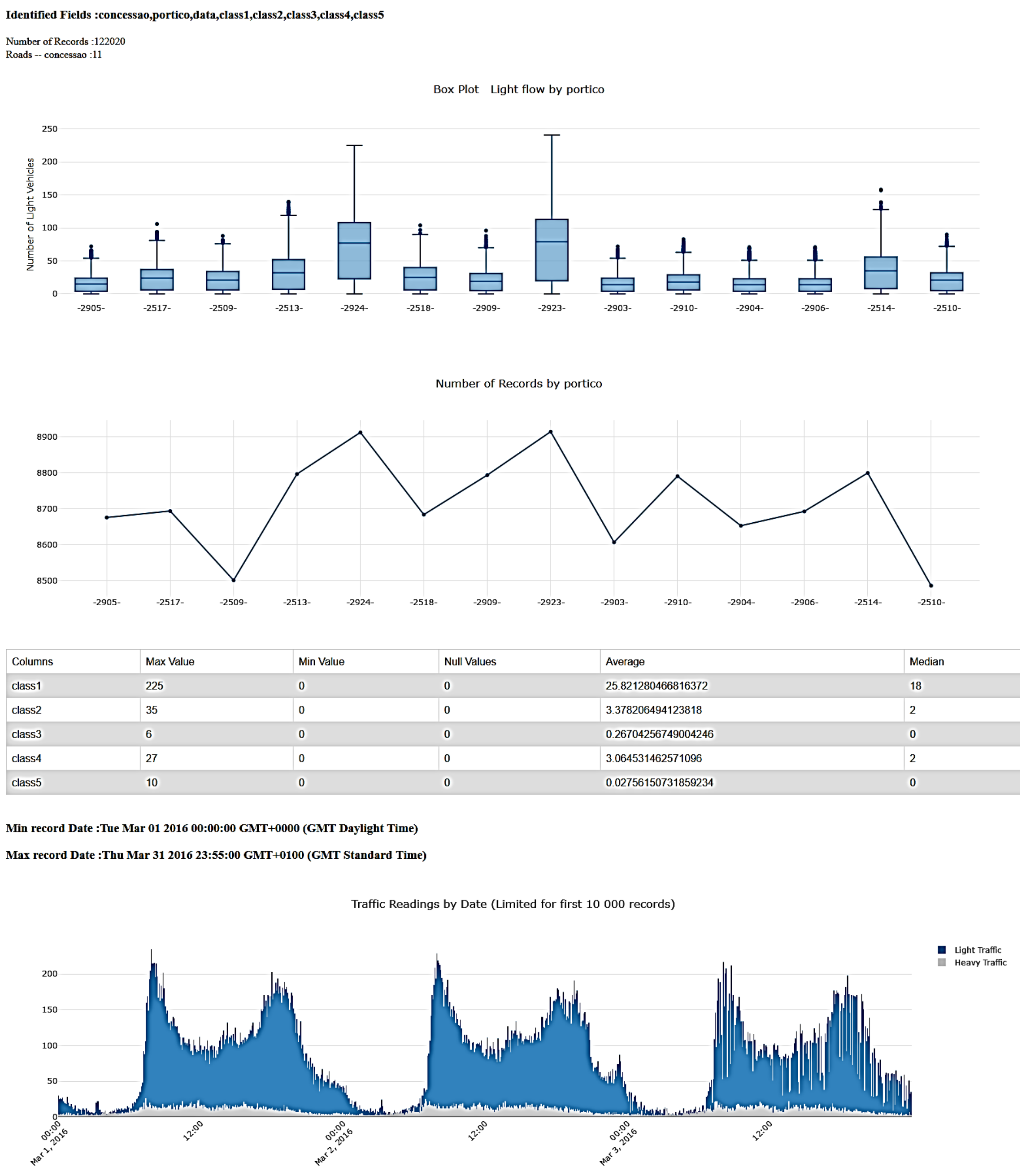


New Data Type: Lisbon Parking Schema

New Schema:

```
var lxParkingStatusSchema= {
   id:   {type: "string", min: 1, max: 1},
   country:    {type: "string", min: 0, max: 1},
   city:         {type: 'string', min: 1, max: 1},
   road_type:         {type: 'string', min: 1, max: 1},
   km_point:          {type: 'number', min: 0, max: 1},
   cluster_id:       {type: 'string', min: 1, max: 1},
   status:    {type: "string", min: 0, max: 1},
   concession_holder:  {type: 'string', min: 0, max: 1},
   free_spots:           {type: bearingEnum, min: 1, max: 1},
   total_spots:           {type: 'string', min: 1, max: 1},
   location:          {type: 'geojson', min: 1,max: 1}
};
```

View 4 — Adding new custom harmonized schemas



Identified Fields :concessao,portico,data,class1,class2,class3,class4,class5

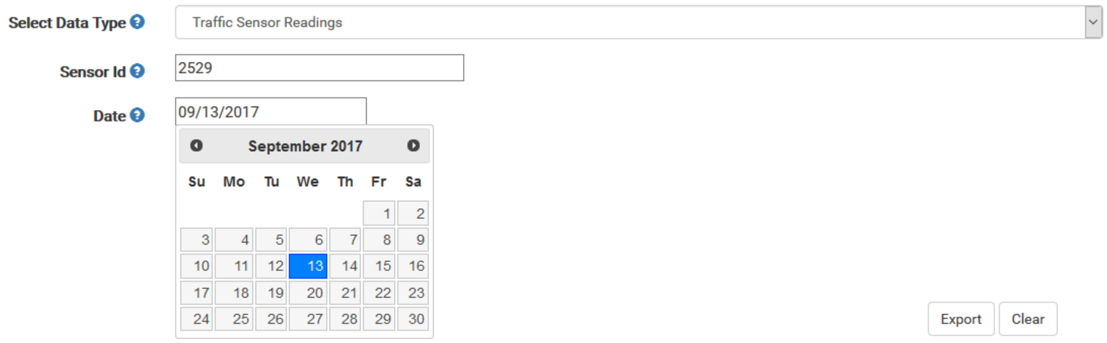
Number of Records :122020
Roads -- concessao :11

Columns	Max Value	Min Value	Null Values	Average	Median
class1	225	0	0	25.821280466816372	18
class2	35	0	0	3.378206494123818	2
class3	6	0	0	0.26704256749004246	0
class4	27	0	0	3.064531462571096	2
class5	10	0	0	0.02756150731859234	0

Min record Date :Tue Mar 01 2016 00:00:00 GMT+0000 (GMT Daylight Time)

Max record Date :Thu Mar 31 2016 23:55:00 GMT+0100 (GMT Standard Time)

View 5 — Insight visualization of harmonized data sets



Select Data Type: Traffic Sensor Readings

Sensor Id: 2529

Date: 09/13/2017

September 2017

Su	Mo	Tu	We	Th	Fr	Sa
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

View 6 — Selecting harmonized data to export

UNIVERSIDADE NOVA DE LISBOA • OMNIS CIVITAS CONTRA SE DIVISA NON STABIT •

A DATA-DRIVEN METHODOLOGY TOWARDS MOBILITY- AND TRAFFIC-RELATED BIG SPATIOTEMPORAL DATA FRAMEWORKS

PAULO ALVES FIGUEIRAS

2021