Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria Informatica, Gestionale e dell'Automazione

---

Intelligent Decision Support Systems in Public Transport

Ph.D. Dissertation of:
Filippo Benvenuti

Advisor:
Prof. Claudia Diamantini

Curriculum Supervisor:
Prof. Francesco Piazza

XVI edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Curriculum in Ingegneria Informatica, Gestionale e dell'Automazione

---

Intelligent Decision Support Systems in Public Transport

Ph.D. Dissertation of:
Filippo Benvenuti

Advisor:
Prof. Claudia Diamantini

Curriculum Supervisor:
Prof. Francesco Piazza

XVI edition - new series

Università Politecnica delle Marche
Scuola di Dottorato di Ricerca in Scienze dell'Ingegneria
Facoltà di Ingegneria
Via Brecce Bianche – 60131 Ancona (AN), Italy

To my Family

Acknowledgments

A good work can not be accomplished alone but needs the support and help of many persons. Here, I'd like to thank all the people who have allowed me to achieve this prestigious goal. First of all, I want to thank my supervisor Claudia Diamantini. Her guidance, help and valuable comments made this thesis possible. I also thank Domenico Potena for his helpful suggestions all through my PhD work. It has been a great experience for me to be part of their group in these years. A special thanks also goes to PluService company and to my company supervisor Eddy Belli that provided me all the data and support that I needed to achieve this goal. I would also like to show my sincere gratitude to my colleagues and friends: Emanuele, Marco, Laura, Gilberto, Leonardo, Daniele, Alex, Marina, Mirco and many others for the good moments spent together in these beautiful years. I am indebted to the many undergraduate students that have significantly helped me in testing, evaluating and implementing some components of the platform described herein. I wish to give a very special thank to my parents, brother and friends who have always supported me in these years. Lastly I want to thanks the reason of my life: Cinzia. Without her I would be empty like a starless sky.

Ancona, November 2017

Filippo Benvenuti

Abstract

In the last years, the significance of improving efficiency of urban Public Transport Systems (PTSs) is taking a key role in the development of modern society. In an ideal world, urban PTSs have no particular problems in providing effective and efficient services that improve the Quality of Life (QoL) of people living in big urban areas. An efficient service would also help to solve some additional problems, such as air pollution and traffic congestions that often afflict high density urban areas. The reality, however, is that urban transportation systems in most big cities are far from ideal. Managers of PTSs services are experiencing ever greater difficulties in maintaining high levels of efficiency due to the ever-increasing congestion affecting the major urban centers. In order to face these issues and, at the same time, improve the efficiency of PTSs services, three different proposals are introduced. First, a framework, aimed to ease the design of a monitoring system in the public transport domain, has been realized by adopting the European standard Transmodel as reference model for a generic PTS system. The proposed framework is built around the definition of a knowledge base that includes a conceptualization of the public transportation domain, on the top of which a set of logic-based functionalities are developed. The second improvement proposed, consists in the implementation of a forecasting algorithm in order to predict arrival time at bus stop in urban areas that allows to improve the travelers' perception of the Quality of Service offered. In literature, several type of algorithms have been tested in order to predict arrival time at bus stops. However, in recent years, in addition to models formed by a single algorithm (called *Simple Models*), models formed by the composition of more algorithms (called *Hybrid Models*) have emerged. An overview of *Hybrid Models* has been performed by comparing them with *Simple Models* in a real-world case study from which resulted that the *Hybrid Models* outperform *Simple Models* in every experiment performed. The third improvement proposed regards the study of the impact of an *"unbalanced"* methodology in vision systems able to solve various issues related to traffic management such as parking discovery and occupational turnaround in order to develop a software application to optimize their employment and, in the same time, minimize traffic caused by parking lot search. Unfortunately PluService company, which provided the data to develop the entire work, had no dataset that can be used for the proposed purpose, and, for this reason,

in order to assess the quality of the proposed methodology, it has been tested with building detection problem from LIDAR aerial data. Building Detection from aerial data, in literature, represents a notoriously *"difficult"* application domain to face, cause the imbalance that characterizes the datasets used. The proposed method takes advantage of the Baesyan Vector Quantizer (BVQ) algorithm and, in order to verify its efficiency in strongly *umbalanced* application domains, it has been compared with other well known methodologies, like Adaboost or Metacost, in a real-world case study. The datasets used are formed by four strongly imbalanced classes (building, high vegetation, low vegetation and streets) and the obtained results demonstrated that BVQ outperform the other methodologies applied in the totality of experiments performed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, the significance of providing a urban Public Transport System (PTS) is taking a key role in the technological improvement of our society. As described in [2], in 2016, 1.7 billion people, representing the 23% of the world's population, lived in cities populated by, at least, 1 million inhabitants. By considering these data, in 2030, a projected 27% of people worldwide will be concentrated in big cities with, at least, 1 million inhabitants. A smaller number of people reside, instead, in what are usually called, *megacities*. People who live in megacities are 500 million and represents the 6.8% of the global population in 2016. But, as these cities increase in both size and number, they will become home to a growing share of the population. By 2030, a projected 730 million people will live in cities with at least 10 million inhabitants, representing 8.7% of people globally. The trend described by these numbers is evident: all the above mentioned percentages are set to grow all over the world as time passes and this represents a problem, especially in Europe where the population density index is already high and fixed to 134 people per square-mile (only the Asian continent introduce an higher index with 203 people per square-mile). It is a fact that public transport plays a crucial role in urban development by providing an easiest access for people to education, markets, employment, recreation, health care and other key services. Especially in big cities of the world, enhanced mobility for the poor and vulnerable groups is one of the most important preconditions for augmenting the people's Quality of Life (QoL). The existing reality, however, is that urban transportation systems in most big cities are far from ideal. The most visible and frequently mentioned transport problem of a city of such dimension is its growing traffic congestion, and it is well known that crescent levels of congestion create significant impact on local and national GDP, without considering side effects like growing air pollution or deaths caused by car accidents. It is evident that, under the conditions just exposed, managers of PTSs are encountering growing difficulties in obtaining high quality of services in order to improve the mobility of citizens, while reducing costs and ensuring safety and low environmental impact of performed journeys. Public Transport systems need to be optimized

from the different points of view in order to encourage people in using it while reducing, at the same time, the use of private vehicles like car and motorcycles with all the resulting benefits. In order to evaluate the *"health state"* of a public transport system, it is necessary to provide a tool that allows a quick and efficient analysis of the service provided by different perspectives. To this end, monitoring systems are built to evaluate specific performances related to processes, in order to determine if business objectives (e.g., minimization of delays, pollution reduction, etc.) are met or not. In these contexts, the use of Key Performance Indicators (KPI) represents a well-established way in order to evaluate the performances achieved by specific activities and tasks, by giving an immediate and synthetic view. Most of available management systems for PTSs are designed by referring to standard models, e.g. the The Service Interface for Real Time Information (SIRI) that represents a standard that allow distributed computers to exchange real time information about public transport services and vehicles or the Transmodel Data Model (TDM) that defines a standard data model for a generic PTS. As for these last, typically it is not required for individual systems or specifications to implement the model as a whole. However, they are large and complex models. To give an example, in Transmodel are defined over 370 classes arranging them in 14 core modules divided into 61 submodules, including a huge variety of measurements about various aspects associated with transport services. Hence, selecting the most relevant KPIs for a certain objective, or the procedures that must be followed to compute a certain indicator are all non-trivial tasks even with a small number of objectives, as well as understanding the modules of the management system for PTS that are to be used or where the information needed to calculate a KPI is stored. Existing management solutions for PTSs are capable to provide only little support on how to setup and configure a monitoring system. Indeed, several studies focused only on a part of the development of a PTS, for instance analyzing in detail only a case studies of interest (e.g., sustainability), or a specific scenario (e.g., urban transport).

In addition to improve planning and management of resources, an other way to improve the attractiveness of a PTS is to give, in a clear and timely fashion, information about bus arrival to passengers, leading to: reduced waiting times at bus stops and improved planning for trips that must be performed. A precise travel time prediction is valuable for both travelers and logistic operators, since its aid allows to evade congested route in order to lessen transport outlays and increase upsurge facility excellence. For traffic managers travel time information is a significant index of traffic system operation. Especially, travel-time data is critical for pre-trip and en route information which is highly informative to drivers and travelers. Recent technologies have introduced in vehicles computers equipped with Global Positioning Systems (GPS), that en-

able collection of vast data like, for example, arrival of a transit vehicle, dwell times and bus speed that can be used both to analyse the *status* of a trip and to predict travel times in urban areas.

Another factor that tends to increase traffic congestion in urban areas is represented by the high demand for parking that some areas of cities cannot fully satisfy. In these situations, the Quality of Life of people decrease significantly because the so-called *"parking stress"*, as such parking search requires physical and mental efforts, above all, on the driver. The reduction of this issue as well as improving the Quality of Life of private car drivers, can, at the same time, significantly reduce traffic congestion in urban areas.

From these motivations, the goal of this thesis is to face the above mentioned issues by introducing three different proposals in order to improve the efficiency of a PTS system. The first proposal consists in a framework aimed to ease the design of a monitoring system in the public transport domain. The second proposal, instead, consists in the implementation of a forecasting algorithm to predict arrival time at bus stop in urban transit system at the end of a careful overview of the literature. The third proposal provides the development of a software application used to both optimize parking employment and minimize traffic caused by parking lot search.

The framework is built around the definition of a knowledge base including a conceptualization of the public transportation domain, on the top of which a set of logic-based functionalities are developed. The forecasting models tested, instead, belongs to two different classes of algorithms: *Hybrid Models* that are formed by the composition of more forecasting algorithms and *Simple Models* that are formed, instead, by a single forecasting algorithm. The idea of application used to optimize parking search is based on the study of an *"unbalanced"* methodology that exploits surveillance and vision systems in order to optimize parking occupation and turnaround. Unfortunately PluService company, which provided the data to develop this entire work, had no dataset that can be used for this purpose. For this reason, the *"unbalanced"* approach has been adopted to exploit the automatic building detection problem with LIDAR areal data. The chosen datasets represent a well-known *"umbalanced"* problem where other approaches have been used.

The thesis is organized as follows: Chapter 1 focuses on introducing urban Public Transport System in Europe by tracing its history and evaluating its current limitations and developments. In chapter 2 the developed framework is described in detail; particular focus is given to the Knowledge Base used to conceptualized the TDM and the Reasoning Framework created to build logic-based functionalities. At the end of the chapter, the Framework is used in real world case study. In chapter 3, the forecasting algorithms to predict arrival

time at bus stop in urban transit system are introduced, tested and discussed. In chapter 4, an automatic approach to identify building in urban areas from aerial high resolution image is discussed. In chapter 5, some conclusion and future works are proposed.

This research has been funded by the European Commission, MIUR and PluService company that co-financed this project and provided the used data.

1.1 Evolution of European Urban Public Transport

The concept of *"Urban Public Transport"* began to emerge around the 17th century, when an ever-growing number of persons needed to be able to cover medium distances in order to reach, for example, the workplace or to carry on an increasing number of daily activities. During that period, governors and technicians started to study and put into practice some solutions that could partially solve the question. One of the first solutions developed, before the introduction of motor vehicles, was represented by the *"Omnibus"*, consisting in large horse-drawn carriages that could carry from 12 to 20 people per trip. In a typical scenario of usage, this vehicle was modeled with two wooden benches along the sides of the passenger cabin with several sitting passengers facing each other. The driver sat on a separate bench, typically in an elevated position. The *"Omnibus"* project was originally made by Blaise Pascal in 1662 but only in the following centuries was actually used in big cities and urban areas due to excessive fares and some restrictions that allowed the use of this service only to high society members. The first application of the Omnibus as a Urban Public Transport service was made by Jacques Lafitte in Paris (1819), which allowed up to 50 people to ride across the city in a shared vehicles, avoiding the city's muddy streets. The growing success of this service, enabled the development, in some years, of more structured and organized transport infrastructures. In the first years of 19th century, in the biggest cities of Europe, took place, parallel to the omnibus service, the so-called *"Drawn-Horse TramWay"*. With the term *"Drawn-Horse TramWay"* is meant the terrestrial transport infrastructure, suitable for trams drawn by horses, for both freight and people transport. The first Horse TramWay were developed in England on September 11, 1795 and connected the towns of Crich and Little Eaton. Ten km. long, this line was used for industrial purposes and ran alongside the Derby Canal. The last trip on this line was made in 1905. However, the great technological revolution that was tacking place in those years brought big changes also in the infrastructures and vehicles that were part of the urban public transport system. Horses that represented the only propulsion system of the Omnibus and Tramway services, very soon, were replaced by innovative engines fueled by steam, electricity and carbon fossil fuel. The first prototype of steam bus

was made in England in 1827 but the very first urban transport service based on steam bus was inaugurated on 18 March 1895 to cover the distance subsisting between the cities of Siegen and Netphen and was managed by a company called *"Netphener Omnibusgesellschaft"*. In order to perform the service just described a vehicle named *"Landauer"* was used, which did not look much like a modern bus. This bus was made by hand from 1895 by Karl Benz family-run company; it was composed by eight seats and powered by a 5-horsepower steam engine. Its average speed was about 15 km/h, allowing to cover the distance subsisting between Siegen and Netphen in 1 hour and 20 minutes. Immediately after the appearance of the first vehicles powered by steam engines, bus powered by internal combustion and, even, electric engines also came up. In the London of the early 20th century, in fact, in addition to the steam powered bus, petrol and electric powered models were also tested. Electric powered models had too little autonomy to be competitive while the steam engines had insufficient performance to ensure efficient urban transport service; so the models powered by fossil fuels took over and became the reference model. The bus evolution continued in the sixties in Germany with the unification of the bus types thanks to *"Verband öffentlicher Verkehrsbetriebe"* (VÖV) company, which in conjunction with some transport companies developed prototypes for line buses, which were then refined by several manufacturers . The first *VÖV I* prototype came from 1968 to the production of the *Daimler-Benz* (1969) *MBO 305*, the *Magirus-Deutz 170S11H* (1967), the *Bussing 110V* (1967), the *SL 200 MAN AG* (1971) and *Ikarus 190* (1973). *VÖV II*, as a successor of the *VÖV I* introduced some improvements from a comfort point of view like, as example, a lower floor that allowed an easier climb on the vehicle. The S80 prototypes tested between 1976 and 1978 developed the Auwärter Neoplan N416, the Mercedes-Benz O 405 and the MAN SL 202. With the "VÖV III" a low-floor bus was built, which was the foundation for example for the Neoplan N4014NF, for the Mercedes-Benz O 405 N, for the MAN NL 202 and the IVECO CityClass. Parallel to bus vehicles, TramWays have suffered the same transformation. From horse-drawn tramways, tramway powered by other sources of energy were built. The first trams were built on a chassis resting on two short-stroke axles with pretty elementary cross-sectional suspension. The short step was a necessity determined by the narrow curves of the tramway lines. Even the trailers were two-stroke shorts. The first structural evolution took place with the advent of trolley trams, almost always on two axes each. In the post-great-war period, the two-element construction was built on three trolleys, a solution that, in addition to being cheaper (saving a trolley), also allowed an easier way to board passengers. The success of the ATAG in Rome was in 1940, which commissioned the *Stanga* workshops for the construction of a similar articulated car like STEFER. The prototype was handed over to war

in 1941 and was registered 7001. However, this prototype was destroyed during the bombings that struck Rome in 1943. During the second world-war, ATAC ordered a subsequent lot of 50 cars (odd series 7001-7099) at which were added after the closure of the STEFER tramway lines in 1980, another 8 cars (501-508) refurbished in Viberti workshops (odd series 7101-7115). Many European tramway companies adopted similar solutions.

After the Second World War, for the first time, large-scale vehicles were introduced in the Hamburg tramway for a rapid flow and outflow of passengers. With the evolution of articulated trams, the use of trailers has been greatly reduced.

The seventies represent an important turning point in constructive philosophies. With new tram projects new trams are also emerging that do not resemble any of the previous solutions: while traditional trams become common, new solutions are being implemented that increase the composition of 3 and more elements resting on common carriages.

In addition to the evolution of the structures, major transformations were performed in traction systems. Even in the trams, the DC engine was replaced by the three-phase drive motor. Power electronics has now been universally adopted for traction and speed control. Electricity is now captured by both an aerial line with a pantograph and a particular type of third rail underneath and drowned in normally insulated ground but which is subjected to the rolling passage. The newly designed accumulator system is also used again for tram traffic in historic buildings of particular architectural value.

Another element that has played a huge role in European public transport, is represented by the *subways*. After a fierce expansive period in the construction of new networks, between the end of the nineteenth century and the fifties of the twentieth century, where capitals and major metropolis of the northern hemisphere were mainly equipped with metropolitan areas, new programs were launched in this area since the seventies, due to the need to de-congest urban traffic from traffic and to the growth in oil prices, which has made it economically less costly for transport by car.

The first real metro line in the world was that of London, still called today *"Underground"* or *"The Tube"*. It began operating on January 10, 1863 (Metropolitan Line) and currently has 414 km of lines. The proposal seems to have been advanced by then-mayor Charles Pearson, motivated by the unbearable chaos on the streets of the city-centre also because of the lack of direct interchange between the various railway stations in the city. Thus in 1860 the Metropolitan Railway Company was established, whose name will be reported with the first line.

Until 1890 the London metropolitan was steam-powered, with open-air trains: only that year the electrification allowed it to be submerged, with the first en-

tirely underground line. Still in the United Kingdom are operating the ancient metropolitan of Glasgow (1896) and Newcastle upon Tyne. The first subway line in Europe was built in 1896 in Budapest, Hungary, and still today it is largely preserved in its original state, as in its name, *Földalatti*; It was also the first in Europe to have the electric traction provided by aircraft cables.

Nowadays Urban Public Transport system is characterized by various vehicles and by different network infrastructures, both in promiscuous and by wheel or rail, based on parameters such as, for example, expected or actual passenger demand a given path. However, the exponential growing of urban areas is giving rise to a wide range of problems, discussed in the following paragraph, which must be addressed also by the public transport service.

1.2 Recent Issues in Urban Transit

As anticipated in the previous paragraphs, urban areas are highly congested, in particular the historical part of city centres. The main reasons of congestion are principally: the high level of urbanization (about 70% of the European population lives in cities); and the conformation of historical city centres, characterized of narrow roads, in some cities also steep and with steps, where also conventional public transport operates with difficulty. Congestion causes delays in trips: it is estimated that about 1% of the European GDP is lost by considering the cost associated to urban congestion [3]. Congestions have also other bad aspects: the number of accidents increases proportionally to urban congestion, and, at the same time, causes an increase in pollution, whose long-term effects are dangerous for people's health. Therefore, a shift from private to public transport is desirable. This can be achieved through limitations on private transport, and through improvements of public transport. In order to avoid the decrease of the transport demand, public transport must be attractive and provide a high quality service. Moreover, the population is ageing. Transport systems must therefore take into account the mobility needs of all categories of users.

1.2.1 Air Pollution

Air pollution represents the largest environmental health risk in European Union, in fact some recent estimates suggest that the disease burden resulting from air pollution is substantial [4] [5]. Is proven that heart diseases and stroke, that represents the most common reasons for premature death, are attributable to air pollution and are, also, responsible for 80% of cases of premature death or lung diseases (Figure: 1.1). In addition to causing premature death, air pollution increases the incidence of a wide range of diseases like

respiratory, cardiovascular diseases and cancer, with both long and short-term health effects. The International Agency for Research on Cancer has classified air pollution in general, as well as particulate matter (PM) as a separate component of air pollution mixtures, as carcinogenic.

Recent studies [6], [7] have proven that air pollution is also responsible for the crescent rate of fertility or pregnancy issues. Also children may have several problems in growing. These problems include negative effects on neural development and cognitive capabilities, which in turn can affect performance at school and later in life, leading to lower productivity and quality of life. There is also emerging evidence that exposure to air pollution is associated with new-onset type 2 diabetes in adults, and may be linked to obesity and dementia.

People who live in big urban areas suffer more because they live in really high polluted areas and are exposed to higher levels of air pollution. In this case, an efficient urban public transport system which encourage people in using it, can help in reducing air pollution and can increase the QoL in big cities. Air pollution has, also, several important environmental impacts and may directly affect vegetation, as well as the quality of water and soil and the ecosystem services that they support. For example, ground-level ozone damages agricultural crops, forests and plants by reducing their growth rates. Other pollutants, such as nitrogen oxides, sulphur dioxide and ammonia, contribute to the acidification of soil, lakes and rivers, causing biodiversity loss. In addition to causing acidification, NH3 and NOX emissions also disrupt terrestrial and aquatic ecosystems by introducing excessive amounts of nutrient nitrogen. This leads to eutrophication, which is an oversupply of nutrients that can lead to changes in species diversity and to invasions of new species.

Air pollution and climate change are intertwined. Several air pollutants are also climate forcers, which have a potential impact on climate and global warming in the short term. Tropospheric O3 and black carbon, a constituent of PM, are examples of air pollutants that are short-lived climate forcers and that contribute directly to global warming. Other PM components, such as organic carbon, ammonium, sulphate and nitrate, have a cooling effect. In addition, changes in weather patterns due to climate change may change the transport, dispersion, deposition and formation of air pollutants in the atmosphere. For example, a warmer climate leads to an increase in ground-level O3 production, and increased O3 levels then contribute to more warming.

1.2.2 Street Accidents And Road Safety

In year 2016, 25.991 deaths for accidents have been registered in European area. Compared to 2015 (26.065 deaths), a very slight decrease is registered in

the number of deaths (-0.3%). Compared with the target set by the European Union in the White Paper of 2001 [8], much progress has been made with reducing the number of fatalities. The average reduction between 2005 and 2007 was 3,1% per year. The number fell more rapidly in the 2007-2010 period, with an average reduction of about 10%, which decreased the following years. It is estimated that the number of road accident fatalities in the EU fell by 42% between 2005 and 2014. The population of the EU countries grew by 2,5% over the decade, but the growth occurred mainly among the older age groups and indeed the population declined in the age groups between 10 and 44 years. Fatalities in the over 85 year old age group increased by 28% in 2014 compared with 2005, while the respective fatality rate decreased by 17%. There are, also, a lot of differences subsisting in deaths among the different countries of the European area. For example, in countries like Cyprus and Ireland the average age of people death in road accident is younger than in other countries like Italy, Portugal or Sweden (Figure 1.2). Far more males than females are killed in road accidents: 76% of all fatalities were male and 24% were female (Figure 1.3). The type of road also has a major impact on fatal accidents. Overall, only 7% of road fatalities in 2014 occurred in accidents on motorways, and 55% of road users died in accidents on non-motorway urban roads. It is clear that congestion of urban roads play a key role in road safety. The prove is represented by the fact that almost half of fatalities on urban roads were pedestrians or cyclists, and about one quarter were car occupants. This statistic is fairly clear and tells us that the main victims of extreme urban road congestion are the individuals who do not travel on vehicles but who proceed vulnerable on roadway.

1.3 Policies to improve sustainable mobility

As early as 1957, when the European Economic Community (EC) was created, the Member States agreed to develop a common transport policy. On 1992 (the same year as the Treaty of Maastricht) the EC issued the so called *White Paper* [8], representing a set of documents containing proposals for European Union action in a specific area, on the future development of the common transport policy. In 2001 the document was updated, setting medium-term objectives. The paper, titled, *European transport policy for 2010: time to decide* aims to guide the development of the European transport sector in a sustainable and modern direction. It proposes around 60 *measures* to develop a transport system capable of shifting the balance between modes of transport (wheel, railway, sea and air). In 2007 the EC published the *Green Paper on Urban Mobility: Towards a new culture for urban mobility.* It set the foundations for a new European agenda for sustainable mobility policy and invited stakeholders

to a debate on what support the EU should provide, and how best to provide it (eg. how to achieve optimal European added-value through the effective promotion of best practices).

The Green Paper identified five points that must be faced by big cities and urban areas:

- **Congestion**, which creates negative economic health, environmental and social impacts, and affects mobility not only at the local (city) level, but also long-distance transport routes which go through urban areas;
- **Dependence on fossil fuels**, which create a lot of air pollution that contributes in climate changing and reduce the human QoL;
- **Increase in freight and passenger flows**, in combination the limited possibility of expanding the transport infrastructure;
- **Accessibility to the urban mobility system**, which must possess the following characteristics: fast, frequent, comfortable, reliable, flexible, affordable and accessible to the more vulnerable groups;
- **Safety**, including he safety of infrastructures and of the rolling-stock, as well as citizens' safety in reaching the system.

In parallel to Green Paper, European Union an other document called *Sustainable Urban Transport Plans – Preparatory document in relation to the follow-up of the Thematic Strategy on the Urban Environment.* This publication focus its attention in other four points that must be relevant for European governments in planning sustainable mobility system. The points are the following:

- Technological progresses, alone, are not sufficient to obtain a sustainable mobility system;
- Efforts to achieve sustainable transport systems must be done in unison by local, national and continental realities;
- A close collaboration between the urban transport management and the land use planning departments is necessary to generate sustainable mobility synergies;
- Internalisation of external costs is suggested as a way to account for the full extent of societal costs involved in transportation.

In summary, the EU has dedicated efforts throughout the last few decades to try to create a common transport policy, and, in addition, in more recent years to try to reduce the strong negative impacts of the sector on the environment, human health and the economy.

1.3.1 Improvement of urban public transport system

The measures aimed to discourage the usage of private car are not effective if a high quality public transport is not provided to users. These measures therefore may result only in the decrease of the transport demand and on the settlement of some activities and residents in other areas, and not in a real shift from private to public transport. In order to perform a high quality Public transport, all the segments of demand must be satisfied, specially people living in far and low populated residential areas and people aged or with some disability.

In order to better capture percentages of transport demand, conventional public transport must be improved. Public transport is of high quality for users if it is:

- fast, i.e. transit times should be kept low;
- reliable, i.e. it should cross the stops and stations at the scheduled time. Reliability assume extreme importance when user trips involve different 18 means of transport: for example they have to take different bus lines, or tranship from train to bus or from bus to underground;
- capillar, i.e. it should reduce walking distances and therefore be capable to board the user as close as possible to his origin, and bring him as close as possible to his destination;
- frequent, at least 4 services by hour for urban areas; however if frequencies overcome 12 services by hour, there is no longer a benefit to users;
- with a reduced number of transhipments: much time is lost when users commute between different means of transport; moreover time spent waiting is perceived in a worse manner than time spent on board.

High quality of transport can be achieved through several measures; for example:

- build new lines of underground or improve the existing lines, when required;
- develop bus lanes in some major roads interested with high traffic flows and provide public transport a priority phase in signalized intersection, in order to decrease the journey time. This also result in a decrease of management costs as fewer vehicles are necessary to perform the service;
- in small and medium cities, where the demand is not enough for building an underground or other fast transit solutions, rapid bus lines, with only few stops, should developed.

- Build parking spaces in peripheral areas in proximity of public transport stops. Therefore commuters reach the urban area by private car and after commute to public transport.

Some small villages located in proximity of cities, or in isolated and low populated residential areas, where also the demand of transport is low, cannot be served by a high quality and frequent transport line. Therefore, for such situations on-demand services must be developed. In several small cities, some public transport lines have a main path and several areas around to serve on-demand. These lines have a high frequency, and deviate from the main path as they receive a request.

Country	Population	$PM_{2.5}$		NO_2		O_3	
		Annual mean (¹)	Premature deaths	Annual mean (²)	Premature deaths	SOMO35 (³)	Premature deaths
Austria	8 451 860	15.7	6 960	19.3	910	5 389	330
Belgium	11 161 642	16.6	10 050	23.6	2 320	2 520	210
Bulgaria	7 284 552	24.1	13 700	16.5	570	4 082	330
Croatia	4 262 140	16.8	4 820	15.8	160	5 989	240
Cyprus	865 878	17.1	450	7.3	< 5	7 900	30
Czech Republic	10 516 125	19.6	12 030	17.1	330	4 266	370
Denmark	5 602 628	9.6	2 890	13.0	60	2 749	110
Estonia	1 320 174	7.8	690	10.8	< 5	2 545	30
Finland	5 426 674	5.9	1 730	9.4	< 5	2 011	80
France	63 697 865	14.5	45 120	18.7	8 230	4 098	1 780
Germany	80 523 746	14.2	73 400	20.4	10 610	3 506	2 500
Greece	11 003 615	19.7	13 730	14.6	1 490	8 532	840
Hungary	9 908 798	18.2	12 890	16.8	390	4 604	460
Ireland	4 591 087	9.2	1 520	11.6	30	2 043	50
Italy	59 685 227	18.2	66 630	24.5	21 040	6 576	3 380
Latvia	2 023 825	12.8	2 080	13.7	110	2 614	60
Lithuania	2 971 905	13.9	3 170	11.5	< 5	2 703	90
Luxembourg	537 039	14.3	280	23.4	80	3 167	10
Malta	421 364	12.5	230	12.0	< 5	7 403	20
Netherlands	16 779 575	14.3	11 530	21.3	1 820	2 410	270
Poland	38 062 535	22.8	48 270	16.1	1 610	3 792	1 150
Portugal	9 918 548	10.0	6 070	14.0	150	5 091	420
Romania	20 020 074	18.5	25 330	17.9	1 900	2 221	430
Slovakia	5 410 836	20.1	5 620	16.0	< 5	5 116	200
Slovenia	2 058 821	17.4	1 960	17.6	150	6 540	100
Spain	44 454 505	11.0	23 940	18.0	4 280	5 895	1 760
Sweden	9 555 893	6.0	3 020	11.5	< 5	2 317	160
United Kingdom	63 905 297	11.8	37 930	22.8	11 940	1 606	710
Albania	2 874 545	20.3	2 010	15.9	10	7 179	100
Andorra	76 246	11.9	40	14.3	< 5	7 303	< 5
Bosnia and Herzegovina	3 839 265	16.0	3 620	15.7	80	5 670	180
former Yugoslav Republic of Macedonia	2 062 294	30.4	3 360	20.8	210	6 326	100
Iceland	321 857	6.5	80	14.3	< 5	1 473	< 5
Kosovo (⁴)	1 815 606	28.0	3 530	19.3	230	5 691	100
Liechtenstein	36 838	11.4	20	22.7	10	5 221	< 5
Monaco	36 136	13.8	20	23.2	10	7 795	< 5
Montenegro	620 893	17.1	600	17.2	30	6 674	30
Norway	5 051 275	7.1	1 590	14.4	170	2 443	70
San Marino	33 562	15.1	30	15.4	< 5	5 067	< 5
Serbia	7 181 505	21.1	10 730	20.2	1 340	4 505	320
Switzerland	8 039 060	13.9	4 980	22.4	1 140	4 919	240
Total (⁵)			**467 000**		**71 000**		**17 000**
EU-28 (⁶)			**436 000**		**68 000**		**16 000**

Figure 1.1: Premature deaths attributable to PM2.5, NO2 and O3 exposure in 41 European countries and the EU28 in 2013 [1].

	0-14	15-24	25-59	60-99	Total	Median age
BE	1%	19%	50%	30%	727	43
BG	3%	21%	52%	23%	901	38
CZ	2%	16%	57%	25%	688	44
DK	3%	14%	45%	38%	182	47
DE	2%	17%	46%	35%	3.377	49
EE	1%	19%	49%	31%	78	45
IE	3%	21%	48%	27%	188	37
EL	1%	17%	52%	30%	795	45
ES	2%	9%	55%	34%	1.688	49
FR	3%	21%	49%	27%	3.384	40
HR	3%	13%	53%	31%	308	47
IT	2%	13%	47%	38%	3.381	50
CY	0%	31%	40%	29%	45	37
LV	4%	17%	54%	26%	212	43
LT	6%	16%	51%	27%	267	44
LU	3%	14%	74%	9%	35	35
HU	2%	11%	56%	31%	626	49
MT	8%	31%	54%	8%	13	29
NL	4%	18%	38%	40%	476	48
AT	2%	17%	47%	33%	430	49
PL	3%	18%	51%	29%	3.202	44
PT	1%	10%	50%	39%	638	51
RO	5%	12%	52%	31%	1.818	48
SI	2%	19%	50%	28%	125	46
SK	2%	17%	52%	29%	321	47
FI	4%	19%	45%	31%	229	43
SE	3%	12%	43%	42%	270	53
UK	3%	20%	47%	30%	1.854	43
EU	**3%**	**16%**	**49%**	**32%**	**26.258**	**46**
IS	0%	50%	50%	0%	4	25
NO	3%	15%	43%	39%	147	48
CH	4%	16%	40%	40%	243	51

Figure 1.2: Total number and distribution of road fatalities by country and age group in European Area, 2014 [1].

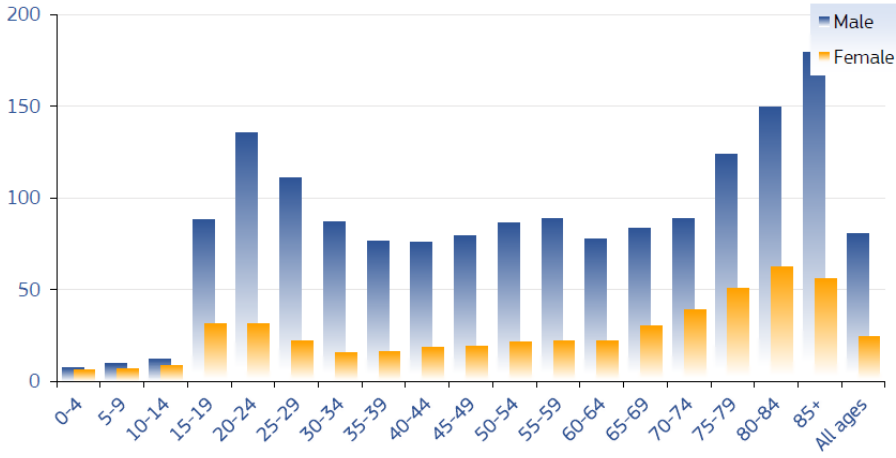


Figure 1.3: Road fatality rates per million population by age group and gender in European Area, 2014 [1].

Chapter 2

The ontology-based framework to support performance monitoring in PTS

2.1 Overview

In this chapter the ontology framework to monitor the performance of a Public Transport Service will be introduced and discussed. After carrying out a thorough analysis of concepts such as Knowledge Management (KM), Decision Support System (DSS) and Ontology used in decision making processes, the innovative part that this work introduced in urban public transport sector will be shown, in more detailed way. The proposed novel approach is based, mainly, on two concepts:

- A **Knowledge Base** composed by an ontological representation of all the knowledge regarding indicators and their formulas, business objectives, dimension analysis and their relation with the Transmodel Data Model (TM), the European reference data model for public transport information systems;
- A **Reasoning Framework** made in Logic Programming (LP) that provides logic functionalities to interactively support designers in a set of common and high level design tasks.

For each of the above-described elements, in the following sections, a detailed analysis will be carried out and, after that, a case study will be discussed highlighting the high-level tasks provided by the framework to PTS managers.

2.2 Related Work

In Literature the problem of extracting information and Knowledge from the available data has been addressed several times and it has been analysed from

different perspectives. The increase of data storage systems' capacity and the subsequent growing need to analyse such data, have pushed many researches in developing new techniques and algorithms in order to extract information from the large amount of data available [9]. In recent years, for example, the emerging of Data Warehouses and Big Data as ones of the most used data repository architectures [10] allowed companies and public institutions to store a huge amount of data.

However, storing a large amount of data is not enough for the intended purpose. In order to extrapolate information useful to the decision-making process from the huge amount of data available, techniques and methodologies are needed to select only useful information and separate them from the potentially misleading or damaging one. For example, in [11] the proposed approach covers the entire lifecycle of a DataWarehouse where the user can check the relationships that exist among the various qualitative factors in such a way that they can be optimized in order to fulfil specific quality goals. Even the most up-to-date data usage is of high importance in decision-making process as demonstrated in [12] where the authors presented a methodology on how to adapt data warehouse schemas and user-end OLAP queries for efficiently supporting real-time data integration. To accomplish this, authors used techniques such as table structure replication and query predicate restrictions for selecting data, to enable continuously loading data in the data warehouse with minimum impact in query execution time. They have also shown the method's efficiency in a real world case study. In [13] authors have developed a different approach by considering an hybrid system, called HyPer, that can handle OLTP and OLAP systems simultaneously by taking advantage of hardware-assisted replication mechanisms in order to maintain consistent snapshots of the transactional data. The utilization of the processor-inherent support for virtual memory management raise the performances of the system ensuring both high transaction rate (10000 transactions per second) and systems' stability.
Also Big Data has drawn huge attention, in the last years, from researchers in information sciences. As mentioned above, it is known that the speed of the information growth higher than Moore's Law and some new techniques must be considered to handle the problem. In [14] the authors have focused their attention on the opportunities that Big Data's systems offers by considering both Big Data applications and the state-of-the-art techniques and technologies usually adopted to deal with the Big Data problems. The authors of [15], instead, illustrated how the big data technology is influencing the cloud computing by introducing some relationship subsisting among Big Data storage systems, Hadoop technology and cloud computing. In [16] and [17] researchers faced the problem of performing Data Mining in Big Data's envi-

ronments. More specifically in [16] author presented a broad overview of the topic by considering, principally, four articles covering the most interesting and state-of-the-art topics on Big Data mining. In [17], instead, a new Big Data processing model that implements the HACE theorem involving, at the same time, demand-driven aggregation of information sources, mining and analysis, user interest modelling, security and privacy considerations.

However, the extraction of useful information is only a step of the process of creating knowledge from the available data. Such information must be organized and managed in such a way that it may be useful for the purposes previously assessed. The task of managing and organize the available information is called *"Knowledge Management"* (KM) that, literally, represents the systematic management of knowledge assets for the purpose of creating value and meeting, at the same time, tactical and strategic requirements [18]. In literature, this concept has been discussed several times and from different perspectives/purposes allowing the development of various theories and methodologies ([19], [20], [21] and [22]). The technologies used to manage the KM issue can be divided into seven categories [23] as: KM Framework ([24], [25]), knowledge-based systems ([26], [27]), data mining ([28], [29]), information and communication technology ([22], [30]), artificial intelligence/expert systems ([31], [32]), database technology and ITS application ([21], [33]) and modelling ([34]).

The choice of the KM class type is a part of the problem-oriented domain and it is strongly influenced by the class of the problem that must be addressed.

As previously said, one of the possible solutions is represented by frameworks that allows to organize and structure the information extracted. Some of these frameworks use a set of Key Performance Indicators (KPI) as performance metrics in order to support the decision-making process, giving rise to the Decision Support System (DSS). Several works and project, like SCOR [35] and VRM [36], have focused their attention in developing standard templates and guidelines in order to define the best set of indicators in order to facilitate the decisional process that must be performed. However, this topic is not easily standardized and results to be heavily influenced by the application domain. Various researches have been carried out to apply such methodologies to the most varied analysis domains, as in [37] where authors have developed a number of templates to be implemented within a real company. They have also presented a real word case study where the templates are used to identify KPIs for a manufacturing solution. In [38] authors describe a KPI modeling environment, coined Mozart, where modellers can use formal models to explicitly define the services of KPI and their relationships which are depicted by KPI net. They applied this methodology in an example scenario where they try to mining KPI from an automobile dataset to generate a monitor model.

In recent years, however, organizations and companies are changing their

structures and architectures in order to become more collaborative with each other, trying to break down the barriers to interoperability. These barriers, in Literature, are divided into three different categories [39]:

- Conceptual barriers: They are concerned with the syntactic and semantic differences of information to be exchanged. These problems concern the modelling at the high level of abstraction (such as for example the enterprise models of a company) as well as the level of the programming (for example XML models).

- Technological barriers: These barriers refer to the incompatibility of information technologies (architecture and platforms, infrastructure...). These problems concern the standards to present, store, exchange, process and communicate the data through the use of computers.

- Organisational barriers: They relate to the definition of responsibility (who is responsible for what?) and authority (who is authorised to do what?) as well as the incompatibility of organisation structures (matrix vs. hierarchical ones for example).

In order to break down these barriers some solutions have been proposed. In [39] author has developed an Enterprise interoperability framework formed by three different dimension: Enterprise dimension representing enterprise levels, Interoperability dimension representing interoperability barriers and the Interoperability approaches dimension that allows categorising knowledge and solutions relating to enterprise interoperability according to the ways of removing various interoperability barriers. Other researches, instead, have faced the issue observing it from a different perspective. In [40], for example, authors propose a logic model for the representation of KPIs that supports the construction of a valid reference model (or KPI ontology) by enabling the integration of definitions proposed by different engineers in a minimal and consistent system. Also in [41] authors have exploited ontologies to build a collaborative platform dedicated to musical instrumental practice by describing both technical aspect with an ontology and pragmatical aspect with a descriptive model.

It is evident that the use of ontology can represent the keystone to break down the interoperability barriers, as shown in [42] where is argued that ontologies in particular and semantics-based technologies in general will play a key role in achieving seamless connectivity.

2.2.1 Decision Support System

Researchers began to show interest in this topic around the 1960s when Scott Morton in 1971 studied how computing and mathematical models could help

managers in the Decision Making process by using some recurring business keys. At the start of 70's, journals started to publish articles on management decision system, strategic planning system and decision support system. The first use of the term *Decision Support System* occurred in [43] where authors argued that *"Supporting information systems for semi-structured and unstructured decisions should be termed Decision Support System"*.

Since then, DSS has been one of the most popular research topics for the scientific community given the innumerable application possibilities. In years various models of DSS were investigated and classified in different ways and from different perspectives. For example, in 1977 Donovan and Madnick described the first DSS classification in: **Institutional DSS** supporting ongoing and recurring decisions and **Ad hoc DSS** that supports a one off-kind of decision. In 1981 Hackathorn and Keen classified DSS as: **Personal DSS**, **Group DSS** and **Organizational DSS** while Alter, in the same years, opined that decision support systems could be classified into seven types based on their generic nature of operations. He described the seven types as: **File drawer systems**, **Analysis information systems**, **Accounting and financial models**, **Representational models**, **Optimization models** and **Suggestion models**. However, the most recent and used classification was made by J. Power in 2002 when, starting from model-driven DSS up to knowledge-driven DSS, he classified the DSS models in [44]:

- **Model-Driven DSS**: it emphatizes to and optimization and/or simulation models. It use, usually, limited data and parameters provided by user to aid the decision makers in analyzing situations;
- **Data-Driven DSS**: it emphatizes the access and manipulation of time series data belonging to a single organizational subject or belonging, sometimes, to external data sources. Data-Driven DSS with on-line analytical processing provide highest levels of functionality and decision support linked to the analysis of a large collection of historical data;
- **Group Communication-Driven DSS**: it uses network and any sort of communication technology to help decision makers in the decisional process. In this model of DSS the communication technology (like groupware, computer-based bullettin boards and video conferencing) represents a dominant architectural component;
- **Document-Driven DSS**: it uses principally computer storage and processing technologies in order to perform an accurate document analysis. The documents that can be accessed by this model of DSS are: product specifications, policies and procedures, catalogs and corporate historical documents like minutes of meetings and corrispondence;

- **Knowledge-Driven DSS**: this type of DSS can be used to drive managers in choosing the best decision possible by suggesting or recommending actions. These category of DSS are person-computer systems with specialized problem-solving expertise. To apply this model of DSS a particular knowledge of the application domain is required;

- **Web-Based DSS**: this particular category of DSS tends to exploit the potential of the World-Wide-Web in order to enhance the capabilities and the deployment of computerized decision support.

The above mentioned DSS models, in years, have been applied in a wide variety of application domains. As for Model-Driven DSS (also called in the past Model-Oriented or Computationally-Oriented DSS [45]) some works have been proposed. In [46] authors presented a Model-Driven architecture called MODA-CLOUDS (*MOdel-Driven Approach for the design and execution of applications on multiple CLOUDS*) with the aim to support system developers and operators in exploiting multiple Clouds for the same system and help them in migrating the systems from Cloud to Cloud when needed. The developed Model-Driven DSS approach helped authors in abstracting the complexity of Cloud platforms and allowed early definition of quality at design time. In [47] authors have focused their attention in investigating different strategies of developing a Model-Driven DSS. In a detailed way they have studied the effects of three particular choices that must be made during the development of a DSS: user vs. system-guided model manipulation, variable vs. exception-based report content and display of incremental changes vs. actual outcomes on strategy formulation. They have performed several laboratory experiments with the help of 46 undergraduate business students. At the end, with the findings obtained, authors suggest that a system-guided or a more structured model manipulation strategy and the display of incremental changes will significantly improve performance of a DSS. However Model-Driven DSS have not had much success in supporting the decision making process. One of the main issues is represented by a mismatch between DSS design/performance and the requirements of decision makers. The causes of the mismatch can derive of two different problem categories: technical (poor response times) or non-technical (different personal preferences) [48].

Data-Driven DSSs have reported some success cases principally in 90's, but one of the first DSS of this category, was built in 1974 by Richard Klaas and Charles Weiss at American Airlines. In 1990 the raise of data warehousing and On-Line Analytical Processing (OLAP) allows to define a broader category of Data-Driven DSS. These concepts were used by Bill Inmon and Ralph Kimball (known, respectively, as *"father"* and *"doctor"* of DSS) to promote the

building of DSS with the guidance of relational database technologies. In more recent years, other tools have been added to Data-Driven DSS in order to create Web-Based dashboards and scorecards [49]. As in Model-Driven DSS case, Data-Driven approach has been applied on different topics and domains also in recent years. In [50], for example, authors have applied an emerging computer model called DDDAS (Dynamic Data-Driven Application Systems) in support of emergency medical treatment decisions in response to a crisis. The considered complex multi-layered dynamic environment has been demonstrated that both feeds and responds to an ever-changing stream of real-time data that enables coordinated decision-making by heterogeneous personnel across a wide geography at the same time. In [51] an elaborate analysis of six different case studies is presented in which is shown that database usage and information processing practices have indeed grown more sophisticated and the implementation of more complex analytical database architectures, like data-warehouses and data-marts, are doing well in the technological landscape. Authors have also introduced a business intelligence value chain model that helps decision makers in the building phase of a decision making environment.

In the late 80's, the growing interest of academic researchers in studying a new type of software used to support the group-decision making, has pushed DeSanctis and Gallup [52] in focusing their attention in defining the theoretical foundations of the research area called group decision support systems (GDSS) or Group Communication-Driven Support Systems. They have made a conceptual overview of GDSS based on an information-exchange perspective of the decision making process in such a way that they could define three different levels of systems representing varying degrees of intervention into the decision process. Level 1 GDSS software where represented as software that used some tools and third party instruments to reduce communicative barriers among decision-makers. Level 2 GDSS, instead, have more sophisticated instruments that can be used like the use of tools that would allow the implementation of problem structuring techniques. Level 3 GDSS systems are characterized by machine-induced group communication patterns and can include expert advice in the selecting and arranging of rules to be applying during a meeting.

In the same years, Tung Bui and Matthias Jarke [53] have developed a framework in order to develop a Communications component for the GDSS. The component supports conceptualization of a communication system as being composed of four main modules: the Group Norm Monitor, the Group Norm Filter, the Invocation Mechanism, and the Individual Decision Support System (IDSS)-to-GDSS Document Formatter.

In the late 70's and early 80's, also the use of digital documents in the decision-

making process attracted the interest of the academic world. In 1978 E. Burton Swanson and Mary J. Culnan built the first framework used to classify document-based information systems for management planning and control and made a literature survey in order to perform some examples and classify then in accordance with the developed framework. Twenty-two illustrative systems are identified, described, and classified along two major lines of development [54]. The research of this type of DSS continued; in the three year period 93-96 J. Fedorowicz helped to explore and highlight the need of that systems [55].

Knowledge-Driven DSS emerged in 1980 when Alter proposed a model to develop a *suggestion* DSS [56]. Klein and Methlie in 1995 called this type of systems knowledge-based DSS while Goul, Henderson, and Tonge in 1992 examined Artificial Intelligence (AI) contributions to decision-making process. In 1965, a Stanford University research team led by Edward Feigenbaum created the DENDRAL expert system. DENDRAL led to the development of other rule-based reasoning programs including MYCIN, which helped physicians diagnose blood diseases based on sets of clinical symptoms. The MYCIN project resulted in development of the first expert-system shell. In 1983, Dustin Huntington established EXSYS. That company and product made it practical to use PC based tools to develop expert systems. By 1992, some 11 shell programs were available for the MacIntosh platform, 29 for IBM-DOS platforms, 4 for Unix platforms, and 12 for dedicated mainframe applications. Artificial Intelligence systems have been developed to detect fraud and expedite financial transactions, many additional medical diagnostic systems have been based on AI, expert systems have been used for scheduling in manufacturing operation and web-based advisory systems. In recent years, connecting expert systems technologies to relational databases with web-based front ends has broadened the deployment and use of knowledge-driven DSS.

The interest for Web-Based DSS started approximatively in 1995 when the World-wide Web and global Internet services provided the instruments to extend the computerized capabilities of the decision support. Tim Berners-Lee [57] in 1996 introduced some interesting infrastructures and platform in order to share information about decision support and to define a new means of delivering decision support capabilities. Power immediately exploited these newly introduced features by presenting in [58] some examples of Web-Based DSS implementation and some thoughts about the opportunities offered by World Wide Web and DSS technologies. In [59] authors, in 2001, made a survey of Decision Support Systems in the context of developments in Web Technologies. They suggested to address the research in the coming years to overcome the three main challenges categories: Technological challenges, Economic chal-

lenges and Social and Behavioural challenges.

Decision Support System in Public Transport

Much research has been conducted to define standard criteria in order to improve the evaluation of procedures and processes related to the public transport sector. Comprehensive collections of data and objective measures are made available by several work with the aim of evaluating the *Quality of Service* (QoS). For example, in [60] the QoS is evaluated by studying the *Transit Capacity, Speed, Reliability* and *Position of transit stops or stations*, while in [61] an extensive overview and an interpretative review of KPIs are presented, aimed to suggest the set of suitable performance indicators. In [62] the measures related to the quality of service were adopted in order to evaluate the transit system in Kelvin Grove Urban Village. Authors of [63] and [64] have exploited the potential of the latest global location technologies (GPS) to measure and, eventually, improve the scheduling efficiency in public transport system. Due to the spread of Web 2.0 and social media applications, some research has focused their attention on studying the impact that these technologies can perform in terms of efficiency and safety of a PTS. Authors of [65], tacking advantage of the traffic sentiment analysis (TSA), proposed some models used to improve the safety of a PTS, analysing, at the same time, both advantages and disadvantages of solutions adopted. The sustainability of the Public Transport System is also considered a critical aspect during decision making activities. In [66], performance measures used to evaluate the environmental, economic and social sustainability of a PTS are discussed.

More similarly to this work, several approaches in the literature propose systems aimed to support monitoring of public transport services. In [67] a method to model a public transport system, depending on the result obtained by well-defined performance indicators, is presented and discussed. Authors of [67] present a model consisting of a combination of several analytical computer programs that, taking advantage of the information obtained from the performance indicators, seeks to restructure the local public transport system.

In [68], using well-established approaches in the literature and past benchmarking experiences, the development of a standardized Measurement System to clearly evaluate the performances of a generic PTS is provided. Authors of [69] have focused on developing and improving the performance measurement systems of PTS in order to support the transport managers in decision making processes. In particular, a guidebook has been presented in order to develop a measurement system to evaluate the transport services from both customer and PTS perspectives.

Different standards have also been introduced with the aim to support the design of a PTS software. In [70] the Trasmodel Data Model (TDM) is in-

troduced and discussed. Other standards, like NTCIP, can be considered in designing PTS software in order to improve system interoperability. However, only a few work document a real implementation of these standards. An example in this sense is [71], where TDM has been exploited to improve the reliability of the public transport services in the city of Trieste (Italy). Such standards are extremely useful to facilitate interoperability among software, designed by different software houses, that describe the same application domain. However, along these standards, specific solutions are needed to provide guidance to deeply recognize relationships among the concepts handled by PTS software. To the best of our knowledge, besides the specific techniques adopted by our approach, this is the first work to address a support framework for the choice of KPIs for a management system relying on the Transmodel.

2.2.2 Ontologies in Data Representation

In the last 15 years if people think about the word *"ontology"* has immediately recalled *"technical"* concepts such as semantic reasoning and knowledge representation. Researchers, instead, knows, that the notion of ontology originates from a remote past far from the modern and technological world of computers and internet. From a philosophical perspective the word ontology literally means *"the study of the nature of being as such"*. The first reference to a concept that can be associated with the word *"ontology"* was made by Aristotle in [72] where he defined a list of categories and types that can represent an inventory of what there is in terms of the most general kinds of entities. These categories can be used to differentiate things as well as to refine specific aspects of things [73] and can represent the first recognized ontology containing the following concepts:

- **Substance** (e.g.:man, dog, horse, etc.);
- **Quantity** (e.g.: one litre, two kilos, etc.);
- **Quality** (e.g.: red, tall, small, etc.);
- **Relation** (e.g.: half valued, double, etc.);
- **Place** (e.g.: at central square, on the street, etc.);
- **Time** (e.g.: one hour ago, yesterday, tomorrow, etc.);
- **Position** (e.g.: in the back, in the front, etc.);
- **State** (e.g.: stopped, armed, etc.);
- **Action** (e.g.: to stop, to launch, etc.);

- **Affection** (e.g.: to be stopped, to be launched, etc.);

The most important characteristic of these categories is that they are not composite and they can be composed in order to make statements about the nature that can yield affirmation. The word *"ontologia"* was born in Germany in around 1600 and was coined by by Rudolf Gockelin [74], a German scholastic philosopher. In 1721 Smith defined the word ontology as *"an Account of being in the Abstract"* [75]. Science began to show interest in the concept of ontology in 1967 when, for the first time, the word "ontology" was associated with computer and information science in the work made by George H. Melay [76]. In this work Mealy tried to answer many question about the abstract concept of *"data"* (what data are? how data should be fed and cared for? or, what is the relation subsisting among data and how data can be represented with programming languages and operating systems?). To answer these questions Mealy proposed a theoretical model for data and data processing. The model was composed by sets of entities, values, data maps and procedure maps.The entities correspond to the objects in the real world about which data are recorded or computed. The data maps assign values to attributes of the entities; these maps are regarded as being sets of ordered pairs of entities and values, or data items. This represents the first work of data modelling which exploits ontology concepts to represent data. From this moment the term *"ontology"* became really popular in the field of computer science and more specifically in domains such as knowledge management, knowledge engineering, cooperative information systems, natural language processing or intelligent information integration. One of the most important work is [77] in which authors presents a formal ontology based on meta properties built around the *"fundamental philosophical notions of identity, unity, essence and dependence"*. The formal ontology is a part of the methodology called *Ontology-Driven Conceptual Analysis* (ODCA) which combines formal ontology's concepts with the needs of information system design. During the same period, the growing interest in the concept of ontology has resulted in a large number of ontologies belonging to the most disparate domains. This caused the rise of another need that developers and systems engineers had to face: the possibility to *reuse* concepts of a specific domains but build for other domains. Reuse is one of the principles of programming methods, and it is also valid for knowledge and ontology. In [78] authors have faced the rise to the need for suitable architectures for knowledge and concepts sharing by focusing their attention in the analysis of ontologies integration in a way such that different inheritance mechanisms within the ontology are supported, and the conflicts resulting to multiple inheritance will be resolved. At the end they propose a semi-automatic method to face with the two main conflicts just described by using a knowledge model that extends the usual frame-based model in order to associate with each attribute a degree

of strength and other information concerning the behaviour of the attribute. In [79] authors presents a novel approach to ontology module extraction that aims to achieve more efficient reuse of very large ontologies; the motivation is drawn from an Ontology Engineering perspective. Authors provide a definition of ontology modules from the reuse perspective and an approach to module extraction based on such a definition. At the end they demonstrated that the developed tool can generate small modules that retain the properties of the original ontology facilitating the reuse of ontological concepts defined for other domains.

Separating the domain knowledge from the operational knowledge is another common use of ontologies, similar to the method adopted in object-oriented programming. Analysing domain knowledge is possible once a declarative specification of the terms is available. Ontologies are composed by formal concepts about a specific domain, and thus need a formal logical language associated to them. A model that can be used at this purpose is represented by the Description Logics (DLs) that are a family of knowledge representation languages that can be used to represent the knowledge of an application domain in a structured and formal way. Many languages have been used to represent ontologies such as XML, XMLS, and RDF but, recently, a work-group at W3C produced recommendation that gives the birth to the Web Ontology Language (OWL) [80]. This language is designed to be used by applications that need to process the content of information instead of just presenting them to humans. It allows to machine to interpret the Web contents better than other proposed languages such as XML, RDF, and RDFS. OWL is derived syntactically by RDF, but it adds more vocabulary for describing properties and classes: among others, relations between classes (e.g. disjointness), cardinality (e.g. *"exactly one"*), equality, richer typing of properties, characteristics of properties (e.g. symmetry), and enumerated classes. The basic elements of the OWL ontology are classes, properties, instances of classes, and relationships between these instances:

- **Class**: A class is a collection of individuals (object, things, etc.) and it is the most basic concept for describing part of the world. Every individual in the OWL world is a member of the class owl:Thing. Domain specific root classes are defined by simply declaring a named class;

- **Individual**: An individual is an object of the world, and in particular a member of a class. Individuals are related to other objects and to data values via properties;

- **Property**: A property is a binary relation that lets us assert general facts about the members of classes and specific facts about individuals

(e.g. hasFather, hasPet, serviceNumber). There are two types of properties: datatype property and object property. While the former expresses relations between instances of classes and RDF literals and XML Schema datatypes, the latter expresses relations between instances of two classes.

OWL includes three different sub-languages classified in according to language expressibility [80]: **OWL Lite** that supports those users primarily needing a classification hierarchy and simple constraints. For example, while it supports cardinality constraints, it only permits cardinality values of 0 or 1. It should be simpler to provide tool support for OWL Lite than its more expressive relatives, and OWL Lite provides a quick migration path for thesauri and other taxonomies, **OWL DL** that supports those users who want the maximum expressiveness while retaining computational completeness (all conclusions are guaranteed to be computable) and decidability (all computations will finish in finite time) and **OWL Full** that is meant for users who want maximum expressiveness and the syntactic freedom of RDF with no computational guarantees. For example, in OWL Full a class can be treated simultaneously as a collection of individuals and as an individual in its own right.

Ontologies in the DSS design for Public Transport Systems

In previous section ontologies and how they have been applied in the computer science engineering were presented and discussed. In the present section some examples of ontologies' application in the public transport domain, in parallel to the use of DSSs, will be shown and discussed. In order to drive the development of performance monitoring systems, several work focused on models capable to properly catch all the relevant knowledge in the PTS domain, including relationships among business objectives, KPIs and data. In [81] a Knowledge Base Management System is defined to support the design of business objectives satisfying predefined levels of performances, while in [82] authors propose a model aimed to profile the unscheduled transport network in Mexico City. In [83], a knowledge base is used to design a DSS named *Transportation Sustainability Decision Support System* (TSDSS), in order to improve the sustainability of the public transport service in the Shangai region. In [84] the authors propose an innovative fuzzy ontology-based sentiment analysis algorithm that helps, with the support of a semantic web rule language (SWRL), the analysis of the main transportation activities by studying the social networks tweets.

In order to represent more expressive relations among concepts in a formal fashion, to support more advanced analysis tools and methodologies, and for the purpose of integration of distributed data sources, recent work propose to structure knowledge about KPIs into ontologies [85, 86, 87].

Similarly to our approach, authors of [85] propose an ontology for a knowledge-

based system for performance evaluation in a PTS. They discuss a taxonomy of KPIs mainly based on business objectives, and provide a set of datatype properties for storing values about various aspects like the rate of services. On the contrary, our ontological model deals with generic indicator definitions, which may be used as a reference for specific implementations to deal with run-time observations. On the other hand, in [86] and [87] authors propose ontologies partially overlapping with the Transmodel, aimed respectively to facilitate information retrieval for an application for user travel planning and determine the best path between two points. The former makes use of rules in SWRL to allows to perform reasoning to classify journey patterns.

Among the main issues that must be addressed in the development of a DSS, heterogeneity of distributed data sources is one of the most critical. Authors of [88] developed an ontology-based spatial context model that uses a combined approach to model the contextual information. The model called *"Primary-Context Model"* allows to facilitate interoperability among independent transportation systems while an ontology permits the reasoning upon available information. In [89] a transportation ontology is used to support the building of custom user-interfaces for transportation interactive systems.

As formal conceptualizations of a specific domain [90, 91], ontologies in the realm of performance monitoring can be used to both integrate heterogeneous data sources and improve the understanding of the measures provided [40].

In [92, 87, 93, 94] some ontologies are considered for modelling the PTS domain. In more detail, in [92] a urban features ontology, aimed at defining physical objects in urban environment, is extended by considering the Public Transport Ontology that describes the main concepts present in a Public Transport System domain. The obtained multi-modal ontology is used to both integrate the transit informations coming from multiple agencies and improve performances and effectiveness of a multi-modal passenger information system, providing more information on general service operations and itinerary planning. In [87] another simple Public Transport Ontology is defined in order to implement an improved public transfer query algorithm that helps customers of a PTS in minimizing the transfer times from the start point to the arrival point of interest. In [93] two ontologies that describe the PTS domain from both traveller and Public Transport System perspectives are defined in order to prove that the traveller perspective can be represented as a subset of the PTS, leading to the idea of nested ontologies. Authors of [94] discuss an approach based on a shared domain ontology for integration of transport data on railway services coming from multiple heterogeneous data provider, within the InteGRail European project.

All these proposals focus on domain ontologies for the transport domain, while in this work we take a different approach, as we refer to an ontological

meta-model for the Transmodel which is independent from the specific vocabulary used by the data model. A similar approach is taken by [95], where authors propose to represent a portion of Transmodel and IFOTP models in RDF as Linked Data. However, their modeling approach focuses only on a small subset of the Transmodel data model, namely related to points of access to vehicles and paths between them, and also the purpose of the work, namely data integration of disparate data sources to find optimal routes in a public transport service, differentiates this work from ours.

Ontologies are also useful in various domains to provide a logical representation of KPIs by defining explicit relationships among different indicators. Formal models of indicators were recently proposed [96, 97], providing means to evaluate the consistency of formally represented KPIs for design/specification of organizations, also enabling reuse, exchange and alignment of knowledge and activities between organizations.

In [98] the notion of *composite indicators* and their representation in a tree structure is introduced, together with their calculation with full or partial specification of the formula relating the indicator to its components.

However, in most of these papers, formula representation does not rely on logic-based languages, hence no inference mechanism and formula manipulation is possible. Recently, in [99, 100] a formal representation for formulas is used to build reasoning systems that exploit these relationships in order to calculate KPIs by manipulating automatically their mathematical formulas. In particular, our previous work [99] is focused on supporting automatic calculation of KPIs defined by multiple organizations, for cross-organization monitoring of shared business processes.

2.2.3 Management Software Standards for Public Transport Systems

In the world, different management software has been created in order to manage all the components that define a public transport service. Initially, the developed management software (MS) tended to reflect the specific requirements of the final customer without respecting any design or data modelling standards. However, the growing need of interoperable software, required the creation of some guidelines and common rules to make the communication of MS made by different software house quick and easy. This standardization process become necessary by considering that various public transport systems, albeit managed in different ways, have fundamentals concepts in common. It is a fact that all public transport systems have to face the management of six different domains [101]:

- **The service planning**: representing the planning of the service that

takes place weeks or days before the actual trip. It comprises a set of different activities used to plan the various components of a public transport service, like: network planning, timetable compilation, vehicle scheduling, drivers scheduling, etc.;

- **The dispatch of the service**: representing the assignment of a specific driver and vehicle to the timetable and duty scheduling;
- **The Transport control service**: representing the process of controlling and checking the fleet that is currently running. The control center has an overview of the traffic situation; when irregularities occur (delays, accidents, etc.), it can quickly find solutions or initiate actions. In addition to this, the vehicles can have on-board computers with communication and control functions, which can be used to control traffic lights or the displays in the vehicle;
- **The passenger information service**: that can serve the passengers by providing informations about the current departure times first and foremost at stops or by providing journey and timetable informations;
- **The ticketing service**: that provides the transport company of its revenue. In order to work properly, a fare structure for efficient ticket sales is required;
- **The evaluation service**: that compares the planned and actual values and stands at the end of the chain.

Starting from these concepts several standards have been developed, each focusing on some particular aspects of public transport service. In the next subsections, some of the most widespread in the world will be analysed.

Service Interface for Real-time Information (SIRI)

The Service Interface for Real Time Information (SIRI) represents the European Standard in exchanging information about the services provided in real-time by public transport between different computer systems. Like any self-respecting standard, it comprises a set of discrete functional services used to operate and modelling transport information systems by incorporating various national and proprietary standards from across Europe and unifying them through XML schemas and modelling concepts. The core concept on which all the features provided by SIRI are developed, is represented by a standardized communication layer made up, primarily, by a set of Web services. SIRI is intended to be used to exchange information between servers containing real-time public transport vehicle or journey time data. These include the control centres

of transport operators and information systems that utilise real-time vehicle information to operate the system, and the downstream systems that deliver travel information to the public over stop, on-board displays, mobile devices, etc. This standard tends also to carefully separate how data is transported through the various systems (**Transport**) and the representation domain of the data exchanged (**Payload**) making it extremely modular: over time additional services can be added applying the same communication bearer. In this way an incremental approach can be applied where only the subset of services actually required must be implemented in order to manage a public transport system. The expectation is that users may start with just one or two services and over time increase the number of services and the range for supported options.

The main services available in SIRI are:

- **Production Timetable Service**: it exchange information about the operations expected in a specified day in the near future;
- **Estimated Timetable Service**: it provides details of the operation of the transport network for a period within the current day, detailing real time deviations from the timetables and control actions affecting the Timetable (cancellations, additional Journeys and Detours);
- **Stop Services**: this service provide a stop-centric information about the current vehicle arrivals at monitored stops;
- **Vehicle Monitoring Service**: it provides information about of the current location and expected activities of a set of vehicles monitored, and it gives the journey planned for each vehicle of the fleet, together with the scheduled and expected arrival times;
- **Connection Protection Services**: It manages the information about the Connection Points that represents the point of contact between two different vehicles. It is extremely important to monitor the *Guaranteed Interchange* points;
- **General Messaging Service**: It allows to exchange informative messages between participants (travel news or operational advice).

The communication layer used by SIRI is formed by a set of general communication protocols in order to exchange information between client and server. The main communication patterns used by SIRI are:

- **Request/Response**: it allows for the ad hoc exchange of data on demand from the client;

- **Publich/Subscribe**: allows for the repeated asynchronous push of notifications and data to distribute events and Situations detected by a Real-time Service;

Transport Direct (TD)

Transport Direct (TD) was originally founded in 2004 as a project developed by a special division of the UK Department for Transport (DfT) in order to define new standards, data and better information technology systems to support public transport services. This project originated a new web portal, called *Transport Direct Portal*, which represented a public multi-modal journey planner. After an accurate review made over 200 reports [102], started in 1995 and ended in 2003, it was decided that the topic areas on which this project must focus its attention were the following:

- consumer demand for information;
- information requirements of the end user;
- embracing walk, cycle and car information;
- the importance of awareness and marketing;
- the importance of awareness and marketing;
- effects of information on behaviour;
- willingness to pay for information;
- the importance of partnership and buy-in;
- making the business case;
- media and presentation formats;
- feasibility of including retailing with information;
- technical standards and technological solutions;
- integration of real time systems into travel information systems;
- interpreting integration and distinguishing it from coordination.

This study highlighted the key-points that must be integrated in TD. To reach the prefixed goals, a number of data standards were developed. These standard supported the collection, transfer and management of the required transport data and to satisfy the topic areas shown before. Some of the developed (and integrated) standard were:

- **CycleNetXChange**: a data protocol for exchanging information about infrastructure to support the development of a national cycle journey planning function within the Transport Direct Portal;
- **IFOPT**, a CEN standard for defining public transport access information;
- **JourneyWeb**: a protocol to allow the development of a distributed journey planning service;
- **NaPTAN**: for the exchange of information associated with bus stops, railway station and other public transport access point.

However this project was closed by UK government in September 2014 due to the suspension of Community funding.

Transmodel Data Model (TDM)

The Transmodel Data Model (TDM) is the European reference data model for PTSs and provides an abstract model and data structures for common concepts around transport, with the aim to drive the development of public transport information systems. Currently, it is widely adopted by enterprises and it constitutes the reference model for most European projects on PTSs. Its development started on 1989 as a part of the Cassiope project while further developments were achieved within the EuroBus project and the Cartridge and Harpist working groups,resulting in the Transmodel V4.1 ENV 12896, published in 1997. The TDM development continued until 2006, when the version 5.1 was formally adopted by CEN as the European standard EN12896. In the following this last version will be taken as reference.

Transmodel includes a comprehensive conceptual model covering 14 subdomains[1](hereafter referred to as *packages*) related to the public transport sector. The main packages adopted in the Transmodel DataModel standard are:

- *Network Description*: this package describes the network and the principal concepts related to them. These concepts, for example, are: routes, lines, journey patterns, stop points, route points and so on;
- *Versions, Validity and Layers*: it has the information related to data versioning. It also stores all the changes that occur over time on the data;
- *Tactical Planning Components*: in this package are considered all the information needed to define vehicle journeys (*Vehicle Scheduling*) and

[1]http://sitp.transmodel.org/transmodel_v5_en/pages/91af92133d890002.htm

drivers' duties (*Driver Scheduling*). After the definition of the journeys and works, they will be logically combined in order to create the full service necessary to cover the needs of passengers (*Rostering*);

- *Personnel Disposition*: the package is related to information on *physical drivers* like name, surname, date of birth and so on. It also provides support to assign the physical to logical drivers and associate them to a duty previously defined;

- *Operations Monitoring and Control*: this package describes the real time monitoring by using technologies like AVL (Automated Vehicle Location) or APC (Automated Person Counter). The monitoring operation consists in a frequent detection of the different resources scattered over the service network (i.e. Busses, Physical Drivers, etc...);

- *Passenger Information*: represents the information related to the service, which can be used to aid passengers about the status of the service itself;

- *Fare Collection*: in this package the main abstract concepts concerning the core of the fare system used in public transport are considered (i.e. Types of fares, fare rules, etc..). In addition, information related to customers and customer type are represented;

- *Management Information*: the goal these modules is to give some statistical information in order to support strategic decisions;

- *Multi-modal Operations*: this package manages the information related to the cooperation of the different public transport modes, i.e.: train, bus and ship;

- *Multiple Operators Environment*: it concerns multiple operators working in the same geographical area. This allows to solve problems regarding the management of resources and services offered by multiple operators in the same area.

Finally, packages are split in parts, hereafter called sub-packages, including classes and dealing with specific topics, e.g. $DetectionAndMonitoring$ is a subpackage included in the package $OperationsMonitoringAndControl$. Figure 2.1 shows a small fragment of four sub-packages with some classes included and mutual associations. Although in the following only a subset of Transmodel packages will be taken as reference model, the approach is general and is meant to take into account the full set of them.

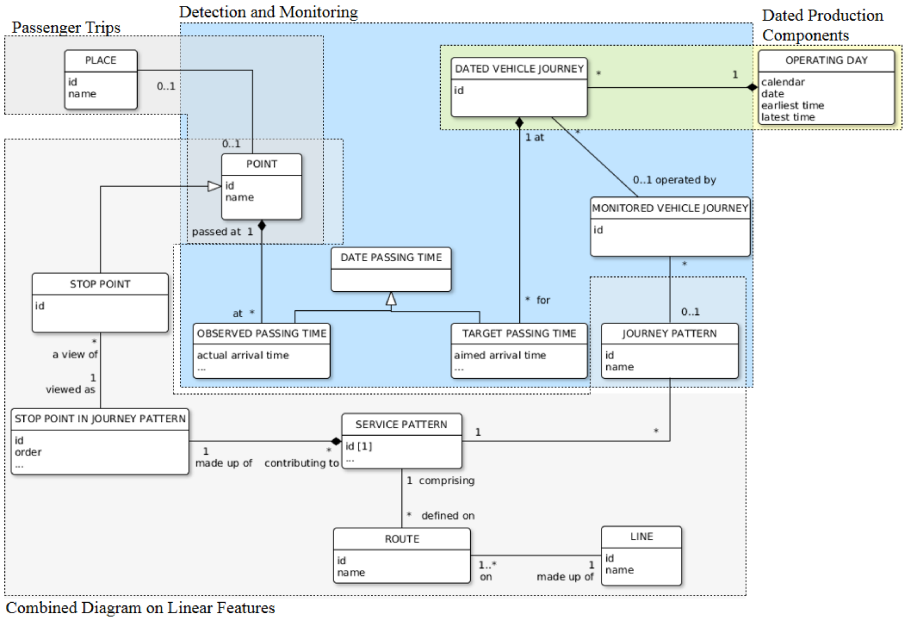


Figure 2.1: UML diagram with a fragment of four Transmodel subpackages: subpackage *Passenger Trips* from package Passenger Information, subpackage *Detection and Monitoring* and subpackage *Dated Production Components* from package Operations Monitoring and Control, and subpackage *Combined Diagram on Linear Features* from package Network Description.

2.3 The Knowledge Model

The approach proposed relies on a knowledge base which integrates different typologies of knowledge that is relevant for the domain of PTS. In more detail, next subsection provides an overview of KPIOnto, an ontology for the representation of indicators and their properties, including their formulas. Both Transmodel and KPIOnto are implemented as OWL ontologies, hence they include a set of classes and properties through which the domain can be represented in a formal fashion. Major benefits of such a representation include its machine-understandability and the possibility to interpret and process information by means of logic frameworks like that proposed in next section. In detail, it's OWL24 the reference language, representing the current and revised version of OWL. Finally, in Section 3.3 the two ontologies are properly interlinked to an integrated model.

2.3.1 KPIOnto

KPIOnto [99] is an ontology devoted to formally represent indicators and their formulas, that was developed starting from the analysis of existing dictionaries of performance indicators (e.g., VRC, SCOR) and by referring to the multidimensional model adopted for data management in data warehousing. So far, KPIOnto has been used in a variety of applications, ranging from performance monitoring in the context of collaborative organizations [99], to serving as a knowledge model to support ontology-based data exploration of indicators [103] or the development of ambient assisted living environments [104]. In general, KPIOnto can serve as a shared vocabulary for the definition of (a library of) KPIs for various domains. As such, through integration with other ontologies or extensions, it can be used to give meaning to monitored observations of KPI values, according to the Linked Data approach.

The core[2] of the ontology is composed by a set of primitive disjoint classes, as reported in Figure 2.2, where datatype properties are shown as attributes of the corresponding classes:

- `Indicator`, that represents a quantitative metric (or measure) used to evaluate the performance of an activity or a process. Properties of an indicator include an acronym, a description (i.e., a plain text giving a detailed description of its meaning and usage), one or more compatible dimensions (∃hasDimension.`Dimension`), a formula (∀hasFormula.`Formula`), the unit of measurement, a business objective (∃hasBusObj.`BusinessObjectives`)

[2] interested readers are encouraged to consult the full ontology specification, that is available online at *http://w3id.org/kpionto*.

and an aggregation function (∃hasAggrFunction.`AggregationFunction`). Example of indicators for PTSs include those detailed in Table 2.3.

- `BusinessObjective`, describing the optimization goal for which the indicator is used. In the field of transport systems, the following are among the most relevant: *Reliability*, that is the evaluation of the quality of service related to a PTS (related KPIs are e.g., "Scheduled times", "Effective times", "Dwell times"[105, 106]), *Sell Analysis, Validations & Controls*, i.e. the evaluation of ticket sales performances and revenues obtained by the company (related KPIs can be used to evaluate the trends in ticket selling, e.g. "Number of travel documents sold", "Time per passenger/ride" and "Penalties applied").

- `Dimension`: the coordinate/perspective along which an indicator is measured. Following the multidimensional model, a dimension is usually structured into a hierarchy of levels, i.e. instances of class `Level`. Each level represents a different way of grouping elements of the same dimension [107] (∃inDimension.`Dimension`). As an example, in Table 2.1 are reported some dimensions and corresponding levels for the indicators in the case study of Section 2.5. Hence, the indicator *"DelayAdvancePassingTime"* can be measured through the specific Day (TimeDimension) and the City (StopsDimension). A level is instantiated in a set of elements known as `Member`s of the level (∃inLevel.`Level`), e.g. "2013-01" for level "Month".

Analysis Dimension	**Dimensional Levels**
TimeDimension	Year, Quarter, Month, Week, Day
ServiceDimension	Vector, Line, Route, Ride
VehicleDimension	VehicleType, VehicleClass, Vehicle
StopsDimension	Nation, Region, City, Location, Stop

Table 2.1: The list of analysis dimensions and their relative levels for the case study.

- `Formula`: it allows to make the computational semantics of an indicator explicit, i.e. to formally represent an indicator as a function of other indicators. An indicator can indeed be either atomic or compound, built by combining several other indicators. Dependencies of a compound indicator *ind* on its building elements are defined by means of a mathematical expression $f(ind_1, \ldots, ind_n)$, i.e. a *Formula* capable to express how the indicator is computed in terms of $\{ind_1, \ldots, ind_n\}$, which are in turn (formulas of) indicators [99].

As shown in Figure 2.2, in KPIOnto the mathematical expression of a for-

mula is represented through a set of classes and relations capable to codify its operators and operands. About the former, operators are represented as URIs and externally defined by OpenMath [108] (∃hasFunction.`xsd:anyURI`), an extensible XML-based standard for representing the semantics of mathematical objects, which includes a wide set of operators in Content Dictionaries (CD), i.e. collections of symbols and their definition expressing their meaning. Different CDs are available in the standard OpenMath for various subsets of mathematics, including linear algebra, polynomials and group theory, transcendental functions, combinatorics and many other, although new CDs can be defined at need. As for operands, in the ontology they are defined as `FormulaArguments`, i.e. objects of mathematical operators, which in turn can refer to another `Indicator`, a `Constant` (either an integer or a float) or another `Formula`.

In order to maintain self-consistency of the ontology, any formula defined for an indicator must be coherent with all the others, according to the following definition.

Definition 1. *(Consistency of the knowledge base).* Given a new indicator, its formula is consistent with the other previously defined formulas if (a) it is not mathematically equivalent to any already defined formula and (b) it does not contradict any other already defined formula.

Two formulas are considered equivalent if one can be rewritten as the other through mathematical transformations (e.g., application of commutative or distributive property, or equation solving). This check is useful to individuate and manage duplications and redundancies. Various policies can be implemented when equivalent formulas are identified, e.g. merging the duplicates leaving only one definition or allowing multiple definitions, hence explicitly representing possible alternatives. Here is assumed that equivalent formulas correspond to identical indicators. Hence, if a new indicator is found to be equivalent to another, a reconciliation between them is required, to minimize redundancy in the knowledge base.

Please note that the evaluation of consistency according to the definition is implemented by specific functionalities of the reasoning framework as discussed in Subsection 2.4.4, that rely on the explicit representation of formulas shown here.

2.3.2 TransmodelOnto

Given that in the considered approach the framework is meant as a knowledge base on which a set of reasoning functionalities are defined, the model itself needs to be defined through a machine-understandable, logic-based representation. For such a reason the main concepts of the Transmodel are represented through an ontological language. Besides the availability of reference ontologies

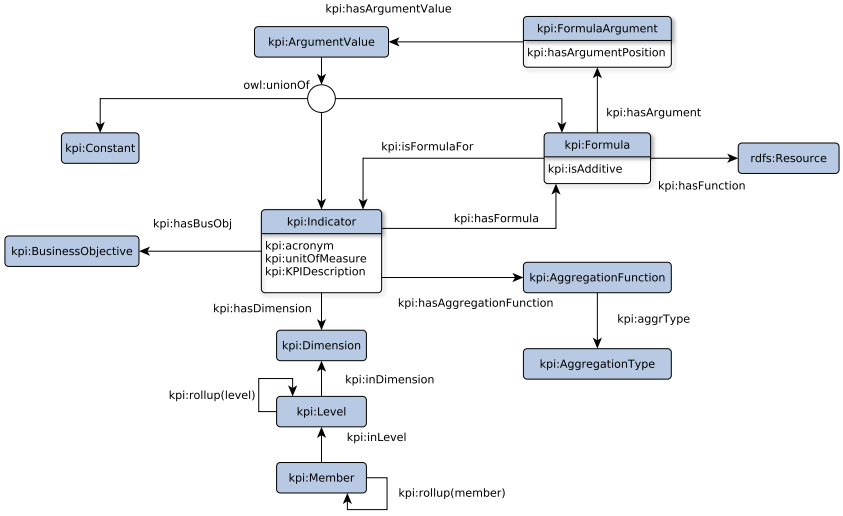


Figure 2.2: KPIOnto: an ontology for the representation of indicators and their properties.

aimed at the formal description of the fundamental concepts in the software domain, existing solutions (e.g., the Core Software Ontology (CSO) and the Core ontology of Software Components (COSC) [91]), are more focused on describing with the finest details the architecture of specific implementations more than conceptual schemas. An ontological representation of the Transmodel has not however been developed yet. Given the need for a formal representation of the relations among packages and classes, in this work a minimal ontology is considered, a meta-model for the representation of the main concepts related to the Transmodel architecture, describing the relations of inclusion among Transmodel classes and (sub)packages, and functional relation between classes:

- `TPackage`, which corresponds to a Transmodel package, as defined above. Subpackages are here represented as packages themselves (∃subPackageOf.`TPackage`).
- `TClass`, corresponding to a Transmodel class that is contained in one or more subpackages (∃inPackage.`TPackage`), belonging to the same or different packages. A class can be in a functional relation (i.e., with maximum cardinality 1) with another class (∃dependentOn.`TClass`); this relation is transitive, i.e. $\forall a, b, c$ dependentOn(a,b) $\wedge$ dependentOn(b,c) $\rightarrow$ dependentOn(a,c).
- `TBasicData`, which represents a basic information that can be used to

calculate an indicator. An instance of TBasicData is included in a class (∃inClass.`TClass`) as an attribute.

In Figure 2.2 are represented these classes in rectangles. As an example, in Figure are also shown in ellipses some instances taken from packages *"Passenger Information"* and *"Operations Monitoring and Control"* (see also Figure 2.1). The prefix *tmo* has been used to refer to the namespace of this ontology. The representation of more specific relations among Transmodel classes will be investigated by future work.

2.3.3 Linking Transmodel and KPIOnto

With the aim to integrate the KPIOnto ontology with the ontological model for Transmodel discussed in Subsection 2.3.2, corresponding concepts must be put in connection with each other through explicit links. Within the techniques dealing with semantic integration in the literature, this operation is knows as interlinking [109]. The mapping is performed by defining a `owl:sameAs` property between the couple of instances that are linked, manually or through semi-automatic approaches (the interested reader is referred to the above-mentioned publication).

In this work, KPIOnto indicators and levels are connected to corresponding Transmodel's basic data. To make an example, the following code in Turtle syntax[3] represents the mappings between two KPIOnto atomic indicators and basic data previously shown in Figure 2.2 , and between a KPIOnto level (Day_level from the TimeDimension) and a basic data. Prefixes *tmo* and *kpi* represent namespaces of Transmodel and KPIOnto ontologies respectively.

```
:AimedArrivalTime rdf:type tmo:TBasicData;
tmo:inClass :TargetPassingTime.
:ScheduledPassingTime rdf:type kpi:Indicator;
owl:sameAs :AimedArrivalTime.

:TargetPassingTime tmo:dependentOn :DatedVehicleJourney.
:DatedVehicleJourney tmo:dependentOn :OperatingDay.

:Day rdf:type tmo:TBasicData;
tmo:inClass :OperatingDay.
:Day_level rdf:type kpi:Level;
kpi:inDimension :TimeDimension;
owl:sameAs :Day.
```

As shown in the next section, in such a way it is possible to define reasoning functionalities capable to work over an integrated knowledge model.

[3] https://www.w3.org/TR/turtle/

As long as the dimensional schema is stable in terms of dimensions and their hierarchies of levels, the mapping between levels and basic data can be done once and does not require further changes. As a consequence, at any time a new KPI is introduced, only a mapping between such an indicator and the corresponding basic data is needed. In practice, this is needed only for atomic indicators, while compound indicators usually cannot be directly mapped to basic data, as they do not have counterparts in the Transmodel. In case of changing schemas[4], only the mappings that still reflect the new schema are kept, while the others are discarded and new mappings must be redefined. Mappings for atomic indicators must satisfy a number of conditions in order to be fully coherent with the rest of the links. Indeed, in KPIOnto every indicator has a set of compatible dimensions, each with a hierarchy of levels that are linked to basic data. In practice, for monitoring an indicator, i.e. a specific Transmodel basic data in a certain class, however it must be referred to levels that are measurable for the indicator at hand. In practical terms, this means that, starting from the basic data for an indicator, it is possible to functionally determine the value for its levels. This holds if a N:1 relation exists between their classes. In other terms, for a certain indicator can be considered as compatible levels only those that are in a functional relation with it in the Transmodel package. The following definition introduces the notion of measurable levels for an indicator.

Definition 2. *(Measurable levels for an indicator).* Given an indicator *ind* linked to a basic data in a class c, and given a set of compatible dimensions D, whose levels are linked to a set of basic data in classes $\{c_1, \ldots, c_n\}$, the measurable levels for *ind* are only those levels that are linked to basic data in classes $\{c_i : \text{dependentOn}(c, c_i)$[5]$\}$.

According to this definition, an indicator can be measured only along a subset of the levels of its compatible dimensions: in particular, those that belong to classes that are functionally related. To make an example, from the code above, indicator ***Scheduled Passing Time*** is linked to basic data `aimedArrivalTime` belonging to class `TargetPassingTime`, and this last has a (indirect) functional relation with `OperatingDay` class. In turn, level "Day" of the TimeDimension is linked to a basic data in `OperatingDay`. Hence, this level is measurable for the indicator at hand.

Stemming from this definition, the notion of coherent interlinking, as a mapping such that an indicator has at least one measurable level for each of its compatible dimensions is introduced.

[4]For instance, when a new version of the Transmodel is published.

[5]The property dependentOn(c_i, c_j) represents a functional relation between class c_i and class c_j of the Transmodel (see Subsection 2.3.2).

Definition 3. *(Coherent interlinking: atomic indicators).* Given an atomic indicator *ind*, the mapping to a basic data *bd* belonging to a class *c* is a coherent interlinking if, for each compatible dimension, the set of measurable levels for *ind* is not empty.

This property is checked when a new indicator is defined. In case the new indicator is compound, the following definition holds.

Definition 4. *(Coherent interlinking: compound indicators).* Given a compound indicator *ind*, there is a coherent interlinking if, after rewriting its formula in terms of atomic indicators $\{ind_1, \ldots, ind_n\}$, each component ind_i is coherently interlinked and if, for each compatible dimension, the intersection of the sets of measurable levels is not empty, i.e. there are some common measurable levels among all the components.

2.3.4 Discussion and evaluation

Various approaches have been proposed in the literature for ontology evaluation, targeting a number of different criteria (see [110] for a reference). In this work, the ontology *verification* is referred through the classic set of requirements that a formal ontology should satisfy, as proposed by various authors (e.g. among others [111, 112]):

- coherence, as the ontology must be non-contradictory. This property was checked through HermiT v. 1.3.8.413, which is an OWL2 reasoner fully compliant with OWL2 Direct Semantics and is built-in Protégé 5.2.0.

- Accuracy, as the ontology should correctly represent the relevant aspects of the domain at hand. The core of the ontology has been developed together with business experts in the context of the FP7 European project BIVEE (see also http://www.bivee.eu), hence it reflects their knowledge and needs in terms of performance metrics, that are overall similar to those in other domains.

- Minimal ontological commitment, i.e. the definition of only those terms needed to support the intended knowledge sharing. This goal was addressed considering a domain-independent vocabulary and a small number of basic classes and properties. *Encoding bias* is also considered minimised, i.e. avoiding representation choices for convenience of notation or implementation, by firstly developing a logic-based definition of the ontology and implementing it later on a specific language.

- Extensibility, that is the capability to be easily extended by other ontologies; KPIOnto has been designed to be used within a larger knowledge

repository, with the capability to be interoperable with other ontologies and data repositories, that were semantically aligned with KPIOnto definitions.

Finally, as for *accessibility*, KPIOnto specifications are available at [6], together with links to various formats for download. According to the structural ontology metrics calculated by Protégé 5.2.0, KPIOnto includes 200 axioms, 13 classes, 11 object properties, 7 data properties, with an overall DL expressivity of $\mathcal{ALUI}(\mathcal{D})$.

2.4 The Reasoning System

The model introduced in the previous section enables a formal representation of the knowledge related to KPIs for monitoring transport systems and their relations with relevant packages of Transmodel. The model is implemented as an OWL2 RL ontology [7]. This OWL2 profile is specifically aimed at applications requiring scalable reasoning without sacrificing too much expressive power. This is achieved by defining a syntactic subset of OWL2 which is amenable to implementation using rule-based technologies. The ontology can be queried by means of the SPARQL language[8] to extract relevant information. For instance, through simple queries it is possible to obtain the list of KPIs to monitor starting from a general *business objective* to achieve, which classes are dependent to a given *class* or to retrieve the set of subpackages that are needed given a certain *indicator*:

```
SELECT ?ind
WHERE {?ind a kpi:Indicator.
?ind kpi:hasBusObj <businessObjective>.}

SELECT ?class
WHERE {?class tmo:dependentOn <class>.}

SELECT ?basicData ?class ?package
WHERE {<indicator> owl:sameAs ?basicData.
?basicData a tmo:BasicData.
?basicData tmo:inClass ?class.
?class tmo:dependentOn ?class2.
?class2 tmo:inPackage ?package.}
```

However, non-explicit knowledge can not be directly accessed through queries. For example, whether a certain indicator depends on another one or not, or if

[6] http://w3id.org/kpionto
[7] https://www.w3.org/TR/owl2-profiles/#OWL_2_RL
[8] https://www.w3.org/TR/rdf-sparql-query/

a given package includes the basic data needed to monitor a set of indicators. Similarly, understanding which is the minimal set of subpackages to consider for monitoring a set of indicators is not a trivial task and would require a considerable effort, in terms of execution of a number of queries and complex elaboration of their results.

For these reasons, a reasoning framework that takes advantage of the knowledge model to provide developers with a set of basic reasoning services has been developed. These services are then exploited to realise more complex functionalities as shown in Section 2.5. The framework relies on Logic Programming (LP) as a common logic layer capable to provide a unified view over (and reason on) the two main different sources of knowledge involved in this scenario, namely ontological knowledge about Transmodel and indicators, and the mathematical knowledge related to how to manipulate KPI's formulas according to sound algebraic operations.

In the following is shown how the knowledge model is represented within the LP framework. Given that the Prolog theory includes more than 950 predicates, hereafter are introduced only the main reasoning services. The code of the whole Prolog framework is publicly available at [9].

2.4.1 Knowledge representation in Prolog

A part of the ontological knowledge, namely the fragment that is needed to support specific reasoning functionalities, has been translated in Prolog. It includes a set of facts and Logic Programming predicates capable to perform basic reasoning tasks. At first, the framework translates class membership axioms related to classes Indicator, BusinessObjective, TBasicData, TClass and TPackage as Prolog unary predicates of type `Indicator('ActualPassingTime')`, `BusinessObjective('Reliability)` and so forth. Then, indicator formulas, available in the ontology as a set of instances and relations, are converted into facts, e.g. `formula('DelayAdvancePassingTime', 'ActualPassingTime' -'ScheduledPassingTime')`. Finally, some OWL ObjectProperties are included as facts in the LP knowledge base:

- `hasBusObj(indicator,business_objective)`, between a KPIOnto indicator and a business objective.

- `sameAs(indicator,tBasicData)` or `sameAs(member,tBasicData)`, for the owl:sameAs relation between a KPIOnto indicator/member and a Transmodel basic data, e.g. sameAs('ScheduledPassingTime', 'AimedArrivalTime').

[9] https://github.com/KDMG/PRESS4KPI

- `inClass(tBasicData,tClass)`, between a Transmodel basic data and its class.

- `dependentOn(tClass1,tClass2)`, between two classes, where the former is connected to the latter with a functional relation; given that this property is defined as transitive (see Subsection 2.3.2), the reasoner performs a transitive closure for the full generation of these facts (see also the Prolog definition in Subsection 4.3), e.g. dependentOn('ObservedPassingTime', 'Point') and dependentOn('Point','Place') implies dependentOn('ObservedPassingTime', 'Place') (see diagram in Figure 2.1).

- `inPackage(tClass,tPackage)`, between a Transmodel class and its package.

2.4.2 Services for mathematical manipulation of formulas

As for predicates, reasoning about measures is mainly based on the capability to manipulate formulas according to strict mathematical axioms, like commutativity, associativity and distributivity of binary operators, and properties of equality needed to solve equations. To this purpose, the framework includes at its core a library of predicates for the manipulation of mathematical expressions, called PRESS (PRolog Equation Solving System) [113]. This is a formalisation of algebra in Logic Programming for solving symbolic, transcendental and non-differential equations. Its code can be represented as axioms of a first-order mathematical theory and the running of the program can be regarded as inference in such a theory.

The predicates in which it is organised are mainly aimed to enable manipulation of mathematical formulas and resolution of equations. The first ones implement an essential reasoning functionality that consists in deriving relations among indicators, manipulating a formula to achieve a specific syntactic effect (e.g. to reduce the occurrences of a given variable in an equation) and rewriting a formula accordingly. The second type enables the symbolic resolution of equations by applying mathematical properties (e.g., commutativity, factorisation) and properties of equality. For instance, the equation $A = \frac{(B*C+B*D)}{B}$ can be rewritten by factorisation of B as $A = \frac{B*(C+D)}{B}$ and then as $A = C + D$. Finally, it can be solved with respect to C, with solution $A - D$. The number and kinds of manipulations the reasoner is able to perform depend on the mathematical axioms described by means of logical predicates. XSB[10] was chosen as LP database system for its efficiency.

[10] http://xsb.sourceforge.net/

2.4.3 Services for dependency analysis

On the top of the core library of math functions, other predicates are defined to provide some basic reasoning services for dependency analysis. Central to all these functions is the notion of "common measures", where measure is taken here as synonym of indicator: given a set of measures $\phi = \{M_1, M_2, ..., M_n\}$, common measures of ϕ is the minimal set of measures needed to compute all formulas of ϕ. This concept has been implemented through the following predicates:

- `indToMea(`L_i`,`L_m`)`, which takes as input a list L_i of indicators and generates its common measure set in L_m. For instance, let us consider the formulas in Table 2.3.
  If $\phi = \{DelayAdvancePassingTime, AverageDelayAdvance\}$, then a solution for `indToMea(`ϕ`,`L_{out}`)` is $L_{out} = \{NumberOfJourneys, ActualPassingTime, ScheduledPassingTime\}$.
  In fact, with $ActualPassingTime$ and $ScheduledPassingTime$ it is possible to calculate $DelayAdvancePassingTime$.
  Then, with $NumberOfJourneys$, $AverageDelayAdvance$ can be calculated.

- `meaToInd(`L_m`,`L_i`)` which implements the inverse of indToMea: given a set of available measures L_m, the predicate returns in L_i all those indicators that are derivable from them. Formula manipulation and formula rewriting functionalities are exploited here to expand the set of computable measures. To make an example, if $L_{in} = \{ActualPassingTime, AverageDelayAdvance, DelayAdvancePassingTime\}$ it is possible to derive $L_{out} = \{ScheduledPassingTime, NumberOfJourneys\}$. Indeed, by having $ActualPassingTime$ and reverting the formula for $DelayAdvancePassingTime$ it is possible to obtain $ScheduledPassingTime$. Finally, by reverting $AverageDelayAdvance$ it is possible to derive the $NumberOfJourneys$.

On the basis of these definitions, a further set of Prolog predicates have been implemented:

- `indToBasicData(`L_i`,`L_d`)`, given a set L_i of indicators, returns the sets L_d of basic data needed to compute them. This relies on determining from `indToMea` the minimum set M of measures actually needed, and then on the execution of a Prolog goal in the form `sameAs(m,X)` for each measure `m`$\in M$. Basic predicates for formula manipulation are exploited to derive alternative ways to compute a measure. The result of the predicate will include all possible combinations of basic data capable to overall satisfy the request, that is a set of sets of basic data:

```
indToBasicData(Li,Ld):-
indToMea(Li,Lm),
indToBasicData1(Lm,Ld).

indToBasicData1([[M|R]|Lm],[B|Lb]):-
decomp([M|R],B),
indToBasicData1(Lm,Lb).
indToBasicData1([],[]):-!.

decomp([M|R],[B|Lb]):-
sameAs(M,B),
decomp(R,Lb).
decomp([],[]):-!.
```

- `basicDataToInd(`L_d`,`L_m`)`, given a set L_d of basic data, returns the set L_m of measures that can be calculated: at first the list of available measures is retrieved by exploiting the goal `sameAs(X,d)` for each basic data `d` $\in L_d$. Afterwards, predicate `meaToInd` is executed to expand the set of computable measures obtained in the first step.

- `basicDataToSubpackages(`L_d`,`L_s`)`, given a set L_d of basic data, it generates all possible alternative sets L_s of subpackages including (`inPackage`) those basic data. For instance, if L_s includes id in MonitoredVehicleJourney and id in DatedVehicleJourney, as shown in Figure 2.3, the output L_s includes many alternative solutions, among which $\{Events\}$, $\{DetectionAndMonitoring\}$ or $\{RecordedUseOfServices\}$.
  Each of these solutions are capable to provide the full set of basic data in L_d:

  ```
  basicDataToSubPackages([D|Ld],[P|Ls]):-
  inPackage(D,P),
  basicDataToSubPackages(Ld,Ls).
  basicDataToSubPackages([],[]):-!.
  ```

  Conversely, given a set L_s of subpackages, the predicate returns in L_i the possible basic data that are available, by exploiting the same approach.

- `getMeasurableLevels(`I`,`L`)`, given an indicator I, with a corresponding set of compatible dimensions, returns the set L of levels that are measurable according to Definition 2. This implies to retrieve the basic data corresponding to indicators in I, and then to verify whether their corresponding classes are `dependentOn` or not with classes linked to dimension levels:

  ```
  getMeasurableLevels(I,L):-
  compatibleDim(I,D),
  ```

```
inDimension(L,D),
sameAs(L,Bl),
inClass(Bl,Cl),
sameAs(I,Bi),
inClass(Bi,Ci),
isDependentOn(Ci,Cl).

isDependentOn(X,Y):- dependentOn(X,Y).
isDependentOn(X,Z):- dependentOn(X,Y), isDependentOn(Y,Z).
```

A further service, useful to develop more advanced applications, is `get_formulas(ind,`L_{out}`)`, which returns all possible alternative formulas for a given indicator. For instance, for indicator *AverageDelayAdvance* defined in Table 2.3 , it will return formula $\frac{DelayAdvancePassingTime}{NumberOfJourneys}$ but also formula $\frac{(ActualPassingTime-ScheduledPassingTime)}{NumberOfJourneys}$, because indicator *DelayAdvancePassingTime* can be replaced by its own formula. Even if the indicator is atomic and a formula is not provided, this service can calculate an answer by reverting other formulas, e.g. *ActualPassingTime=DelayAdvance-PassingTime+ScheduledPassingTime* by applying the mathematical service for equation solving.

2.4.4 Services for consistency management

According to the definition of consistency given in Subsection 2.3.1, a set of predicates are defined to check if an indicator is consistent (not equivalent and coherent) with the others previously defined. Some predicates are executed to support such a verification, namely `equivalence(I,Formula,`L_e`)` and `incoherence(I,Formula,`L_i`)`, which respectively return the list L_e and L_i of indicators whose formulas are equivalent or incoherent with the one at hand. The general formulation of these predicates are as follows, where `expand_equation` and `solve_equation` are PRESS predicates for formula manipulation and solution:

```
equivalence(X,Equation,L) :-
expand_equation(Equation,ExpandedEquation),
solve_equation(ExpandedEquation,X,X=Solution),
formula(L,S),
expand_equation(L=S,L=ES),
solve_equation(L=ES,L,L=Solution2),
Solution=Solution2.

incoherence(X,Equation,L) :-
expand_equation(Equation,ExpandedEquation),
solve_equation(ExpandedEquation,X,X=Solution),
```

```
formula(L,S),
expand_equation(L=S,L=ES),
solve_equation(L=ES,X,X=Solution2),
Solution \= Solution2,
tolist(Solution,LSol),
tolist(Solution2,LSol2),
\+ notin(LSol,LSol2).
```

Typically, these predicates are used before a new formula fact is added to a repository. To make an example, let us assume that all the indicators and formulas in Table 2.3 have been defined in the knowledge base and a new indicator *ind* is asked to be added, with formula *ind*=*DelayAdvancePassingTime* +- *ScheduledPassingTime*. The predicate `equivalence` would then recognize that the formula for *ind* is actually mathematically equivalent to {*ActualPassingTime*}, by reverting the formula for *DelayAdvancedPassingTime* originally put in the knowledge base (see the Table 2.3) On the other hand, if the new formula to add is $ActualPassingTime = \frac{ScheduledPassingTime}{DelayAdvancePassingTime}$, then the predicate `incoherence` would determine that it contradicts the formula for *ActualPassingTime* that can be derived by rewriting the formula for *DelayAdvancePassingTime*.

2.4.5 Discussion and evaluation

A set of experimental tests have been performed on the logical framework with the purpose to evaluate the efficiency in terms of running times. Hereafter the main results are discussed, which have been focused on the basic logic functions introduced in the previous subsections, that are involved in the provisioning of services described in Section 2.5.

In detail, running times for predicates `indToMea` and `meaToInd` have been tested, which are the most computationally complex and are used in most of other predicates. The input consists of synthetically generated knowledge bases of increasing size, containing a set of formulas. In order to determine a reasonable number of indicators for the tests, some KPI libraries including indicators for transportation (see also Section 2.2) have been analysed. For instance, the "Study on key performance indicators for intelligent transport systems", the final report in support of the implementation of the EU Legislative Framework on ITS [114], includes 228 KPIs covering different areas, the biggest thereof including 57 indicators. Even though some transit companies refer to large KPI libraries with more than 180 indicators, the TCRP Report 88 [115], produced by the Transport Research Board, recommends an average number of used measures around 20, which seems appropriate for most companies. Many other frameworks include a similar amount of indicators [61], even in other domains. For instance, KPILibrary includes at most a few hundred KPIs for

indToMea				
Level	num. KPIs	k=1	k=5	k=50
2	7	0.08	0.08	0.08
3	15	0.08	0.09	0.13
4	31	0.08	0.09	0.31
5	63	0.08	0.09	0.45
6	127	0.08	0.09	0.94
7	255	0.08	0.09	1.36

meaToInd				
Level	num. KPIs	k=5	k=50	k=250
2	7	0.05	0.07	0.12
3	15	0.09	0.17	0.42
4	31	0.09	0.17	0.62
5	63	0.09	0.20	1.22
6	127	0.09	0.23	1.57
7	255	0.09	0.24	2.50

Table 2.2: Execution times (in seconds) of predicate (a) indToMea and (b) meaToInd for ontologies of different sizes.

most topics, while in the European project BIVEE [99] around 350 KPIs are defined that cover production and innovation aspects. These libraries include KPIs that may have or not have a formula. Even in the latter case, only a subset of the indicators in a library are in connection each other. The term *graph of formulas* is used to indicate the set of indicators that are mutually linked through formulas. In general, several independent graphs of formulas co-exist in the same knowledge base.

The logic predicates performs manipulation within each (connected) graph, hence the existence of other graphs does not have any impact on the running times. By taking into account the provided numbers, an input ontologies which are connected graphs of formulas including up to 255 KPIs has been defined. The number of operands per formula has been fixed to 2, while formulas are generated as summation of two randomly chosen indicators. In the following with the term *level* is meant the number of layers in the graph of formulas, from which the number of KPIs is derived. In the tests, the maximum number of formulas that can be inferred at each iteration to a value specified has been bounded by a parameter k, to keep the complexity of the experimental procedure under control.

The predicate `indToMea` has been executed for every indicator in the input file and averaged at the end, with k from 1 (which corresponds to most concrete cases, where just one solution is enough) to 50. The predicate `meaToInd` has been executed by providing a list corresponding to the whole set of indicators in the knowledge base. In this way, the predicate searches for all possible ways

to calculate all the indicators. Parameter k goes from k=5 to k=250.

Results are shown in Table 2.2. Running times are below 1 second in most cases, whereas larger input files require up to 2.50 seconds. It is considered that in most concrete cases just one solution (or the first few ones) can be sufficient.

2.5 High Level Tasks Definition And Application

In this Section will be introduced a case study that will be used through this work to exemplify the developed approach, and that has been designed within a collaboration with PluService srl, a private company operating in Italy on information systems for public transportation. Consider a typical scenario where a public transport company, responsible for providing a multi-modal local service, wants to perform an accurate monitoring of the performances achieved during the course of daily operations. In particular, as a measure of quality, the company is interested in analysing *reliability* of services offered to customers. It is widely agreed that, in the public transport sector, reliability can be defined as the adherence of particular aspects of the transport service with those previously scheduled and how they are perceived by customers. Given this business objective, the company identified a set of Key Performance Indicators in order to monitor two main aspects of the service reliability:

- the punctuality of the performed rides at stop points, by considering the variability among the scheduled passage time and the actual passage time;
- the adherence of the paths followed by the performed rides with respect to the scheduled ones, by highlighting the differences among them.

The evaluation of the service punctuality is of particular interest especially for customers, who are principally concerned with the end result and are not interested in evaluating where the service has failed in operating or it is operating late [116]. Vice versa, from the operator's perspective, the identification of "weak points" in service operations are of utmost importance both to reduce the waste of resources and to increase the efficiency of the performed services. In order to run the described analysis, the operator considered the list of indicators presented in Table 2.3, where the description and the formula for their calculation, whenever applicable, are described.

2.5.1 Applications

The ontology-based framework described in this work is developed with the purpose to provide guidance to the design of a performance monitoring system, for the evaluation of performances achieved by a public transport system

Indicator	Description	Formula
Actual Passing Time	The actual passing of a public transport vehicle at a pre-defined POINT during a MONITORED VEHICLE JOURNEY	
Scheduled Passing Time	The scheduled passing time of a public transport vehicle at a pre-defined POINT on a particular DATED VEHICLE JOURNEY	
Delay/Advance Passing Time	The difference among the Scheduled Passing Time and the Actual Passing Time	$ActualPassingTime - ScheduledPassingTime$
Number of Journeys	The number of performed journeys during the service	
Standard Deviation Delay/Advance	The standard deviation among the Scheduled Passing Time and the Actual Passing Time	$\sqrt{\frac{Delay/AdvancePassingTime^2}{NumberOfJourneys}}$
Stops in Scheduled Journeys	Number of stops considered in scheduled services	
Scheduled Journeys	The number of journeys that are scheduled	
Executed Journeys	The number of journeys that are scheduled and have been effectively performed	
Rate of Journeys Scheduled Actually Performed	The rate of scheduled journeys effectively performed during the service	$\frac{ScheduledJourneys}{ExecutedJourneys}$

Table 2.3: List of KPIs adopted by the public transport company to evaluate service reliability.

based on the Transmodel data model. In this Section, how the approach can be exploited to support a number of potential real-world applications for the development and the setup of a Performance Monitoring System is addressed:

- definition of new KPIs: in case the operator is interested in introducing a new performance indicator, that is not already in the ontology, the framework provides a function to verify that such an indicator is compliant with a set of requirements;

- identification of the relevant KPIs to monitor, given a set of business objectives;

- identification of required packages from a given KPI: the framework returns the Transmodel packages that are needed in order to calculate the specified indicators;

- identification of evaluable KPIs from a a set of packages: given a set of available Transmodel packages, the framework returns the KPIs that can be evaluated from those. This scenario represents the dual case of the previous one.

The above-mentioned functions are discussed in detail in the following by referring to applicative scenarios from the case study presented in Section 2.5.

2.5.2 Definition of a new KPI

In case the public transport operator is interested to introduce a new KPI that is not available in the knowledge base, the following steps ought to be executed:

1. definition of the new KPI in terms of its properties: acronym, description, compatible dimensions, formula, unit of measurement, aggregation function;
2. check of mathematical consistency of the indicator formula (see Definition 1);
3. link to the corresponding basic data;
4. check of interlinking coherency (see Definitions 3 and 4).

Let us consider, for instance, the introduction of the last KPI in Table 2.3, namely the *RateOfJourneyScheduledActuallyPerformed*, that is used to monitor the rate of scheduled journeys that have effectively been performed during the service. The rest of the KPIs in the Table are assumed to be already defined in the system, as well as the mapping relationships (i.e., sameAs) among KPIs and basic data elements. Similarly, all dimensions and their levels are assumed to be already defined and mapped to corresponding basic data. See Figure 2.5 for an example about the *StopsDimension*. The indicator is detailed as follows:

- `kpi:acronym`: "RJSAP";
- `kpi:KPIDescription`: "The rate of journey scheduled that actually are effectively performed by drivers and vehicles";
- `kpi:unitOfMeasure`: float;
- `kpi:hasBusObj`: Reliability;
- `kpi:hasAggregationFunction`: avg;

According to step 2, the mathematical consistency of RJSAP formula must be checked with respect to others KPIs defined in the knowledge base. As discussed in subsection 2.4.4, this is performed by the execution of predicates `equivalence(RJSAP,` $\frac{ExecutedJourneys}{ScheduledJourneys} * 100$`,`$L_e$`)` and `incoherence(RJSAP,` $\frac{ExecutedJourneys}{ScheduledJourneys} * 100$`,`$L_i$`)`, with the aim to check if the indicator's formula is equivalent or incoherent with any other. In this case, the lists L_e and L_i returned by the two predicates are empty since no equivalence or inconsistency emerge with any other KPI.

As third step, needed links must be defined between the new indicator and corresponding basic data. As mentioned, is assumed that all other KPIs of

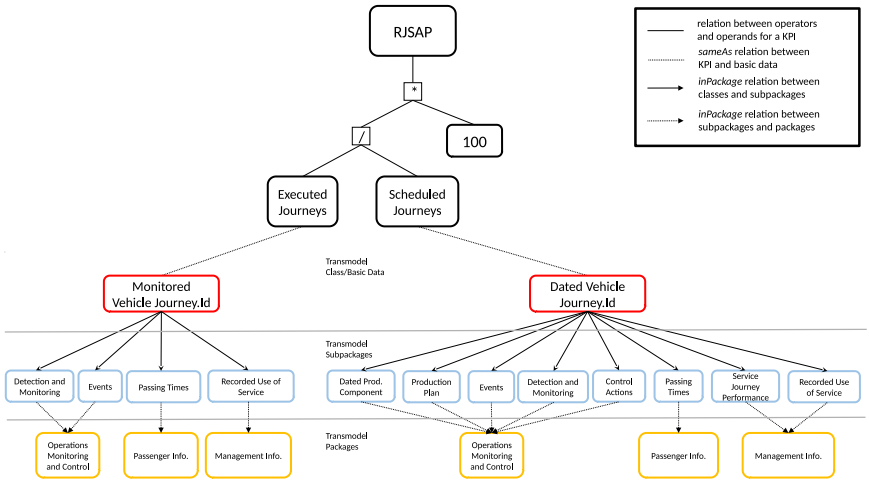


Figure 2.3: Decomposition of RJSAP mathematical formula and mappings with Transmodel Basic Data (in red), corresponding subpackages (in blue) and packages (in yellow).

the case study, including the operands used in the formula, have been already defined in the knowledge base. As such, indicators $ExecutedJourneys$ and $ScheduledJourneys$ are already mapped to corresponding Transmodel basic data, namely the attribute Id in class $MonitorVehicleJourney$ for the former and attribute Id in class $DatedVehicleJourney$ for the latter, as also shown in Figure 2.3[11]. Given that RJSAP indicator has no correspondance in any Transmodel class, no further link is needed in this case. Please note that this approach enables to define and calculate indicators, like RJSAP, that have not been originally considered by Transmodel. As a consequence, the user can freely extend its set of indicators to monitor by simply combining existing indicators into more complex, compound ones through mathematical formulas.

Finally, the framework checks the coherence of interlinking to verify that the indicator can be actually measured for each compatible dimension, at least for one level. According to the definition of coherent interlinking for compound indicators (Definition 4), the check is done by considering if its dependent indicators, in this case $ExecutedJourneys$ and $ScheduledJourneys$, are coherently interlinked. In the example, the two indicators are linked to basic data in classes $MonitoredVehicleJourney$ and $DatedVehicleJourney$. As shown in the UML diagram of Figure 2.1, as for the TimeDimension, the first class is indeed in a functional relation with the second in Transmodel, and this last

[11] Please note that this basic data do not provide the real value for executed journeys and scheduled journeys, but starting from them it is possible to calculate such indicators by aggregation.

with *OperatingDay*, which includes a basic data named *date* from which all the levels of the dimension can be derived. As for the ServiceDimension, the two classes are dependent on *JourneyPattern*, that is linked to class *Route* (and, in turn, is linked to *Line*) which includes a basic data for corresponding level *Route* (and *VectorLine*) of the dimension. Finally, as the framework verifies the coherent interlinking, the KPI can be added to the knowledge base.

2.5.3 Identification of relevant KPIs to monitor from a set of objectives

This task can be easily achieved by selecting those indicators that have a `kpi:hasBusObj` relation with some business objectives. Given a set $B = \{b_1, \ldots, b_n\}$ of business objectives, the set of KPIs to monitor is $\{ind_i : \exists b_j \in B, hasBusObj(ind_i, b_j)\}$.

With respect to the case study, if the operator is interested in *Reliability* as business objective, the set of indicators to monitor are those listed in Table 2.3.

2.5.4 Identification of required Transmodel packages needed to monitor a KPI

Given the complexity of the Transmodel data model, a support functionality is required to drive the operator in the identification of the specific (sub)packages that are needed to monitor a given indicator. This is of utmost importance during the usage of the monitoring system as a reference to precisely locate where a certain basic data can be retrieved, but also at design time, to determine which specific subpackages are needed for a certain set of business objectives to achieve. More formally, given a set I_t of target indicators to monitor, its corresponding set B_t of linked basic data are given by `indToBasicData(`$\mathtt{I}_t$`,`$\mathtt{B}_t$`)`. The set $L_s = \{s_1, \ldots, s_g\}$ of required Transmodel subpackages is determined as `basicDataToSubPackages(`$\mathtt{B}_t$`,`$\mathtt{L}_s$`)`. That is, the set L_s is determined by considering all needed subpackages for all the identified basic data. Finally, the user specifies which specific levels will be used for monitoring the I_t, among the list of the measurable ones (they can be retrieved through predicate `getMeasurableLevels`). A last check verifies

To make an example, suppose that the operator wants to perform an analysis of delays or advances occurring during the progress of the public transport service. To perform the above mentioned task, the operator chooses a set of KPIs and the set of target dimension levels on which the analysis is focused. As for the former, selected indicators are *ActualPassingTime*, *ScheduledPassingTime*, *DelayAdvancePassingTime* and

StandardDeviationDelayAdvance. For what concerns the target levels, among the dimensions that are compatible with all these indicators and within the set of levels that are measurable for all of them, the operators focuses only on the *StopsDimension*, and specifically to level *Location*.

Firstly, as already explained in paragraph 2.5.2, by taking advantage of the relationships already defined in the knowledge base, the system will decompose iteratively the mathematical formulas of the KPIs in order to obtain the minimal set of basic data required to evaluate the indicator at hand (see Figure 2.4). This is done by executing the predicate `indToBasicData` to the above mentioned list of KPIs, obtaining as output the minimal set of basic data needed to calculate all these indicators, as detailed in the following:

- at first, the reasoning service determines the minimal list of the needed indicators to monitor, by means of `indToMea`, namely $NumberOfJourneys$, $ActualPassingTime$ and $ScheduledPassingTime$: indeed StandardDeviationDelayAdvance depends on the first and on DelayAdvancePassingTime, but this last depends on the second and the third;

- secondly, the corresponding basic data for these indicators are found, namely $AimedArrivalTime$ in class TargetPassingTime, $ActualArrivingTime$ in class ObservedPassingTime and Id in class VehicleDetecting.

For each of these basic data, the corresponding subpackages and packages where they are defined are retrieved. Various alternative solutions may be available. As shown in Figure 2.4, both AimedArrivalTime and ActualArrivingTime are available in the same two subpackages, while NumberOfJourneys is available in only one subpackage. Hence, the following two alternative solutions are found: either the subpackage $\{DetectionAndMonitoring\}$ alone or the couple of subpackages $\{DetectionAndMonitoring, PassingTime\}$.

For what concerns basic data related to the target level *Location* from StopsDimension, as shown in Figure 2.5 it is available in 5 subpackages. However, the framework checks which subpackages contain the measurable level: subpackage *PassingTime* does not allow to measure the *Location* level, while subpackage $DetectionAndMonitoring$ enables to measure it. As a consequence, the minimal solution for monitoring the set of KPIs chosen by the operator involves to install the single package "Operations Monitoring and Control" (which includes the subpackage "Detection and Monitoring"), from which the needed basic data about indicators and levels can be retrieved.

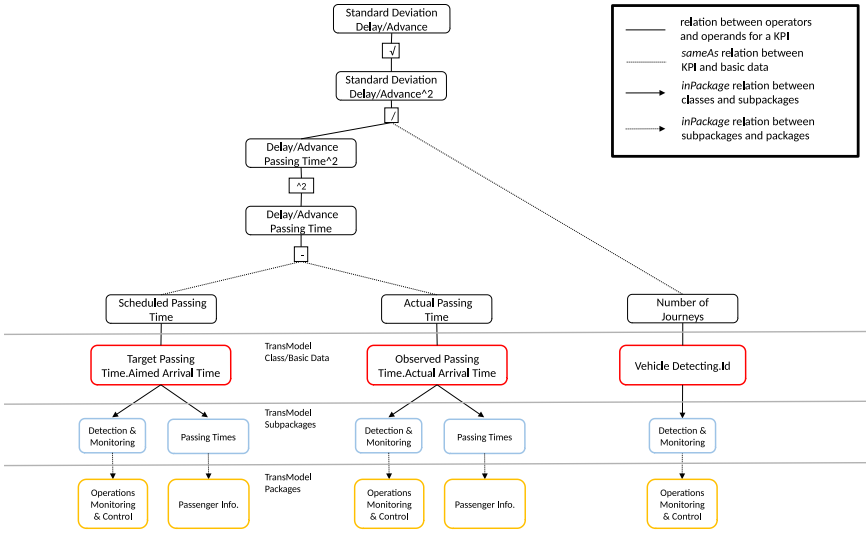


Figure 2.4: Decomposition of the formula for indicator StandardDeviationDelayAdvance and mappings with Transmodel Basic Data (in red), corresponding subpackages (in blue) and packages (in yellow).

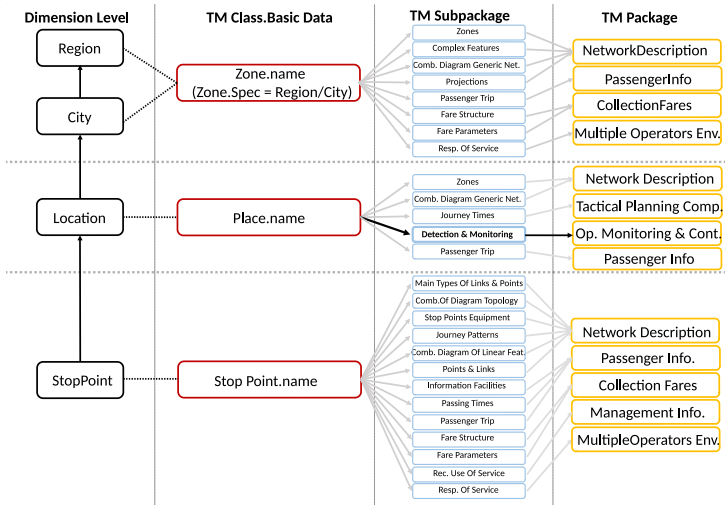


Figure 2.5: Mappings between levels of StopsDimension and Transmodel Basic Data (in red), corresponding subpackages (in blue) and packages (in yellow). In bold is highlighted the only available dimension level that is described in the example of subsection 2.5.4.

2.5.5 Identification of evaluable KPIs from a set of given packages

This scenario (inverse to the previous one) describes the situation in which the operator can use only a subset of the Transmodel packages. This is a typical scenario, as the whole model comprehensively accounts for many aspects related to the transport service, while operators may be interested only in a small part of them. By using the knowledge stored in the ontology and the reasoning services, the framework identifies which KPIs can be monitored from the available packages.

To give an example, let us suppose that the KPIs in Table 2.3 and the dimensions in Table 2.1 are already defined in the knowledge base, and let us consider a set of packages already deployed by the operator: "Passenger Information", "Fare Collection", "Network Description". For each package, by exploiting service `subpackagesToBasicData` it is possible to get to the list of all available classes and basic data, while in turn, executing `basicDataToInd` allows to get to the list of available KPIs. Given the considerable number of basic data and the lack of space, in Table 2.4 the relation between available packages (underlined) and all indicators in Table 2.3 are summarized. As shown, in this case all indicators of the case study are available or can be derived by means of mathematical manipulations, with the only exception of *Number of journeys* and *Standard Deviation Delay/Advance*. If the operator wants to compute also such indicators, the package *"Operations Monitoring and Control"* must be added.

As a further step, once the list of evaluable indicator is found, the need of determining which dimension levels can be used to measure them must be satisfied. Indeed, given that only 5 packages are available in the example, only some levels will be *measurable* in the sense of Definition 2. Consider the KPI *Stops in Scheduled Journeys* and the related dimensions. The KPI is calculated by obtaining the required data from "Network Description" or from "Passenger Information" packages (see Table 2.4). Accordingly, by considering the above mentioned packages, only the following dimensions levels are available for monitoring: *{Line, Route}* for *ServiceDimension* and *{Stop}* level for *StopsDimension* (see Figure 2.1).

KPI	**TDM Basic Data**	**TDM Packages**
Actual Passing Time	{ObservedPassingTime}	{OperationMonitoring-AndControl ∨ PassengerInfo}
Scheduled Passing Time	{TargetPassingTime}	{OperationsMonitoring-AndControl ∨ PassengerInfo}
Delay/Advance Passing Time(•)	{TargetPassingTime, ObservedPassingTime}	{OperationsMonitoringAnd-Control ∨ PassengerInfo}
Number of Journeys (∘)	{VehicleDetecting}	{OperationsMonitoringAnd-Control}
Standard Deviation Delay/Advance (∘)	{TargetPassingTime, ObservedPassingTime, VehicleDetecting}	{OperationsMonitoringAnd-Control}
Stops in Scheduled Journeys	{StopPointInJourney-Pattern}	{NetworkDescription ∨ MultipleOperationEnvironment ∨ PassengerInfo }
Scheduled Journeys	{DatedVehicleJourney}	{PassengerInfo ∨ ManagementInfo ∨ OperationsMonitoringAnd-Control}
Executed Journeys	{MonitoredVehicle-Journey}	{PassengerInfo ∨ ManagementInfo ∨ OperationsMonitoringAnd-Control}
Rate of Journey Scheduled Actually Performed (•)	{DatedVehicleJourney, MonitoredVehicleJourney}	{PassengerInfo ∨ ManagementInfo ∨ OperationsMonitoringAnd-Control}

Table 2.4: Case study: the relation between KPIs, corresponding basic data and packages needed for their calculation. The underlined packages are available in the analysed scenario. KPIs identified with (•) can be calculated only through reasoning functions, while those with (∘) cannot be computed. All the others are directly available.

Chapter 3

Predicting Travel Time at Bus Stop: Overview of Hybrid Models

3.1 Overview

In this chapter the Hybrid Travel Time prediction algorithms based on the cooperation of both Machine Learning and Kalman Filtering Models are introduced and compared with *"Simple Models"* formed by Machine Learning techniques. Travel time prediction has been an interesting research area for decades during which various prediction models have been developed. With the growth of the Advanced Travelers Information Systems (ATIS), short-term travel time prediction is becoming increasingly important. As the key input for dynamic RGS, travel time information enables the generation of the shortest path or alternative paths connecting the current locations and destinations, besides suggesting directions dynamically in case of congestions or incidents. In order to predict travel time in congested urban areas, initially, *Simple Models* composed only by a single algorithm were made but, being Travel time in urban areas an highly stochastic and time-dependant value due to random fluctuations in travel demands, interruptions caused by traffic control devices, incidents and bad weather conditions some more complex model structures became necessary. Moreover, the techniques adopted in literature are evaluated on a few data, with datasets and data pre-processing approaches different from article to article. The purpose of this chapter is to evaluate the effectiveness of *Hybrid Models* compared to *Simple Models* made by a single prediction algorithm, by applying an homogeneous and realistic experimental framework. The proposed *Hybrid Models* are divided into two different parts:

- **Machine Learning Model**: used to obtain an off-line prediction based on historical data;
- **Kalman Filtering Model**: used to adjust the off-line obtained in previous step by using live GPS data received from on-board computers.

After describing the elements introduced above, at the end of this chapter, both *Hybrid* and *Simple* models will be applied to a Real World case study with data obtained from Public Transport Service of Olbia (Italy), in cooperation with the PluService company.

3.2 Travel Time Prediction methods in Public Transport

The prediction of travel time have been demonstrated as one of the most important information for applications related to transportation of persons and logistic. By analysing this information, it is clear that it can be used by users to better understand the traffic condition in a determinate period of the day in urban areas. The knowledge of such information could help to reduce transport costs by avoiding congested sections and increase the public transport quality of service by reaching the pre-established stops within the required time window. It is a fact that the last century was characterized by the extreme expansion of urbanized areas and, at the same time, this kind of expansion has highlighted the need to develop efficient public transport systems all over the world. A contribution that helps to understand the importance of developing a well-organized and efficient transport system is represented by [117] where the focus is centred in revealing that most people strive for cleaner, less congested cities and improved traffic flow, primarily through increased use of enhanced public mass transit systems. In order to make people interested to use public transport with the aim of reducing traffic congestion, it is necessary to offer enhanced public transportation services by applying encouragement solutions that have been forwarded so far. One possible solution is represented by providing travelers with reliable travel information by means of Advanced Public Transport System (APTS) and Advanced Traveler Information System (ATIS), which are the primary key components in Intelligent Transportation Systems (ITS). One of the most important topic of ITS is represented by providing a precise travel time to both travellers and public transport operators in order to give them critical data for pre-trip and en route information which represent a very interesting information to have smart choices for travelers or to design better schedules for public transport operators [118], [119].

In years different approaches have been adopted in order to perform travel time prediction in public transport system and, these existing methodologies, have been categorized into five mostly wide types used of prediction models:

- **Historical Data Based models**: they use both static and dynamic information. Static information is represented by Historical Data that is provided by bus schedule information, recurrent traffic circumstances and

average dwell time. For what concerns, dynamic information, instead, it is provided by real-time measurements, like: real-time bus location data, delay at bus stops, and current traffic circumstances [120]. This kind of prediction algorithms provide current and future bus travel time from the historical travel time of the same bus of the previous journeys on the same time period. Historical approach predicts the travel time at a particular time as the average travel time for the same period over different days. The results of these models are satisfiable under standard circumstances, but their precision, when unexpected situations arise (like delays and traffic congestions), is seriously decreased [121]. In these models traffic patterns and street congestions on specific routes are supposed to be cyclic and then the prediction can be obtained by analysing the route's historical data. However when patterns are not stable, or the set of historical data is not *"large"* enough, these types of models can not adequately ensure acceptable performances;

- **Statistical models**: the sentence pronounced by Glymour makes it quite clear: *"Statistics is the mathematics of collecting, organizing and interpreting numerical data, particularly when these data concern the analysis of population characteristics by inference from sampling"* [122]. They have solid and widely accepted mathematical foundations and can provide insights on the mechanisms creating the data [123]. However, when the domain that must be described is characterised by complex and non-linear data, usually they cannot achieve good performances [124]. As a matter of fact, bus travel time is strongly influenced by several factors like: driver behavior, carriage way width, intersections, signals and etc. and those factors are represented as independent variables in many works. At the end, the performances of these models are strongly influenced by all the dependent variables that they can be incorporated in the model, but, this inclusion into the statistical model represents a very tough procedure [121];

- **Kalman Filtering models**: These models [125] have been used extensively in travel time estimation research. As Kalman [126] said that the Kalman model is used to get a prediction of the present status of the system, it can also used as basis in order to predict future values of variables available in former times. It is also demonstrated that the Kalman filtering model can easily adapt to traffic fluctuation by considering some time-dependent parameters [127]. These models are efficient in predicting travel on short time periods ahead, but their performances deteriorate considerably by using biggest time period [128]. In [129] used a Kalman filter model in order to estimate travel time with buses equipped with

the AVL1. In [130], authors used the Kalman filtering model together with GPS2 data installed on buses in order to predict bus arrival time on Indian traffic circumstances. In [131] authors have applied a macroscopic traffic flow model along with Kalman filtering algorithm to forecast travel time with combination of detector data and probe vehicle data;

- **Machine Learning models**: The Machine learning research branch is focusing its attention in studying how the computer can simulate or realize the behavior of human being. Machine Learning techniques are composed by two stages: choosing a candidate model, and then, obtain a prediction of the parameters model through learning process on existing data and examples [132]. Machine Learning algorithms works better then the statistical methods: In fact, Machine Learning techniques can deal with complex relationships between predictors that can come up within a huge volume of information and can process non-linear relationships between predictors by processing complicated and noisy data [133]. These models can be used for prediction of travel time, without implicitly addressing the traffic processes. Artificial Neural Network (ANN) and Support Vector Machine (SVM) algorithms are, however, the most representative Machine Learning technique used in travel time prediction;

- **Hybrid models**: A big number of researchers suggested to use hybrid frameworks for travel time prediction purposes. Van Lint in [134] proposed a mix of a linear regression model and a locally weighted linear regression model in a Bayesian framework in order to enhance forecast precision and reliability. Although their method may produce larger prediction errors weather each sub-model in the model layer is biased. Authors of [135] proposed a hybrid model based on State Space Neural Networks in cooperation with a particular filter as a trainer called Extended Kalman filter. The issue in SSNN is that the model requires large data set for offline training. It was proven also that ANN models outperform both historical data and regression models. In Jeong and R.Rilett [136] compared: historical data based model, Regression Models, and (ANN) Models. As result, authors found that ANN Models outperformed the historical data based model and the regression models in the case of estimation precision. In [137] authors claimed that neural network and Bayesian represents a good combination in order to estimate the urban arterial link travel times. Chen and Chien proposed, instead, in [138] a comparison among the link-based and path-based travel time prediction by using a Kalman filtering algorithm with simulated data. A real data analysis on the same approach was made by Kuchipudi and Chien in [139] where they proposed a hybrid model with combination of path-based and

link-based models on real data and under different traffic conditions. In [140] and in [141], authors reported the results obtained by applying a short term transit vehicle arrival times prediction algorithm with a combination of both real-time AVL and historical data source in Seattle, Washington. They used a Kalman filter model to track a vehicle location and statistical estimation for prediction of bus arrival time purpose. ;

In literature, variety of classifications for bus arrival time prediction models can be found, introduced by different studies. For instance, Lee [118] grouped them into four categories: regression method, time series estimation method, hybrid of data fusion or combinative model and artificial intelligence method. Sun [142], instead, stated them into three types of prediction models: models based on historical data, multilinear regression models, and artificial neural network models; however the most used classification of prediction methods in literature is the one presented by Lee (5 classification categories). A list of the most used algorithms is shown in Table 3.1.

Table 3.1: A list of the most used algorithms in literature

Authors	**Year**	**Model Type**	**Algorithm(s)**
Fei et al. [143]	2011	Hybrid	-Baesyan Model -FF and FB Logic
Yu et al. [144]	2011	Neural Network	-SVM
Yu et al. [145]	2010	Hybrid	-Linear Regression -Adaptive Custom Model
Padmanaban et al. [121]	2009	Kalman Filtering	-Kalman Filtering
Khetarpaul et al. [146]	2015	Hybrid	-Fuzzy Logic -BPNN
Cong Bai et al. [147]	2015	Hybrid	-SVM -Kalman Filtering
Zaki et al. [148]	2013	Hybrid	-ANN -Kalman Filtering

3.3 Travel Time Prediction Model

As previously introduced, an Hybrid Model is composed by two different steps that contribute in obtaining an accurate prediction of the travel time performed by a public transport service in urban areas. The steps included in this model

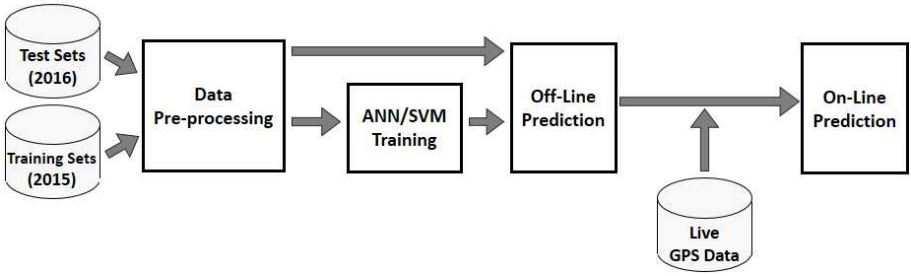


Figure 3.1: The schema representing the prediction steps of the Travel Time Prediction Hybrid Model developed.

are:

- **Historical Data Prediction**: obtained by using a dataset composed by data of travels performed during 2015 on Line 01 of Olbia's public transport service;
- **On-Line Data Prediction**: obtained by considering on-line GPS data from on-board computer during the progress of a specific journey on Line 01.

Historical Data prediction algorithms used in this work are Weka's Artificial Neural Network (ANN) and Support Vector Machine (SVM) with Radial Basis (RB) as Kernel Function [149]. As for the on-line prediction the Kalman Filtering algorithm has been applied [150]. In Figure 3.1 a simple and intuitive operation diagram show how the prediction process to obtain the predicted travel time operates. Before applying the *Historical Data Prediction* step, a *Data Pre-processing* phase is required in order to prepare both training and test dataset by eliminating variables with minor informative content. To perform this operation a Principal Component Analysis (PCA) has been applied to the whole dataset. In order to automatically perform some of the above mentioned processes, a dedicated software, called *"Urban Travel Time Predictor"* (UTTP), composed by a simple and intuitive graphical interface was developed. In section 3.5, UTTP will be then presented and discussed. In the following sub-sections, instead, the algorithms used in both Hybrid and Simple Models will be briefly introduced.

3.3.1 Artificial Neural Network (ANN)

An artificial neural network (ANN) consists of an input layer of neurons, from one to three hidden layers of neurons, and a final layer of output neurons. A feed-forward neural network is an artificial neural network where connections

between the units do not form a directed cycle. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes to, at the end, reach the output nodes. Each neuron of a layer is connected to every neurons of the sequent layer with a connection that is associated with a number called *weight*.

The output of the neuron has the sequent form:

$$h_i = \mu(\sum_{j=1}^{N} V_{ij} x_j + T_i^{hid}) \tag{3.1}$$

where μ is called activation function, N the number of input neurons, V_{ij} the weights, x_j inputs to the input neurons, and T_i^{hid} the threshold terms of the hidden neurons. The purpose of the activation function is, besides introducing non-linearity into the neural network, to bound the value of the neuron so that the neural network is not paralysed by divergent neurons. A common example of the activation function is the sigmoid function:

$$\mu(u) = \frac{1}{1 + exp(-u)} \tag{3.2}$$

However other activation function can be applied to neurons. In this work a Multi-Layer Perceptron (MLP) Neural Network has been used. MLP represents a class of feed-forward artificial neural network composed by three layer of nodes activated by non-linear activation function. In order to learn from a training dataset, a supervised learning technique, called Back-Propagation algorithm, is used [151]. Back-Propagation algorithm represent a generalization of the main square algorithm of the linear perceptron. The purpose of the learning algorithm is to minimize the error $e_j(n)$ represented as the difference subsisting among the desired output $d_j(n)$ and the obtained output $y_j(n)$. The weights are adjusted as following:

$$\epsilon(n) = \frac{1}{2} \sum_{j} e_j^2(n) \tag{3.3}$$

where the change of each weight is:

$$\Delta w_{jn}(n) = -\eta \frac{\delta \epsilon(n)}{\delta v_j(n)} y_i(n) \tag{3.4}$$

where y is the output of the previous neuron and η represents the *learning rate*. The chosen of the learning rate is important to assure that the weights quickly converge to a desired response.

3.3.2 Support Vector Machine with Gaussian Radial Basis Function

Support Vector Machine (SVM) [152] with Radial Basis Kernel Function (RBKF) represents a machine learning algorithm used, principally, to solve classification and regression problems. This classifiers can be divided into linear or non-linear classifiers by considering different kinds of *Kernel Function*. The hyperplane of linear SVM can be represented as:

$$\vec{w} * \vec{x} - b = 0 \tag{3.5}$$

where $\vec{w}$ represents the normal vector of the hyperplane, $\vec{x}$ is a set of points and b is the offset of the hyperplane. By considering that the training data are linearly separable, two parallel hyperplanes that separate the two classes of data can be selected, so that the distance between them is biggest as possible. The region bounded by these two hyperplanes is called the *"margin"*, and the maximum-margin hyperplane is the hyperplane that lies halfway between them. These hyperplanes can be described by the two equations:

$$\vec{w} * \vec{x} - b = 1 \tag{3.6}$$

$$\vec{w} * \vec{x} - b = -1 \tag{3.7}$$

In order to avoid the case that points fall into the margin, the two equations above can be re-written:

$$\vec{w} * \vec{x_i} - b >= 1 if y_i = -1 \tag{3.8}$$

$$\vec{w} * \vec{x_i} - b <= -1 if y_i = -1 \tag{3.9}$$

Obtaining that, in order to minimize the distance between planes $||w||$, the following rule must be applied:

$$y_i(\vec{w}\vec{x_i} - b) >= 1 forall 1 <= i <= N \tag{3.10}$$

A similar result can be achieved by considering data not linearly separable. The function shown above can be re-written by considering the *"hinge loss"* function:

$$max(0, 1 - y_i(\vec{w}\vec{x_i} - b)) \tag{3.11}$$

This function is called *"soft margin"* as it leaves more than the previous case.

The target in this case is set to minimize:

$$[\frac{1}{N}\sum_{i=1}^{N} max(0, 1 - y_i(\vec{w}\vec{x_i} - b))] + \lambda||\vec{w}|| \tag{3.12}$$

where λ value determines the tradeoff between increasing the margin-size and ensuring that the $\vec{x_i}$ lie on the correct side of the margin.

The non-linear classifier is formally similar to linear classifier, except that every *"dot"* product is replaced by a non-linear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The classifier represents an hyperplane in the transformed feature space, it may be non-linear in the original input space. By constructing a set hyperplanes the dimensional problem can be divided into infinite parts that, in the case of classification, separate perfectly the data into two classes. Whereas, in the case of regression, the set of hyperplanes must be constructed by considering that each hyperplane must be placed closest to as many points as possible.

The Kernel used in this work is represented by Radial Basis Kernel Function (RBKF) that is a real-valued function whose value depends only on the distance from the origin:

$$k(\vec{x_i}, \vec{x_j}) = exp(-\gamma||\vec{x_i} - \vec{x_j}||^2) for \gamma > 0 \tag{3.13}$$

Where γ can be parametrized as $\frac{1}{2\sigma^2}$

3.3.3 Kalman Filtering Model

Kalman filtering model is also known as linear quadratic estimation (LQE) and represents an algorithm that uses a finite series of measurements observed over time, that contains noise and other inaccuracies, in order to produces an estimation of unknown variables. This model takes his name from its maker Rudolf E. Kálmán [126] that in 1960 developed a new theory to face linear filtering and prediction problems.

The Kalman filter has been applied in a big number of applications. the most famous application is represented by the guidance, navigation, and control of vehicles. Furthermore, the Kalman filter model can be applied in time series analysis used in fields such as signal processing and econometrics. Kalman filters also are one of the main topics in the field of robotic motion planning and control.

The algorithm works in a two-step process. In the prediction step, the Kalman filter produces estimates of the current state variables, along with their uncertainties. Once the outcome of the next measurement (necessarily corrupted with some amount of error, including random noise) is observed, these estimates are updated using a weighted average, with more weight being

given to estimates with higher certainty. The algorithm is recursive. It can run in real time, using only the present input measurements and the previously calculated state and its uncertainty matrix; no additional past information is required.

The Kalman filter does not make any assumption that the errors are Gaussian. However, the filter yields the exact conditional probability estimate in the special case that all errors are Gaussian-distributed.

Extensions and generalizations to the method have also been developed, such as the extended Kalman filter and the unscented Kalman filter which work on non-linear systems. The underlying model is a Bayesian model similar to a hidden Markov model except that the state space of the latent variables is continuous and all latent and observed variables have Gaussian distributions.

The predicted *a priori* state is represented:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \tag{3.14}$$

while the predicted *a priori* covariance is:

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{3.15}$$

where the state of the filter is represented by the *a posteriori* estimate at time k $\hat{x}_{k|k}$ and *a posteriori* error covariance matrix $P_{k|k}$. The updated *a posteriori* state is:

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k y_k \tag{3.16}$$

while the *a posteriori* covariance is:

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \tag{3.17}$$

For the purpose of this work, this model can be adapted by considering that instant speed of the vehicle can be viewed as a series of values recorded in time. The instant speed $s(t|t)$ can be set as system status variable where t represent a specific *time moment* in which the value of speed has been recorded. By considering the two equations described before, the predicted a priori state can be represented as (without considering input vector):

$$s(t|t-1) = s(t-1|t-1) \tag{3.18}$$

$$P(t|t-1) = P(k-1|k-1) + Q \tag{3.19}$$

$$s(t|t) = s(t|t-1) + Kg(t) * (s_r(t) - s(t|t-1)) \tag{3.20}$$

$$P(t|t) = (1 - Kg(t)) * P(t|t-1) \tag{3.21}$$

where $kg(t) = \frac{P(t|t-1)}{P(t|t-1)+R}$ represents the Kalman Gain, $v_r(t)$ is the observed instant speed of the bus; $P(t|t)$ stands for the calculating covariance; Q is the covariance of the transforming system and R is the observing covariance. Adopting the weighting arithmetic in order to combine the baseline data with the instant speed, the predicted section travel time of the first step of the algorithms $\hat{T_u}$ can be transformed to travel speed s_u:

$$s_u = \frac{S_u}{\hat{T_u}} \tag{3.22}$$

where S_u represents the distance (in meters) subsisting among stop u and $u+1$. The auxiliary speed s_u^{aux}, representing speed variable, the weighting method can be expressed as:

$$s_u^{aux} = \frac{S_u^f * s(t|t) + S_u^l * s_u}{S_u^f + S_u^l} \tag{3.23}$$

where S_u^f is the length (in meters) of the section passed by the vehicle, while S_u^l represents the length of the section that must be passed by vehicle to reach the stop $u+1$. It's evident that by adopting variable weights can different dependence be given to the auxiliary speed: when the bus is more close to the stop u , s_u^{aux} depends on s_u more, while the bus is more close to the stop $u+1$ s_u^{aux} depends on $s(t|t)$ more. Sequentially, when the bus is on the segment between stop $u0-1$ and $u0$, the predicting travel time T_{pred} from the real-time location to stop $u1$ should be given as:

$$T_{pred} = \frac{S_{u0-1}}{s_{u0-1}^{aux}} + \sum_{u=u0}^{u1-1} \frac{S_u}{s_{u0-1}^{aux}} \tag{3.24}$$

3.4 Experimental Setup

In this section the set-up used in the performed experiments will be presented and discussed. Starting from a detailed description of the dataset (Paragraph: 3.4.1) used to test and validate the proposed models, the sequent two phases will be then faced:

- data Pre-Processing (Paragraph: 3.4.1);
- ANN/SVM Parameters set-up (Paragraph: 3.4.2);

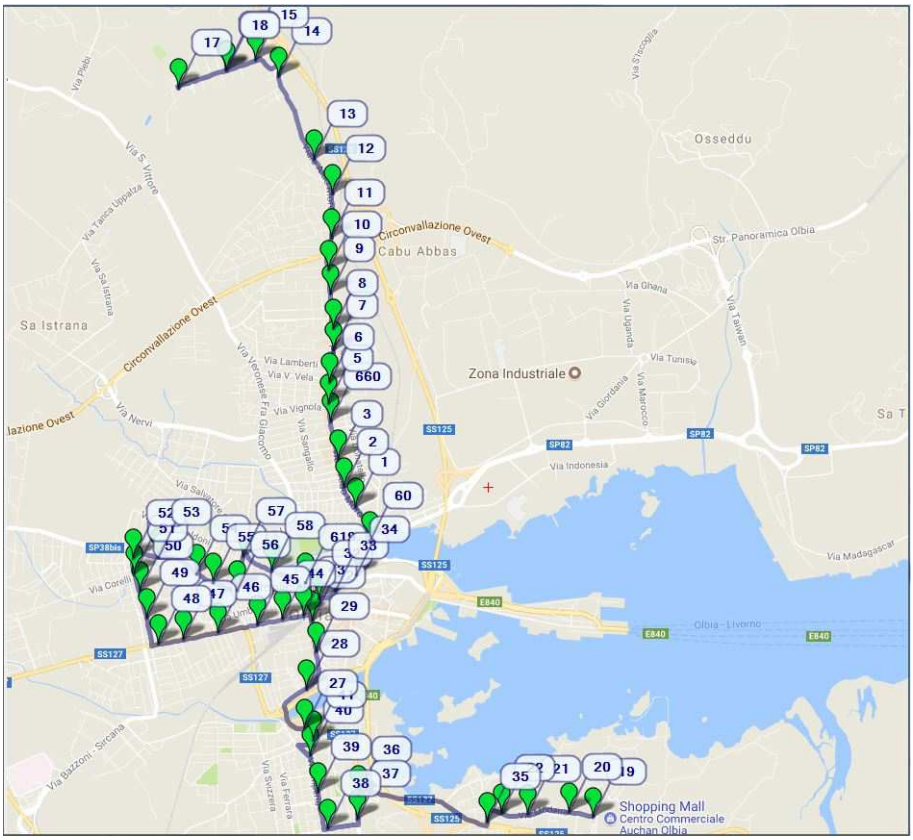


Figure 3.2: The chosen route of Public Transport Service in Olbia, Italy.

3.4.1 The DataSet: Bus Line 1 of Olbia PTS

The proposed models for bus travel time prediction have been evaluated by choosing a real-world data in Olbia, Italy (Figure 3.2). The chosen route is composed by 52 road segments where, at the end of each segment, is placed a bus stop where passengers can get on or get off from vehicles. The route is long about 5 Km. and takes place in a urban area characterized by an high traffic congestion in the morning (about at 8 a.m.), at launch time (about at 1 p.m.) and in the evening (about at 7 p.m.). In the remaining part of the day the traffic does not particularly affect the travel time between two consecutive stops. The collected data consists of the arrival time of the 52 stops in each individual trip performed from the 1st of January of 2015 to 1st May of 2016 and consisted in a total of 5878 valid trips performed with 305656 travel times recorded.

Before proceed, some concept belonging to Public Transport Sector must be clarified.

Concepts and Terminology

In this subsection some concepts and specific terminologies will be introduced and defined in order to help readers in understanding some typical vocabulary used in the Public Transport Sector. The main concepts considered in this work are (Figure 3.3):

- **GPS Point**: it represents a unique point on hearth and it is composed by: Latitude, Longitude and Timestamp;
- **Edge**: it is the GPS point that identifies one of the vertices of a Line. These vertices can represents a **Simple Edge** (top of a specific line) and **Stop Edge** (start or end point of a route);
- **Line**: A line is a segment that combines two **Simple Edge** points. It is identified by a straight line where the length is expressed in meters;
- **Polyline**: it is a *"composite line"*, formed by an orderly sequence of Line elements. The Polyline can represent two distinct concepts: **Route** representing an orderly sequence of Lines that connect two Stop Edges (for examples two bus stops) and **Path** Representing an orderly sequence of Route that connect the starting and ending points of a Bus travel line (for example starting bus stop and ending bus stop);
- **Path Network**: it represents the set of all available Paths for the considered Public Transport Service.

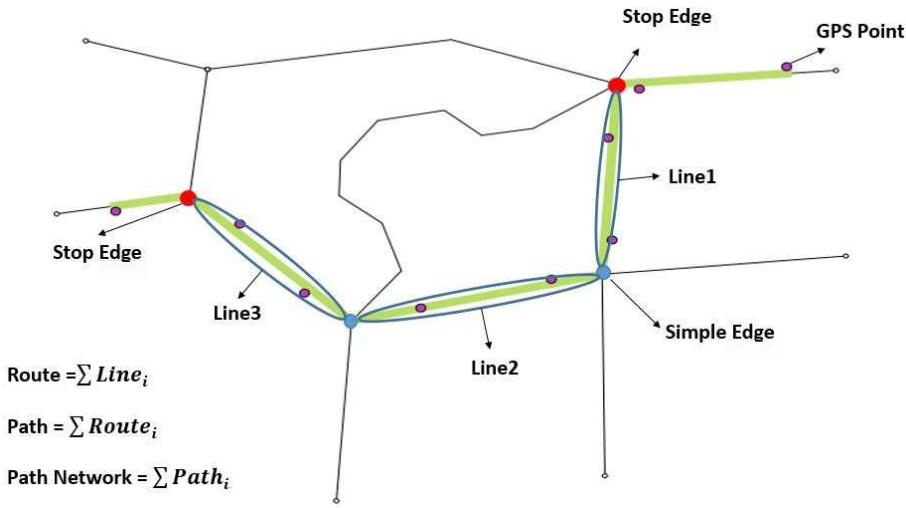


Figure 3.3: A schema representing the objects described in 3.4.1.

Dataset Pre-Processing: PCA

Principal Component Analysis (PCA) represents a statistical procedure that apply orthogonal transformations in order to convert a set of related variables into a set of linearly independent variables, also called *"Principal Components"* [153]. Some trips (about the 5% of the totality) were not considered cause the lack of data recorded (bus skipped those point or on-board computer were not active at the time point). The original Dataset considered was composed by the following attributes:

- **Route Identifier**: it is a discrete value and represents the unique ID of the Route in the Path Network;
- **Month**: it is an integer number between 1 and 12 representing the corresponding month of the year (1=January, 2=February, etc.);
- **Day**: it is an integer number between 1 and 31 representing the corresponding day of month;
- **Hour of day**: and integer number between 0 and 23 representing the corresponding hour of the day;
- **Route Length**: a float value representing the length expressed in meters, of the Route belonging to the analysed Path;
- **Previous week Average Speed**: the average speed (meters in seconds) of the vehicle recorded in the ride performed last week in the analysed Route;
- **Two Past week Average Speed**: the average speed (meters in seconds) of the vehicle recorded in the ride performed two weeks before in the analysed Route;
- **Previous week Travel Time**: it represents the difference (in seconds) subsisting among the recorded times at the start Stop Edge and the end Stop Edge of the analysed Route of the last week;
- **Two Past week Travel Time**: it represents the difference (in seconds) subsisting among the recorded times at the start Stop Edge and the end Stop Edge of the analysed Route of the past two week;

The PCA analysis on the above mentioned dataset was performed by applying a RapidMiner process where for each attribute, the corresponding *"Principal Component"* with the associated information content were calculated (Figure 3.4). By considering the results obtained with the PCA process on dataset, the attribute considered passed from 9 to 6 (Figure 3.5). The chosen attributes are:

	Standard Deviation	Proportion of Variance	Cumulative Variance
PC 1	40.371	0.29982844	0.29982844
PC 2	26.799	0.199031542	0.498859982
PC 3	21.956	0.163063418	0.6619234
PC 4	15.741	0.116905687	0.778829086
PC 5	14.671	0.108958982	0.887788068
PC 6	10.751	0.079845819	0.967633887
PC 7	3.098	0.023008311	0.990642198
PC 8	0.92	0.006832681	0.997474879
PC 9	0.34	0.002525121	1

Figure 3.4: The resulting PCA components and their values of Standard Deviation, Variance and Cumulative Variance.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
Route ID	0	0	0	-0.002	0.003	0	0.147	-0.989	0
Month	0	0	0	0.024	-0.019	-0.002	-0.989	-0.147	0
Day	0	0.029	0	**-1**	0.006	-0.004	-0.024	-0.001	0
Hour	0.002	0	0	0.001	-0.006	-0.04	-0.008	0	-0.949
Route Length	**0.992**	-0.124	0.003	0	-0.025	-0.001	0.001	0	0
Previous week Travel Time	0.089	**0.718**	0.686	0.002	0.059	0.049	-0.001	0	0
Previous week Average Speed	0.01	-0.065	-0.044	0.001	0.684	**0.725**	-0.015	0	0
Two Past week Travel Time	0.089	0.679	**-0.725**	0.002	0.066	-0.046	-0.001	0	0
Two Past week Average Speed	0.01	-0.063	0.052	0.007	**0.723**	-0.686	-0.013	0	0

Figure 3.5: The eigenvectors associated with the attributes of the dataset.

Day, **Route Length**, **Previous week Average Speed**, **Two Past week Average Speed**, **Previous week Travel Time** and **Two Past week Travel Time**.

On the other hand, **Route ID**, **Month** and **Hour of Day** were discarded because of poor informative content they possessed.

3.4.2 Neural Networks Parameters set-up

To perform the set-up of the parameters for the Neural Networks used in this work, specific RapidMiner processes have been realized. The two processes have the same flow structure, the only differences between them are: the parameters to be tested and the algorithm on which these parameters will be used (ANN or SVM). Since the processes are essentially the same for both algorithms, the flow structure will be briefly introduced in the continuation of this paragraph and will be valid for both SVM and ANN neural networks.In the first step of the RapidMiner process, the dataset is prepared to be passed to the *"Optimize Parameter"* operator which, essentially, finds the optimal values of the selected parameters in its sub-process and delivery then to a specific log file. The sub-

process blocks perform the optimization of the parameters. While *"Macro"* and *"Log"* operators do not play any role in parameter optimization but are only used to automatize the storage of the log file, the *"Validation"* operator play a key role in the entire process. This operator performs a cross-validation in order to estimate the statistical performance of the learning operator (ANN or SVM). In the present case, the number of validation was set to 10 which means that the Dataset was divided into 10 parts (each subset has equal number of examples) and the algorithms was trained, iteratively, with the selected part and tested with the remaining 9 parts. In fact the *"Validation"* operator is a nested operator and posses 2 sub-processes: Training and Testing. The training sub-process is used for training a model and is composed by *"Nominal to Numerical"* (in order to transform nominal into numerical values) and *SVM* or *Neural Net* operators (representing the algorithm used). The trained model is then applied in the testing sub-process. The results of each iteration of the entire process is saved in a log file in order to evaluate the performance achieved and select, them, the optimized parameters for each algorithm. In the next sub-section the optimal parameters chosen and the effective training process will be presented for both ANN and SVM algorithm.

ANN Parameters and Training Process

The Optimize process presented above is used in the case of Artificial Neural Network in order to find the best tuning regarding the following parameters:

- **Learning Rate**: this parameter determines how much the weights can change at each step of the learning process. If the value of this parameters is setted too high the performance of the neural net will diverge; on the contrary if it is setted too low the learning is too small and the error rate doesn't descend rapidly enough. The search of the optimal value of this parameter was set in the range $[0.1 - 0.9]$ with a step of 0.1;

- **Training Cycles**: this parameter specifies the number of training cycles used for the neural network training. If this value is too low the training process may be interrupted before the minimum error rate is reached, on the contrary, if the value is too high, the local minima of the training error function can be exceeded. The training cycles was set in the range $[100 - 500]$ with a step of 100;

- **Error Epsilon**: this parameter represents the *"stop point"* on which the optimization process is stopped if the training error gets below this epsilon value. It's evident that an optimal value is represented by the lowest value of this parameter. To find the lowest reachable value the search was set in the range $[0.0001 - 0.004]$ with a step of 0.0001;

In total 200 tests were performed and from the obtained result emerged that the best configuration for the ANN algorithm was:

- **Learning Rate**: 0.4;
- **Training Cycles**: 100;
- **Error Epsilon**: 0.0001.

After the optimized values were obtained, the training process of the Artificial Neural Network could be performed. As for the optimization process, a specific RapidMiner work-flow has been created in order to perform the training of ANN. Here, the dataset is splitted into two parts: Training that contains all the performed trips in 2015 and testing that, on the contrary, contains all the trips performed from the 1st of January to the 1st of May of 2016. At the end of the training process the obtained trained model is saved in a XML file so that UTTP software tool can use it at a later time, while the performance results are saved on a CSV file in order to evaluate them.

SVM RBKF Parameters and Training Process

As for the ANN algorithm, the optimization process is applied also with SVM RBKF algorithm. In this case the parameters that must be optimized are:

- **Maximum Number of Iterations**: this parameter determines when the training phase must be stopped.The stop signal arrives after a specified number of iterations. If the value of this parameters is setted too high the performance of the SVM are to slow and the training session can be too long; on the contrary, if it is setted too low the algorithm doesn't have an enough number of iterations in order to minimize the error function. The search of the optimal value of this parameter was set in the range $[1000 - 10000]$ with a step of 1000;
- **Epsilon Value**: This parameter specifies the insensitivity constant and it is part of the loss function. Is not easy find a good confidence for this value since it is strictly dependent on the application domain. For this reason the value was set in the range $[0.1 - 0.9]$ with a step of 0.2;
- **C value**: this parameter represents the SVM complexity constant which sets the tolerance for misclassification, where higher C values allow for *"softer"* boundaries and lower values create *"harder"* boundaries. A complexity constant that is too large can lead to over-fitting, while values that are too small may result in over-generalization. To find a good compromise value the search was set in the range $[1000 - 10000]$ with a step of 1000;

In total 500 tests were performed and from the obtained results, as for ANN, emerged that the best configuration for the SVM RBKF algorithm was:

- **Maximum Number of Iterations**: 7000;
- **Epsilon Value**: 0.1;
- **C Value**: 7000.

Also for SVM a training RapidMiner work-flow was created and the dataset was splitted in the same way.

3.5 Urban Travel Time Predictor Software (UTTP)

In order to realize the experiments introduced so far in a simpler and faster way, a software that perform in an automatic fashion the entire prediction process has been realized and implemented. The software is called *Urban Travel Time Predictor* (UTTP) and has been realized by adopting the Microsoft .NET development environment. The choice fell on this technology cause the usage environment was entirely based on Windows platform and, moreover, the C# programming language, being optimized for the object oriented paradigm, represented a good compromise among implementation simplicity for graphical application and software performances. The whole process that UTTP starts with the creation of the required dataset tables to arrive at the real-time graphical display of the travel route under review with the predicted, real and scheduled times highlighted for each stop point. UTTP is divided into three main parts:

- **Configurations** (Figure 3.6): it allows users to set-up the entire process. Here users can define various parameters such as: how software connects with the database engine, the tables name where the required data can be found, the type of Neural Network that must be used, the path of the RapidMiner processes, etc.;
- **Path Network Creation** (Figure 3.7): this part allows users to chose the route that must be analysed. In this work the attention was focused on Line 01 of Olbia's PTS, but the UTTP software is ready to be used with any route that is part of the transport service. When user choose a route the system, automatically, creates the dataset tables on database and starts, in background, all the RapidMiner processes introduced before in order to: pre-process data, optimize Neural Network parameters, train and test Neural Network. At the same time, it builds the polyline that defines the path chosen to be displayed in the next part of the software;

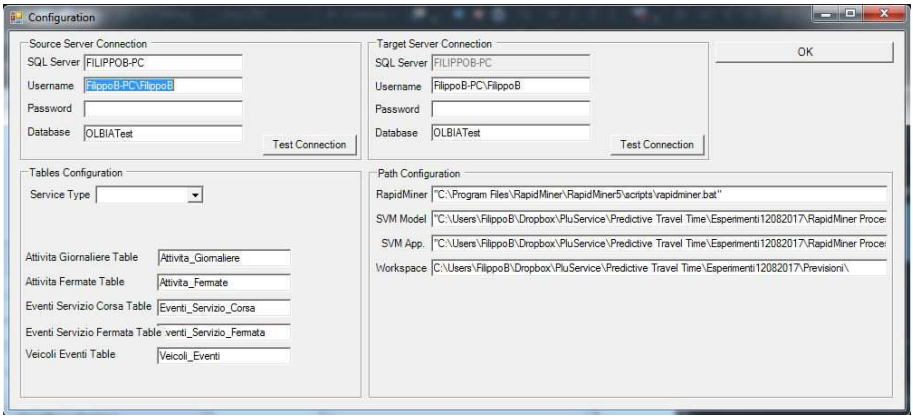


Figure 3.6: UTTP Configuration window.

- **Prediction And Results Visualization** (Figure 3.8): this part represents the core of UTTP. Here the prediction model obtained in the previous step is used to predict arrival times at bus stops of a specific ride of 2016. The window is divided into 2 different parts: in the left part user can select the ride on which perform the prediction, while on the right part of the window the user can appreciate a graphical representation of the path chosen with highlighted the predicted arrival times at bus stops. After obtained the *"Off-Line"* predictions, users, by clicking the button *"Starts On-Line Prediction"*, can start the application of the Hybrid Model by integrating the previously obtained Neural Network prediction with the Kalman Filtering model by using stored GPS signals of the ride in analysis. The Kalman Filtering Model introduced in paragraph 3.3.3 has been completely implemented in C# in a specific UTTP's library.

At the end of each prediction process all results are stored in excel files in order to allow users in performing further analysis. The analysis obtained for Line 01 of Olbia's PTS are discussed in the following section.

3.6 Results

The quality of the models presented in the previous sections was tested by using the data of Line 01 of Olbia's PTS and the result are shown below. The entire path is divided in 52 sections where, for each section, a prediction is obtained by considering both the application of the Simple Models (ANN or SVM) or the Hybrid Models (ANN+Kalman Filtering Model or SVM + Kalman Filtering Model). In order to measure the performances of each model, three different

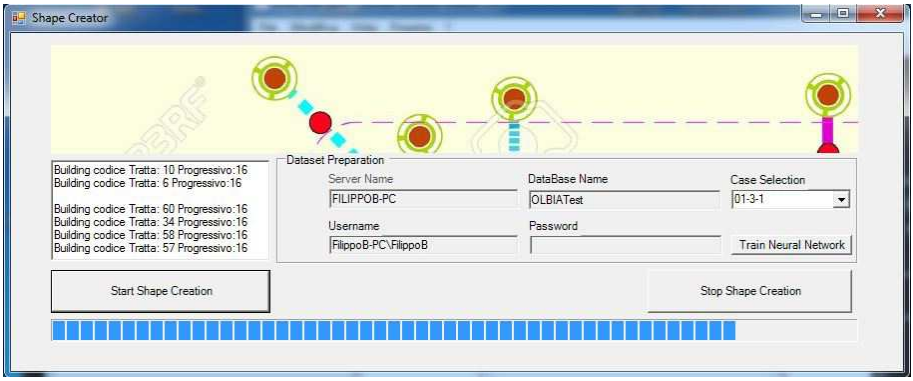


Figure 3.7: UTTP Path Network and Neural Network creation window.

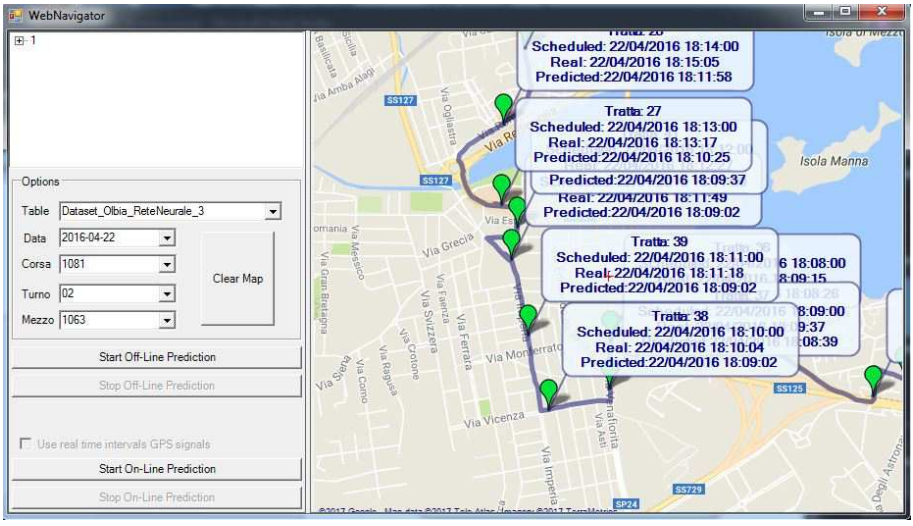


Figure 3.8: UTTP Path and Results visualization window.

Table 3.2: The results obtained for the four models tested on Line 01 of Olbia's PTS

	MAE	MAPE	RMSE
ANN	9.03 s.	21.03%	15.47%
SVM	8.64 s.	21.19%	15.31%
ANN + Kalman Filtering	8.58 s.	20.77%	11.65%
SVM + Kalman Filtering	7.17 s.	19.42%	9.59%

efficiency measures were used:

- **Mean Absolute Error**: $\frac{\sum_{i=1}^{N} |x_i - \hat{x_i}|}{N}$ that represents the difference between two continues variables;
- **Mean Absolute Percentage Error**: $(\frac{1}{N} \sum_{i=1}^{N} \frac{|x_i - \hat{x_i}|}{x_i})$ that provides unit-free measurement of the performance;
- **Root Mean Square Error**: $\sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \hat{x_i})^2}$ that measures the goodness-of-fit of the predictor;

where x_i represents the actual data, $\hat{x_i}$ the predicted data and N is the number of samples considered. The ride chosen was performed on 22 April 2016 performed by bus number 1063. From the results obtained (Table 3.2) it is evident that the Hybrid Models represented by the combination of the SVM RBKF (or ANN) and the Kalman Filtering algorithm (Figure 3.12 and 3.11) outperforms the other two simple applications tested (Figure 3.10 and 3.9). For each predictor, by comparing the prediction results of MAE, MAPE and RMSE it is evident that Hybrid Models can adapt more easily to stochastic events than Simple Models, based only on Historical Data. This thesis become obvious by observing the section between the stops 13 and 19, where several intersections and traffic lights are present. The presence of these elements cause traffic fluctuations that makes travel time more sensible to stochastic events that are not easily predictable by studying only historical data. On the contrary, the Kalman Filtering Model, by adding an estimate of the present status of the system (bus ride in this case), makes the global Hybrid Model to adapt more rapidly to stochastic events and makes it more suitable in travel time estimation for urban public transport systems then the *Simple Models*. By observing numbers, in general, the Hybrid Models have better performances in the totality of the path of analysis; RMSE for Hybrid Models are near the 10% (ANN + Kalman Filtering 11.65%, SVM + Kalman Filtering 9.59%) while, considering the Neural Networks models, its value grows up to, about, 16% (ANN 16.38%, SVM 15.31%). MAE and MAPE measures reflect the trend highlighted by the RMSE indicator where Hybrid Models outperforms Simple Models.

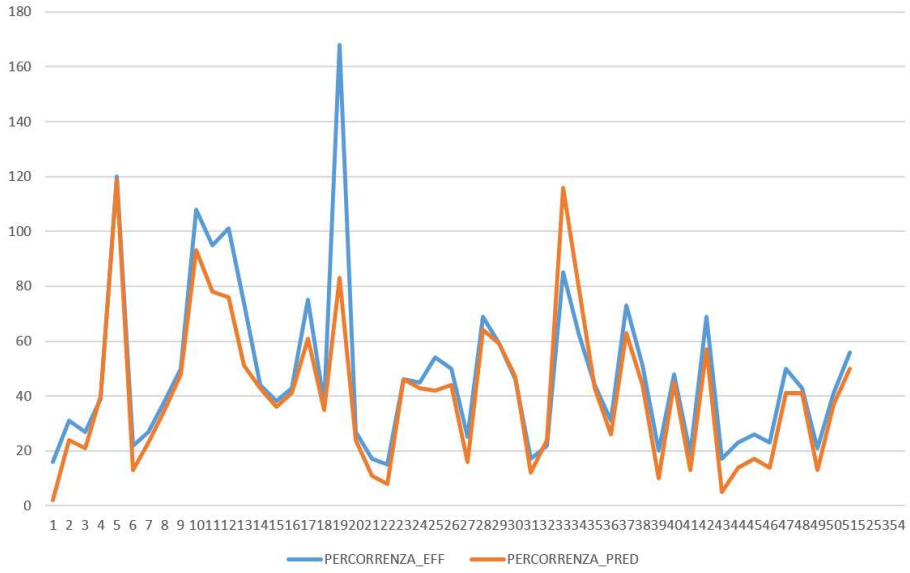


Figure 3.9: Differences among the ANN prediction and real travel times of ride 1081 performed the 22 April 2016 by bus 1063. In light blue is represented the real travel time while in orange the predicted travel time is shown. In the y axis time is represented in seconds while the measures in x axis represents the sequence of bus stops performed.

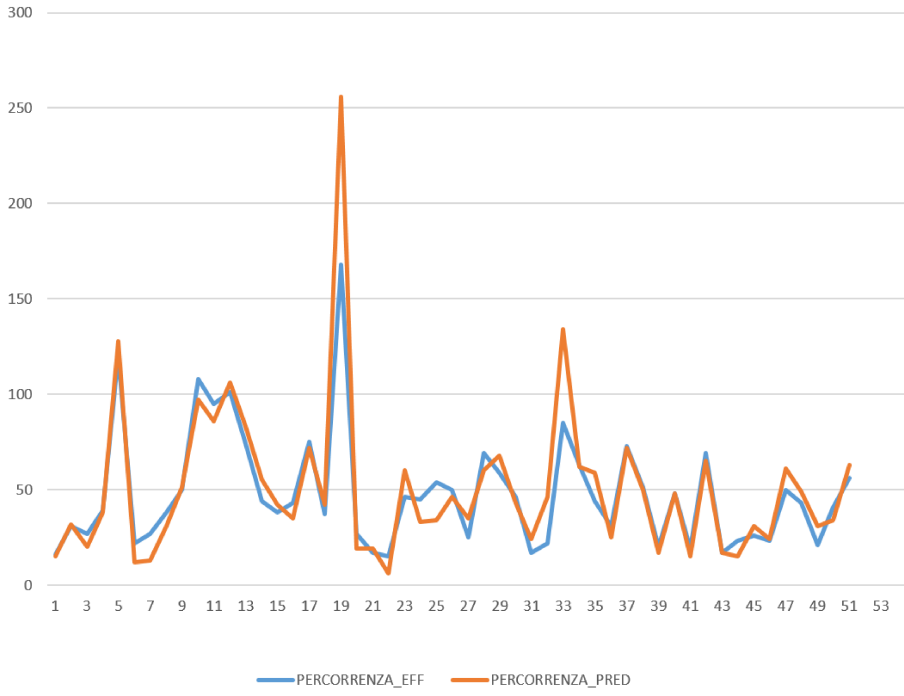


Figure 3.10: Differences among the SVM prediction and real travel times of ride 1081 performed the 22 April 2016 by bus 1063. In light blue is represented the real travel time while in orange the predicted travel time is shown. In the y axis time is represented in seconds while the measures in x axis represents the sequence of bus stops performed.

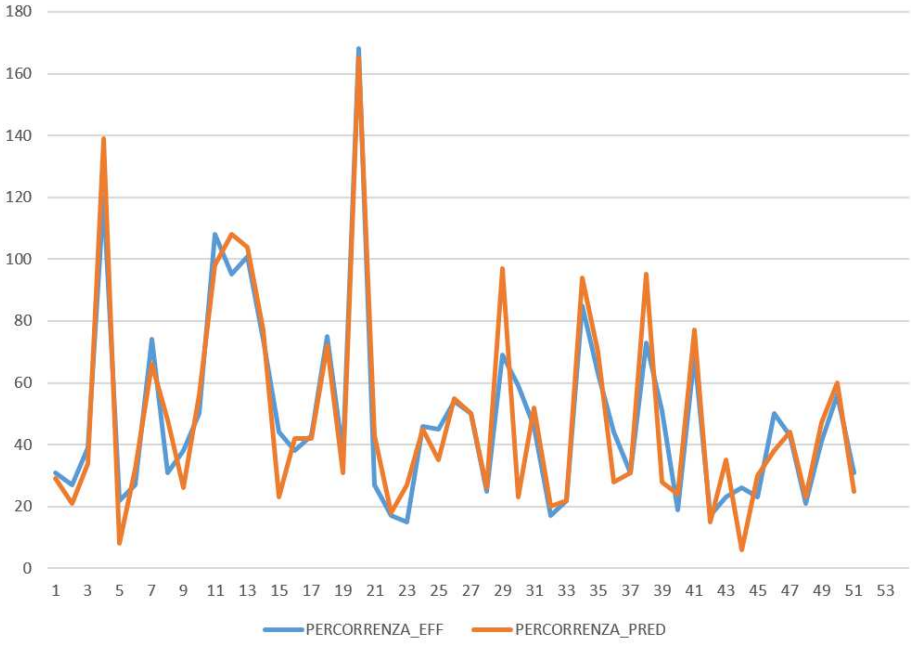


Figure 3.11: Differences among the ANN+Kalman Filtering prediction and real travel times of ride 1081 performed the 22 April 2016 by bus 1063. In light blue is represented the real travel time while in orange the predicted travel time is shown. In the y axis time is represented in seconds while the measures in x axis represents the sequence of bus stops performed.

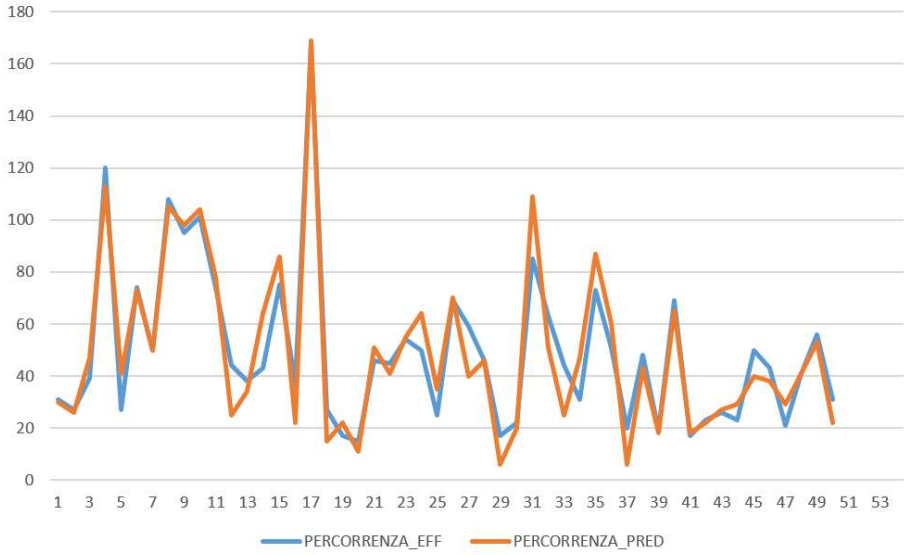


Figure 3.12: Differences among the SVM+Kalman Filtering prediction and real travel times of ride 1081 performed the 22 April 2016 by bus 1063. In light blue is represented the real travel time while in orange the predicted travel time is shown. In the y axis time is represented in seconds while the measures in x axis represents the sequence of bus stops performed.

Chapter 4

Building Detection in Urban Areas with High Resolution Aerial Images

4.1 Overview

The problem of building detection from remote sensed data is a well known in the scientific community that merges approaches of remote sensing, machine learning and image processing. The use of remote sensed data acquired from satellite or airborne pay-loads enormously simplifies the overall work-flow opening the way to an accurate and precise mapping of infrastructures with a lot a applications that vary from map updating to Land Use / Land Cover (LU/LC) mapping.

From the research point of view the problem is interesting and important especially when dealing with high-resolution imagery acquired by manned/ unmanned aerial vehicles. The main problem is to automatize the overall work-flow especially the classification stage in order to: increase the temporal resolution; reduce the costs; setup an automated tool chain to derive repeatable results in terms of quality.

The current trend is to detect buildings from stereo pair-images or fused LiDAR and multi-spectral (MS) data. Obviously the end-user is interested to increase the spatial resolution that sets an hard constraint on the payload. The use of MS and LiDAR ensures a final good Ground Sampling Distance (GSD) that allows to derive high-resolution thematic maps.

When the resolution increases a problem arises: the problem is the variety of buildings on a given area in terms of size, shape, spectral response. Many supervised and unsupervised approaches have been proposed [154]. In the case of supervised approaches the definition of the training set plays a key role and this is the phase where the human expert defines the training samples. The perfect definition in terms of accuracy and significance is hard to ensure and often the problem is the unbalancing of training set samples.

In this thesis very high resolution multi-spectral and LiDAR data is considered in order to prove the effectiveness of the Bayes Vector Quantizer (BVQ)

[155] for the classification of noisy data (high resolution) with a strongly unbalanced problem that outperforms other state of the art algorithms. The case study covers a urban area where the most representative classes are building, tree, land and grass.

The BVQ classifier were adapted to the problem of building detection form multi-source aerial data in an urban area in order to compare the performance for different level of unbalancing. One of the main advantage of BVQ is the capability to handle with strongly imbalanced training set, which is a common problem for supervised approaches [156]. The unbalancing were stressed to evaluate the overall performance over the creation of supervised training set that represents a critical aspect to get accurate and precise results.

This chapter is structured as follows. The following sub-section introduces an overview of KDD in history and the main algorithms of KDD used in building detection domain. Section 4.3 introduces the BVQ algorithm. Section 4.4 presents the experimental setup including the datasets and evaluation metrics. In Section 4.5 the obtained results are discussed.

4.2 Related Work

4.2.1 Knowledge Discovery in Database

"Information is the resolution of uncertainty". With this expression, Claude Shannon tried to define the importance of information, as knowledge, in the modern world because *"Information is Power"* (J. Edgar). These phrases help to understand that, from the huge amount of data that is available today, it is crucial to extract the most useful amount of information for the goals that must be pursued. In the last 40 years data management principles such as physical and logical independence, declarative querying and cost-based optimization have led to profound pervasiveness of relational databases in any kind of organization. The 90's have been exceptional years for the invention and development of KDD techniques with solid data mining and machine learning algorithms. The early 90s, in particular, represented a true *"boom"* for the development of data mining algorithms. In [157], for example, authors present an interesting algorithm, based on estimation and a pruning technique, that can be used to generates and provides to users a number of significant association rules among items in a relational database. They have also proved the effectiveness of this algorithm by applying it on a real-world case study based on sales data obtained from a large retailing company. Also in [158] Mikhail V.Kiselev have described **PolyAnalyst**: a learning technique used in intelligent analysis of the experimental/observational data created in the Computer Patient Monitoring Laboratory at the National Research Center of Surgery in Moscow. This sys-

tem has been developed in 1994 in order to support users in performing three main activities:

1. Construction of a procedure realizing the mapping from the set of descriptions to the set of parameters given by the pairs <description, parameter>;

2. Search for the interdependences between components of the descriptions;

3. Search characteristic features of a gives set of description.

PolyAnalyst was applied to solve classification problems, empirical law inference and choice of the best decision from a fixed set of possible decisions. In [159], instead, authors have highlighted the performance limitation of PolyAnalyst in real-world case studies and have identified three main problem that afflicts this algorithm: Searching in very wide hypotheses spaces consumes great computation time. Most undesirable feature is strong dependence of computation time on number of attributes in explored relational database (number of independent variables). Every method based on the least squares criterion is vulnerable to even small number of far outliers originated from various data collection errors while it may be quite tolerant to strong normally distributed noise added to true values. There exists no good general criterion for search termination. As partially find a solution, authors proposed a supplementary data preprocessing step whose purpose is to try to reduce the noise present in the input data. It is clear that data mining techniques used alone lose much of their potential and need a more robust approach in order to face the problem that must be issued. At the end of 90s the growing number of analytical platforms have pushed the researchers to set up robust methodological approaches in order to develop analytical processes capable of extracting valuable knowledge out of large masses of data and try to release the unexpressed potential of data mining techniques, giving the birth to KDD [160]. KDD has evolved in the last years, and continues to evolve, from the intersection of research fields such as machine learning, pattern recognition, databases, statistics, AI, knowledge acquisition for expert systems, data visualization, and high-performance computing. The unifying goal is extracting high-level knowledge from low-level data in the context of large data sets. Very often is really difficult try to understand the differences subsisting among the concepts of *"Data Mining"* and *"Knowledge Discovery in Databases"*. In [160] a basic definition of KDD is given and authors have defined it as *"the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data"*. Authors of [160] with this definition, have meant that KDD represents a complex process consisting essentially of five fundamental steps (Data Selection, Data Preprocessing, Data Transformation, Data Mining, Data Evaluation) and that

the application of Data Mining techniques is only a single step of the whole process.

From that moment KDD processes were adopted to find Knowledge in a big number of application domains. As previously said, they can be divided, accordingly with the architecture of the systems, in three different generations [161]:

- **First Generation**: stand-alone software;
- **Second Generation**: distribuited architectures;
- **Third Generation**: semantic-based platforms.

In the 1-st generation are included systems that supports single users in local settings. Examples of application of this generation of architectures can be found in a different number of domains. In [162], [163] and [164] authors have compared some tools and solutions whose success was very limited due the extreme specificity and both reduced flexibility and re-usability in different application areas. Over years, more interesting tools have been developed (like Weka [149]). These tools have been applied in different works. For example in [165] authors have compiled a top-ten ranking of classification algorithm, based on performance, in order to face one of the major research problems in network security: Intruder Detection. Authors selected a list of ten algorithms, namely: J48, BayesNet, Logistic, SGD, IBK, JRip, PART, Random Forest, Random Tree and REPTree to perform classification with NSL-KDD dataset and chose, as environment for experimentation, Weka management tool. In [166], instead, authors described an implementation of a Knowledge Discovery in Databases (KDD) process, with RapidMiner 5.0 tool, in order to extract the causes of iterations in Engineering Change Orders. The data set considered was composed by, approximately, 53.000 historical Engineering Change Orders used for this purpose. The obtained result identified as the main cost and technical reasons for the occurrence of iterations. The study concludes that applying KDD in historic Engineering Change Orders data can help in identifying the causes of iterations in Engineering Change Orders. Extensions of these tools are easy to find thanks to the presence of a well-stocked market, but some needs can not be met by extensions already made and must be created to its own use. In [167], [168] and [169], for example, authors have developed some extensions in KNIME introducing new features to solve specific classification problems which could not be solved with the tools originally present.

With the spread of network technologies, however, the late 90s has witnessed the birth of support systems based on distributed architectures: Client-Server, Agent-based, Service-Oriented Architectures, Grid and Cloud [161]. As example, in [170] authors analysed the requirements for data mining systems used in

big organisations and enterprises range from logical and physical distribution of large data and heterogeneous computational resources with high performance. In this work authors have separated the data representation (client-side) from the data manipulation functionalities (server-side) obtaining a parallel computation divided into two different levels. In [171] authors faced the problems associated with efficient parallelizations on a Network of Workstation (NoW), including data transmission over a low bandwidth network, load-balancing, fault-tolerance, interactivity and programming complexity. To address some of these problems, they proposed a programmable, distributed and generic mechanism, which schedules a set of independent tasks on a NoW. This architecture allows incremental reporting of results, which can be used to monitor and, if necessary, interrupt the operation. Most importantly, it reduces communication bandwidth requirements by allowing specification of resource requirements of the tasks at the application programming level. Others applications used a similar, but specular, approach than distributed systems by considering an agent based architecture. This kind of applications adopted the mobile-agent paradigm which can move to the nodes where data reside in order to perform data analysis tasks and coordinating with others agents to achieve the KDD goal. In [172] authors described a component based system for developing distributed data mining applications, called PaDDMAS, that provides a tool set for combining pre-developed or custom components using a dataflow approach, with components performing analysis, data extraction or data management and translation. Each component can be serial or parallel object, and may be binary or contain a more complex internal structure. Concepts of Service Oriented paradigm has been also adopted to, initially, increase performances of KDD applications. In [173] authors focused their attention in developing a new approach to build a service-oriented infrastructure for distributed data mining applications. This kind of infrastructure allow users to use the provided data mining algorithms without the need to know the details about their operational functionalities. In addition this framework allow to perform data mining computations in more than one site, in a distributed manner. A similar approach has been adopted in [174] where authors in order to gives a comprehensive overview of various approaches in different stages of the knowledge discovery process, use a Linked Open Data in various stages for building content-based recommender systems. In the last years Cloud Computing has been adopted also in KDD environment as emerging trends for data mining applications in science and industry. In [175], for example, open standards and, in particular, Predictive Model Markup Language (PMML) are presented and analysed mainly from the point of view of the benefits of interoperability that the use of such technologies will give to the final users. An assessment of emerging technology trends and the impact that cloud computing will have on applications

is also discussed. In [176] authors describe the design and implementation of a high performance cloud that used to archive, analyse and mine large distributed data sets. As storage service, required by Sphere compute cloud, authors have designed a Sector storage cloud that can be used, in cooperation with a specific programming paradigm, in order to analyse large data sets by using computer clusters connected with wide area high performance networks. In conclusion they compared the performances achieved by Sphere cloud computing with the ones achieved by Hadoop.
The 3rd-generation KDD tools introduce a new concept for mining applications: the *semantics*. With the introduction of this approach it has been tried to remedy to semantic gap subsisting among technical and business view of knowledge discovery process. The syntactic description of data, used to model data structures, was replaced by the semantic description of resources that can be classified as:

- **Computational Resources**: that represent all the tools used to manipulate data in order to transform it in model elements.
- **Datasets**: Examples of facts of a specific application domain;
- **Models**: representing the final model produced at the end by Computational Resources;

Examples of these concepts can be found in various application domains. For example, in [177] authors used a domain ontology-driven approach to data mining on a medical database in order to analyse data about patients affected with chronic kidney disease. They have focused their attention in explores the possibility of utilizing a medical domain ontology as a source of domain knowledge to aid in both extracting knowledge and expressing the extracted knowledge in a useful format for users that are not familiar with the metrics and the attributes used to describe the domain of kidney disease and his treatment. Another example of domain information retrieval during a data mining process' application, is represented by [178] where an ontology-based approach used to help biologists in retrieving informations about their studies by exploiting the existing semantic web technologies in parallel with formalised ontologies, semantic annotations of scientific articles and knowledge extraction from texts.

In addition to domains information retrieval's task, ontologies and semantics can be both useful in choosing the best algorithm to use in order to achieve the predefined goals. In [179] focus is centred in helping users to overcome the complex task of the choose the best algorithms suitable to achieve the prefixed goals and, after that, in composing them in a complex mining process. In order to perform this task, authors have introduced KDDONTO, an ontology formalizing the domain of KDD algorithms. For the design of KDDONTO they

have exploited a formal ontology building methodology aimed to define goal-oriented ontologies satisfying quality requirements. After defining the above-mentioned methodology, authors have applied it in a real world case study by developing a DL OWL implementation and addressing some existing issues, like the impossibility of representing n-ary relations, that still need to be resolved. Also in [180] an ontology for representing the knowledge discovery (KD) process based on the CRISP-DM process model is presented and discussed in order to define the most important concepts used in the context of KD in a two-layered ontological structure.

The supervised learning in Building Detection

The latest KDD technologies tools have been used also in the automatic building detection domain. The problem of building detection is important to solve several applicative problems. The map updating, 3D city modelling, road centerline detection [181], change detection [182, 183, 184] benefit of automated approaches to detect building. An interesting review of the current trends of photogrammetry is described in [185] that focuses also on the state of the art of 3D building extraction from monocular, stereo, panchromatic or multi-spectral data. Building extraction requires dense and accurate data. The data density plays a key role and sensor manufacturer are providing more effective solution to increase the detection / extraction accuracy also in presence of small buildings or buildings partially covered by trees.

The problem of building detection belongs to a more complex problem that is represented by the object extraction in urban environments [186, 187] or 3D reconstruction [188, 189, 190, 191] where often it is necessary to extract roofs to enhance the classification of building type [192]. There are many challenges related to the object detection, but buildings are commonly the main source of interest for researchers and city managers [193, 194, 195, 196].

The use of heterogeneous data is the key success for an accurate and precise building extraction. Several authors have combined high resolution multi-spectral and LIDAR data. Zeng et al. [197] show an improvement in the classification of IKONOS imagery when integrated with LIDAR while the work of [198] underlines how this integration makes the object-oriented classification superior to maximum likelihood in terms of reducing salt and pepper. In [199] authors combine LIDAR elevation data and SPOT5 multi-spectral data for the classification of urban areas using Support Vector Machine (SVM) algorithm while the approach described in [200] combines QuickBird imagery with LiDAR data for object-based forest species classification. However the above mentioned approaches are based on the high-resolution images from satellite platform. Other approaches are based on the integration of aerial imagery as described in [201, 202].

The manually detection of building data/photo-interpretation is the classical approach. This technique relies on manually based workflow that requirer domain knowledge, time and money. Obviously automated approaches increase the productivity owing to a strongly reduced time to process data. The automation allows an easy integration of multi-source aerial data that is necessary to obtain excellent performance in terms of precision and accuracy. Multi source usually means multi-spectral and LiDAR data obtained by aerial surveys (also by using low cost platform as Unmanned Aerial Vehicles) [203, 204, 205].

There is a dichotomy in the class of methods: pixel based vs object based. It is out of scope of this paper the detailed analysis of pros and cons of the abovementioned approaches. However, it is possible to integrate the pros and cons of pixel and object based approaches with or without training sets (supervised and unsupervised) by using hybrid algorithms as described in [206, 207, 208]. Many approaches has been presented to solve the problem of automated detection of buildings as Bayesian maximum likelihood [209], Dempster-Shafer [210, 154] theory of evidence, AdaBoost [206], SVM [211, 212], height threshold to a normalized DSM, Nearest Neighbour [154]. A supervised approach requires a training set that is usually created by a domain expert. The training set definition is often a critical aspect that strongly influences the overall performance.

As a matter of facts, in the definition of the training set is important to select the most representative sample for each class. Often the definition of the training area is not accurate and should contain objects that belongs to other classes. Another classical problem is the number of collected samples and their coverage over the overall area of study (usually expressed as ratio of area of collected samples over the overall area). The main consequence is the unbalancing of the dataset, which is a critical problem for many supervised approaches [213]. In our case the minority class is represented by the buildings. Even if the coverage of buildings over the overall area is high the variety of roofs [214], height, shape is hard to capture and should be costly [215].

On these datasets, traditional learning techniques tend to overlook the less numerous classes, at the advantage of the majority class. However, the minority class is often the most interesting one for the task. For this reason, the class-imbalance problem has received increasing attention in the last few years [216, 217]. In order to handle imbalanced datasets, the reference model is represented by the Bayes Vector Quantizer (BVQ) algorithm [218].

The BVQ is a multi-class learning algorithm allowing to adapt the (nearest neighbor) decision rule defined by a Labelled Vector Quantizer (LVQ) toward the optimal Bayes decision rule. It has been demonstrated that the behaviour of this algorithm on imbalanced data sets allows obtaining better classification performances compared to the principal methods proposed in the literature

[217].

4.3 Background

This section introduces concepts that are the basis of the BVQ algorithm. First, definitions from the statistical decision theory are provided. Then, Vector Quantizers are introduced and the Bayesian Vector Quantizer algorithm is described.

4.3.1 Statistical Decision Theory

Let boldface characters denote random variables, and let $(\mathbf{x}, \mathbf{c})$ be a random variable pair taking values from $\mathcal{R}^d \times \mathcal{C}$, where the continuous random vector $\mathbf{x}$ is the *feature* vector, while the discrete random variable $\mathbf{c} \in \mathcal{C} = \{c_1, c_2, \ldots, c_C\}$ is its *class*. Classes are statistically characterized by conditional probability density functions (cpdf) $p_{\mathbf{x}|c}(x|c_i)$, measuring the probability that $\mathbf{x} = x$ given that the class observed is $\mathbf{c} = c_i$. Also, a priori probabilities $P_{\mathbf{c}}(c_i)$, $i = 1, ..., C$ are given. From cpdf, class a-posteriori probabilities can be derived by the Bayes theorem:

$$P_{\mathbf{c}|x}(c_i|x) = \frac{p_{\mathbf{x}|c}(x|c_i) \cdot P_{\mathbf{c}}(c_i)}{p_{\mathbf{x}}(x)}, \tag{4.1}$$

where $p_{\mathbf{x}}(x)$ is the probability density function of the random vector $\mathbf{x}$

$$p_{\mathbf{x}}(x) = \sum_{i=1}^{C} p_{\mathbf{x}|c}(x|c_i) \cdot P_{\mathbf{c}}(c_i). \tag{4.2}$$

In order to ease notation, boldface subscripts will be dropped when this will not make confusion among random variables. A classification rule Φ can be defined as a mapping:

$$\Phi : \mathcal{R}^d \to \mathcal{C}, \tag{4.3}$$

where $\Phi(x) \in \mathcal{C}$ denotes the decision taken according to Φ when $\mathbf{x} = x$ is observed.

The feature space is partitioned by using the rule Φ in C *decision regions*, which can take various forms even concave and disjoint. Boundaries between decision regions are called *decision boundaries*.

To evaluate the performance of a classification rule, the *average misclassification risk* is typically used, as follows:

$$R(\Phi) = \int R(\Phi(x)|x) p(x) dV_x, \tag{4.4}$$

where dV_x is used to denote the differential volume in the feature space, and

$R(\Phi(x)|x)$ is the *conditional risk* (or cost) of deciding in favor of class $\Phi(x) \in \mathcal{C}$ given that $\mathbf{x} = x$ is observed. The conditional risk for the class $c_i \in \mathcal{C}$ is given by

$$R(c_i|x) = \sum_{j=1}^{C} b(c_i, c_j) P_{\mathbf{c}|x}(c_j|x). \tag{4.5}$$

where the element $b(c_i, c_j)$ is a component of the cost matrix B, representing the cost of deciding in favour of class c_i when the true class is c_j. If $b(c_i, c_j) = 1$ for $i \neq j$ and $b(c_i, c_i) = 0$, then average misclassification risk (or, in short, average risk) turns to the well known 0-1 loss, or error probability that is 1 minus the accuracy achieved by the classifier. In this work, the cost matrix is composed by elements calculated with $b(c_i, c_j) > 0$, for $i \neq j$ and $b(c_i, c_i) = 0$.

The optimal decision rule, that is the one which minimizes the average misclassification risk, is the Bayes rule:

$$\Phi_B(x) = arg \min_{c \in \mathcal{C}} R(c|x). \tag{4.6}$$

Unfortunately, the use of the Bayes rule in applications of practical interest is limited since class distributions are usually unknown, and can be estimated by using sample sets (i.e., training sets). To this end, parametric or non-parametric approach can be used. In this thesis a non-parametric approach is adopted, where probabilities are estimated by means of the size of the sample set. In this scenario, in [217], authors demonstrate that, for (strongly) imbalanced two classes problems a good estimation of the element $b(c_i, c_j)$, for $i \neq j$, is:

$$b(c_i, c_j) = \frac{N_j}{N} \tag{4.7}$$

where N_j is the number of samples belonging to class c_j and N is the total number of samples. Hence the cost matrix B for a two classes problem will have the following form:

$$B = \begin{pmatrix} 0 & \frac{N_2}{N} \\ \frac{N_1}{N} & 0 \end{pmatrix} \tag{4.8}$$

4.3.2 Nearest Neighbor Vector Quantizer

A nearest neighbor Vector Quantizer (VQ) with Euclidean distance is a mapping:

$$\Omega : \mathcal{R}^d \rightarrow \mathcal{M}, \tag{4.9}$$

$\mathcal{M} = \{m_1, m_2, \ldots, m_M\}$, $m_i \in \mathcal{R}^d, m_i \neq m_j$, which defines a partition of $\mathcal{R}^d$ into M regions $\{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_M\}$, such that

$$\mathcal{V}_i = \{x \in \mathcal{R}^d : \|x - m_i\|^2 \leq \|x - m_j\|^2, \forall i \neq j\}. \tag{4.10}$$

M is the size of the VQ. $\mathcal{M}$ is called the *codebook*, m_i is called *code vector* and $\mathcal{V}_i$ is called the *Voronoi region* of code vector m_i. The border of a Voronoi region is represented by the intersection of a finite set of hyperplanes.

The VQ architecture can be used in the classification task, equipping it by a further mapping which assigns a label from $\mathcal{C}$ to each code vector. This is called extended VQ a *Labeled Vector Quantizer (LVQ)*.

A Labeled VQ is a pair $< \Omega, \Lambda >$, where Ω is a VQ and Λ is mapping Λ: $\mathcal{M} \rightarrow \mathcal{C}$ assigning a label $l_i \in \mathcal{C}$ to each code vector m_i.

The composition of Ω and Λ is a mapping from $\mathcal{R}^d$ to $\mathcal{C}$, that is a decision rule. The decision taken by the LVQ when $\mathbf{x} = x$ is presented as input is

$$\Lambda\big(\Omega(x)\big) = l_i, \quad \text{if} \quad x \in \mathcal{V}_i. \tag{4.11}$$

In practice, the classification task is performed by finding in $\mathcal{M}$ the code vector at minimum distance from x, and then by declaring the label of the code vector. Under this decision rule, a class $c_i \in \mathcal{C}$ is represented by the set of regions $\mathcal{V}_j$ that have the same label $l_j = c_i$.

When a LVQ is used, equation 4.4 becomes simpler. In particular, for a two-class problem and assuming $b(l_i, c_j) = 0$ for $l_i = c_j$, the formula becomes:

$$\begin{aligned} R\big(\Lambda(\Omega)\big) \quad = \quad & \sum_{i:l_i=c_2} b(c_2, c_1) \int_{\mathcal{V}_i} P(c_1|x)p(x)dV_x + \\ & \sum_{i:l_i=c_1} b(c_1, c_2) \int_{\mathcal{V}_i} P(c_2|x)p(x)dV_x, \end{aligned} \tag{4.12}$$

Note that average risk depends on the labeled partition that, assuming the labeling function fixed, depends only on the mutual position of labeled code vectors. Then, by modifying the position of code vectors one can modify the decision rule, trying to reduce the average risk. A principled way to do this is to adopt gradient descent techniques.

4.3.3 The Bayesian Vector Quantizer Algorithm (BVQ)

The Bayesian Vector Quantizer (BVQ) is a non-parametric algorithm, which implements a gradient descent technique to minimize the average misclassification risk performed by a LVQ. The interested reader can find details on the derivation of the algorithm in [218]. Table 4.1 sums up the steps of the BVQ

Table 4.1: The BVQ Algorithm.

1. Set the values of M, Δ, $\gamma^{(0)}$, and the number of iterations n_{max};
2. Set the initial code vectors $m_1, \ldots, m_M$;
3. For k=1 to n_{max} do
   (a) randomly pick a training pair $(t^{(k)}, u^{(k)})$ from the training set;
   (b) find the code vectors $m_i^{(k)}$ and $m_j^{(k)}$ nearest to $t^{(k)}$;
   (c) $m_\lambda^{(k+1)} = m_\lambda^{(k)} \quad for \;\; \lambda \neq i, j$;
   (d) calculate $\pi_{ij}(t^{(k)})$ as in (16);
   (e) if $\| \, t^{(k)} - \pi_{ij}(t^{(k)}) \, \| \leq \frac{\Delta}{2}$ then

$$\beta = \frac{b(l_j, u^{(k)}) - b(l_i, u^{(k)})}{\Delta \, \| \, m_i - m_j \, \|};$$

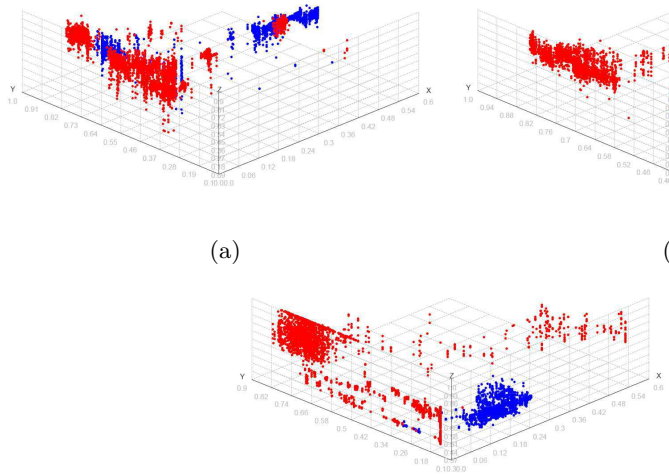
$$m_i^{(k+1)} = m_i^{(k)} - \gamma^{(k)} \beta (m_i^{(k)} - \pi_{ij}(t^{(k)}));$$

$$m_j^{(k+1)} = m_j^{(k)} + \gamma^{(k)} \beta (m_j^{(k)} - \pi_{ij}(t^{(k)}));$$

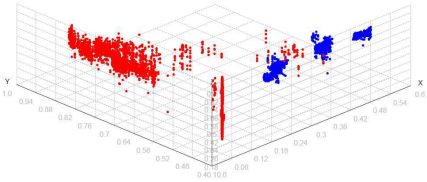
   else $m_\lambda^{(k+1)} = m_\lambda^{(k)} \quad for \;\; \lambda = i, j$

algorithm.

At each iteration, the algorithm randomly picks a training sample from the training set. If this sample *"falls"* close the decision border, the two code vectors determining the border are updated, moving the code vector with the same label of the training sample toward the sample itself. Instead, the code vector with a different label of the training sample is *"punished"* and moved away from the sample itself. Since the border is infinitely small on the feature space, the training sample has 0 probability to fall on it; for this reason an approximation of the decision border is made, and all samples falling into an hypercubic window Δ are considered. $\gamma^{(k)}$ is the learning rate such that $\gamma^{(k)} = \gamma^{(0)} k^{-r}$.



(a)



(b)

(c)

Figure 4.1: Distribution of training samples in domain space considering only δp, δh and NDVI as features. (a) Mannheim1 case study; (b) Mannheim2 case study; (c) Memmingen case study. Blue and red dots represent building and non-building class samples respectively.

4.4 Experimental setup

4.4.1 Datasets

In this subsection, details about the three datasets used in the experiments are provided. They are extracted by the high resolution LiDAR and multi-spectral dataset [154], which is composed by two distinct areas located in Germany:

- Mannheim Area;
- Memmingen Area.

For each pixel in the image (i.e., a data sample) of the related area (1808x1452 pixels for Mannheim and 577x789 pixels for Memmingen), the following seven features are considered:

1. δh: the height difference between the last pulse DSM and the DTM;
2. δp: the height difference between the first pulse and the last pulse DSM;
3. Normalized Difference Vegetation Index (NDVI): the difference between the red and near-infrared channels;

4. Green (G), Blue (B), Red (R) and Near Infrared (Near InfraRed): the features describing the ortho-images that have a resolution of 0.5 m. in Memmingen and 0.25 m. in Mannheim.

The LiDAR samples are based on the first echo DSM and a last echo DSM. The point spacing for the Memmingen was 1.0 m. while in Mannheim was 0.5 m.

The Mannheim urban study area is characterized by large building blocks with cars and vegetation, while Memmingen study area is composed by seventy small buildings surrounded by car boxes and a modest size forest with several tree characterized by a reduced canopy density. In particular a structure larger than 15-30 m2 with a height of at least 2.5 m is considered a building. In the Memmingen dataset the building class covers 14%, while in Mannheim the 37% of the whole study area is covered by buildings.

More details about the datasets are available in [154]. Four main classes are considered: building, tree, land and grass. The main focus was on building extraction by considering building versus tree, land and grass. These classes are the most representative of the objects that are in the study area. Not considered classes are treated as a source of noise even if their effect is not meaningful. This approach is similar to the problem of Land Use / Land Cover (LU/LC) classification with a Corine Land Cover (CLC) legend. The first level contains a limited number of land cover classes that are composed by sub-classes with strongly different spectral data. However, in our case, the problem is simpler than the LU / LC case whereas the chosen classes cover almost the totality of the study area. The building extraction is fundamental for a correct set-up of LU/LC map, especially for urban areas [219].

In order to define the training set, homogeneous sub-areas have been first extracted from the original image and then labelled by an expert as one of the four classes. The definition of a precise and accurate training set is fundamental to ensure a correct performance evaluation of algorithms. Two training sets for the Mannheim area, i.e. Mannheim1 and Mannheim2, are used. The former is more challenging considering that the samples have been collected with a lot of noise (presence of other classes, i.e. the definition of sub-areas is not accurate). The main reason was to stress the algorithms also in presence of noisy training samples that represents a typical scenario when the training set, created by a user, is affected by errors. The latter dataset has been derived in a rigorous way by an accurate sampling, minimizing the influence of other classes. In the Memmingen area, one dataset (Memmingen) in a rigorous way was derived. Mannheim1 and Mannheim2 training sets are composed by 10082 samples, which represent the 0.38% of the image; whereas Memmingen training set is formed by 8303 samples (the 1.82% of the image).
As test set, the whole images have been used.

Since the parameter Δ of the BVQ algorithm represents an hypercubic window, each feature has been normalized in the range [0, 1], so to give equal importance to each feature during learning. To this end, the following normalization formula has been applied:

$$x'_i = \frac{x_i - min_i}{max_i - min_i} \tag{4.13}$$

where x_i is the original value of the feature, x'_i is the normalized one, max_i and min_i are, respectively, the maximum and the minimum value of the feature i.

In order to deal with building detection problem, a further data transformation was performed: classes representing trees, grass and land have been relabelled as class that represents non-building samples. Hence, two-class problems were obtained:

- Class 1 representing building samples;
- Class 2 representing non-building samples: tree, grass and land.

Figure 4.1 shows the distribution of all training samples in the feature space, with respect to the first three features (δp, δh and NDVI).

All datasets represent class-imbalance problem. In details, Mannheim and Memmingen have 68.37% and 80.77% of samples in non-building class respectively. In order to demonstrate the effectiveness of BVQ in dealing with strongly imbalanced problems, the original datasets were imbalanced by randomly removing samples of the building class from the training sets. Hence, 3 training sets were defined for each dataset, as follows:

- Mannheim1_100, Mannheim2_100 and Memmingen_100 with all the training samples of building class;
- Mannheim1_50, Mannheim2_50 and Memmingen_50 with 50% of training samples of building class;
- Mannheim1_10, Mannheim2_10 and Memmingen_10 with 10% of training samples of building class.

4.4.2 Performance Evaluation Metrics

To perform an accurate evaluation of the algorithms, the following quantities will be used:

- TP (True Positive) represents the number of pixels correctly classified as building;

- FP (False Positive) represents the number of pixels not correctly classified as building;
- FN (False Negative) represents the number of pixels belonging to building class that the classifier assigned to a different class;
- TN (True Negative) represents the number of pixels not belonging to building class that the classifier has correctly classified;
- UP (Unclassified Positive) represents the number of pixels belonging to building class that the classifier has not classified;
- UN (Unclassified Negative) representing the number of pixels not belonging to class c not classified.

From the above quantities, several metrics can be defined [220, 221]. In this work, the same metrics as in [154] are adopted, as follows:

- Overall Accuracy defined as:

$$OA = \frac{TP + TN}{TP + FP + TN + FN}. \tag{4.14}$$

  When the cost matrix is equivalent to the identity matrix, the average misclassification risk is given as $1 - OA$;

- Detection Rate defined as:

$$DR = \frac{TP}{TP + FN + UP}; \tag{4.15}$$

- Reliability defined as:

$$R = \frac{TP}{TP + FP}; \tag{4.16}$$

- False Negative Rate defined as:

$$FNR = \frac{FN}{TP + FN + UP}; \tag{4.17}$$

- False Positive Rate defined as:

$$FPR = \frac{FP}{TN + FP + UN}; \tag{4.18}$$

- Unclassified Positive Rate defined as:

$$UPR = \frac{UP}{TP + FN + UP}; \tag{4.19}$$

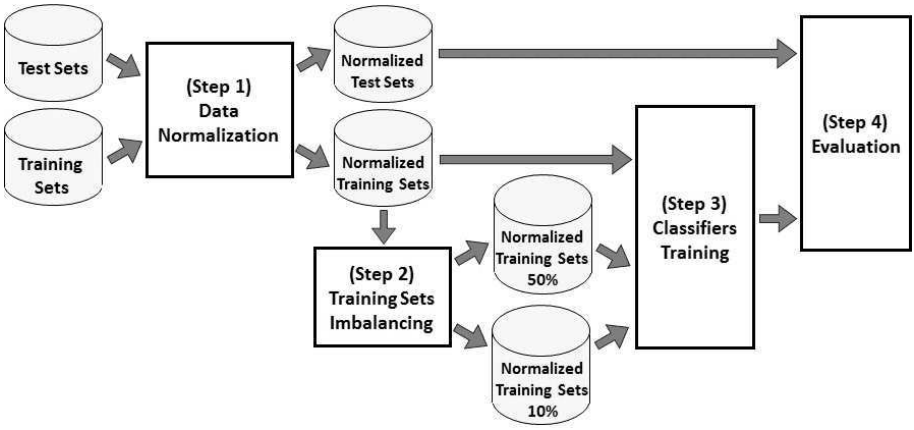


Figure 4.2: The Procedure adopted for experiments performed.

- Total Unclassified Rate defined as:

$$TUR = \frac{UP + UN}{TP + FP + FN + TN + UP + UN}; \tag{4.20}$$

4.4.3 Procedure And Parameters Setup

Figure 4.2 shows the main steps followed in this work to perform experiments. In the initials two steps of the procedure (Step 1 and Step 2), the data preprocessing activities described in the previous sub-sections are performed.

At the end of Step 2 nine different training sets are present; three for each dataset corresponding to nearly balanced problems (with 100% of building class samples) and strongly imbalanced ones (with 50% and 10% of building class samples).
By comparing the results achieved by the BVQ classifier with the performances obtained by other algorithms commonly employed in classification tasks. This comparison is made in order to carry out a full assessment of the performances obtained by the BVQ algorithm. The evaluation is made by considering the following classifiers:

- Key-Nearest Neighbors (K-NN) [222];
- MetaCost [223];
- Weka J48 [224];
- AdaBoost (Gentle) [225];
- AdaBoost (Real) [226].

As concerns Step 3, several experiments were performed by varying the algorithms parameters. Table 4.2 shows the parameters configurations used to train the classifiers during experimentation. For each algorithm, the same parameters configuration was applied on all datasets considered. In this thesis two different type of classifiers are taken into account: Cost-Sensitive and Non Cost-Sensitive classifiers.
In first place, the configuration parameters used to set up the non Cost-Sensitive classifiers were analysed in detailed way, in which the configuration of the cost matrix B (4.8) is not required.
Observing the Table 4.2, the non Cost-Sensitive algorithms are: K-NN, AdaBoost (Real and Gentle versions) and Weka J48.

By considering K-NN algorithm, the k parameter was configured in the range $[1, 100]$ with a step size of 10. The distance measure used is the Euclidean Distance that is represented as follow:

$$d(p, q) = \sum_{i=1}^{N} \sqrt{(q_i - p_i)^2} \tag{4.21}$$

where $p = (p1, p2, ..., pn)$ and $q = (q1, q2, ..., qn)$ represents two points in the Euclidean n-space.

Weka J48 decision tree is a Weka implementation of the well-known Quinlan algorithm (C4.5) [224]. The number of training iterations i varies within the range $[10, 20, 30, 35, 40, 50]$. The *"Confidence Threshold"* parameter c, representing the minimum confidence value below which, the corresponding subtree, is pruned, was made to vary among the range 0.05 and 0.50 with a step size of 0.05. This interval represents a good confidence threshold compromise between algorithm accuracy and training overfitting. The minimum number of instances per leaf m varies in the range $[1, 3]$ while the flag *"unpruned tree"* is set to false.

The lasts non Cost-Sensitive algorithms considered, are the Real and Gentle AdaBoost algorithms. Real AdaBoost [226] is the generalization of a basic AdaBoost algorithm presented in [227]. The Gentle approach is showed in [225] and, generally, obtain better performances than Real version. The only parameter required by AdaBoost algorithms, is represented by the iterations parameter i. Several tests on the above mentioned datasets were performed by varying the i parameter between the interval $[10, 50]$ with a variable step size. After 35 iterations a saturation of the overall accuracy is obtained with a subsequent reduction of computational performance. A set of binary classifiers

to manage multi-class problems were adopted. A pixel (sample) is assumed to be correctly classified if the binary classifier returns a positive result; otherwise, it is recursively passed to the next element in the sequence.

The Cost-Sensitive algorithms used in this work are: MetaCost and BVQ. Those algorithms requires the set-up of the costs matrix B for each different training set. In order to perform this task, the formula in (4.7) has been used, obtaining the following matrices:

- Mannheim1 and Mannheim2 with all building training samples

$$B_{man100} = \begin{pmatrix} 0 & 0.6837 \\ 0.3163 & 0 \end{pmatrix}$$

- Mannheim1 and Mannheim2 with only 50% of building training samples

$$B_{man50} = \begin{pmatrix} 0 & 0.8122 \\ 0.1878 & 0 \end{pmatrix}$$

- Mannheim1 and Mannheim2 with only 10% of building training samples

$$B_{man10} = \begin{pmatrix} 0 & 0.9558 \\ 0.0442 & 0 \end{pmatrix}$$

- Memmingen dataset with all building training samples

$$B_{men100} = \begin{pmatrix} 0 & 0.8077 \\ 0.1922 & 0 \end{pmatrix}$$

- Memmingen dataset with only 50% of building training samples

$$B_{men50} = \begin{pmatrix} 0 & 0.8937 \\ 0.1063 & 0 \end{pmatrix}$$

- Memmingen dataset with only 10% of building training samples

$$B_{men10} = \begin{pmatrix} 0 & 0.9767 \\ 0.0233 & 0 \end{pmatrix}$$

MetaCost algortihm uses an underlying classifier as a black box [223], making it Cost Sensitive. In our case the weak classifier is the aforementioned Weka J-48 with the same configuration parameters explained before. The training set

was considered entirely, during the learning phase, by setting the *"Size of Bag"* parameter to 100. As for AdaBoost and Weka J-48 algorithms the iterations parameter i was set in varying in the range $[10, 50]$ with a variable step size.

As concern BVQ algorithm, the number of training iterations i was fixed to 40000 iterations in such a way that, in the learning phase of the algorithm, most of the pixel in the training sets are considered more times. By observing Figure 4.1, it is evident that for Memmingen the building class can be easily represented by an unique cluster of sample, while for Mannheim it is disjoint into at least three clusters; this is particularly evident in Mannheim2. For this reason, in order to generalize the parameters configuration, 2, 4 and 8 code vectors were used for every case study considered. The window Δ was made to vary within the range $[0.01, 0.3]$ with a step size of 0.01 while the learning rate parameter $\gamma^{(0)}$ varies in the range $[0.5, 2.5]$ with a step size of 0.1. This range has been selected by considering that, higher the imbalance rate is, greater the gap between the values of the cost matrix is. Thus, in problem with high imbalance rate, in order to have a significant update of code vectors, a higher value of $\gamma^{(0)}$ were needed.
In order to prove the effectiveness of the BVQ algorithm, in addition to consider the best One Shot result obtained, each experiment was repeated 5 times by varying the random seed used to randomly pick up training samples during the learning phase of the algorithms. For each seed, the metrics defined in 4.4.2 have been computed and average values are reported as outcome (Step 4).

The BVQ results will be discussed in the next Section and compared with the results achieved by using the other algorithms considered in this work and applied on the same datasets. It is noteworthy that, for each algorithm, a cleaning technique has been applied to the resulting image in order to remove small set of pixels from the classified image, which in most cases corresponded to errors in the acquired data [209].

4.5 Results

Tables 4.3, 4.4 and 4.5 show the results obtained on Mannheim1, Mannheim2 and Memmingen datasets using BVQ, AdaBoost (Gentle and Real), K-NN, W-J48 and MetaCost algorithms. In the Tables, are represented the best outcomes obtained for the algorithms selected.
Figures 4.3, 4.4 and 4.5 show graphically the results of the classification obtained by the considered classifiers. Figures refer to the seed with returns the best overall accuracy.

Table 4.2: Parameters used in the experiments.

Algorithm	Cost-Sensitive	Parameters
K-NN	No	$k = [1; 100]$ step 10 Distance = Euclidean Distance
MetaCost	Yes	$i = [10, 20, 30, 35, 40]$ Size of Bag = 100% weak classifier = Weka J48
Weka J48	No	$i = [10, 20, 30, 35, 40]$ $c = 0.25$ $m = [1; 3]$ step 1 unpruned tree = false
AdaBoost (Real & Gentle)	No	$i = [10, 20, 30, 35, 40]$
BVQ (Seed & One Shot)	Yes	$i = [40000]$ Code Vectors=$[2, 4, 8]$ $\Delta = [0.01; 0.3]$ step 0.01 $\gamma^0 = [0.5; 2.5]$ step 0.1

Table 4.3: Obtained results with Mannheim1 dataset with 100%, 50% and 10% of building samples in the training set.

		Mannheim 1						
Building Samples	Algorithm	OA	DR	R	FNR	FPR	UPR	TUR
100%	BVQ (One Shot)	92.96%	85.21%	95.30%	14.79%	2.48%	0%	0%
	BVQ (5 Seeds)	90.96%	79.27%	95.61%	20.73%	2.16%	0%	0%
	K-NN	84.71%	74.41%	82.61%	25.59%	9.22%	0%	0%
	MetaCost	81.71%	71.53%	77.40%	28.47%	12.30%	0%	0%
	Weka J-48	79.87%	72.32%	73.07%	27.68%	15.69%	0%	0%
	AdaBoost (Gentle)	82.93%	64.09%	78.32%	17.56%	10.49%	18.40%	23.12%
	AdaBoost (Real)	88.75%	72.08%	96.71%	27.92%	1.44%	0%	0%
50%	BVQ (One Shot)	93.19%	85.08%	96.09%	14.92%	2.04%	0%	0%
	BVQ (5 Seeds)	90.73%	77.96%	96.34%	22.04%	1.74%	0%	0%
	K-NN	84.24%	64.70%	89.94%	35.30%	4.26%	0%	0%
	MetaCost	79.32%	68.40%	73.86%	31.60%	14.26%	0%	0%
	Weka J-48	80.60%	69.72%	75.95%	30.28%	12.99%	0%	0%
	AdaBoost (Gentle)	83.10%	61.96%	80.64%	20.22%	8.80%	17.86%	22.80%
	AdaBoost (Real)	88.84%	71.93%	97.23%	28.07%	1.21%	0%	0%
10%	BVQ (One Shot)	93.03%	85.27%	95.43%	14.73%	2.40%	0%	0%
	BVQ (5 Seeds)	91.14%	79.55%	95.84%	20.45%	2.04%	0%	0%
	K-NN	72.77%	32.40%	84.65%	67.60%	3.46%	0%	0%
	MetaCost	83.32%	66.73%	85.04%	33.27%	6.91%	0%	0%
	Weka J-48	81.78%	62.78%	84.03%	37.22%	7.02%	0%	0%
	AdaBoost (Gentle)	82.55%	44.20%	92.98%	32.61%	1.98%	23.26%	23.40%
	AdaBoost (Real)	70.37%	20.10%	99.65%	79.90%	0.50%	0%	0%

Table 4.4: Obtained results for the Mannheim2 dataset with 100%, 50% and 10% of building samples in the training set.

Mannheim 2								
Building Samples	Algorithm	OA	DR	R	FNR	FPR	UPR	TUR
100%	BVQ (One Shot)	95.36%	91.56%	95.73%	8.44%	2.40%	0%	0%
	BVQ (5 Seeds)	95.05%	89.66%	96.74%	10.34%	1.78%	0%	0%
	K-NN	94.48%	87.96%	96.85%	12.04%	1.68%	0%	0%
	MetaCost	92.00%	80.87%	97.05%	19.13%	1.45%	0%	0%
	Weka J-48	92.13%	81.28%	97%	18.72%	1.48%	0%	0%
	AdaBoost (Gentle)	95.18%	82.62%	97.45%	9.67%	1.28%	7.73%	8.62%
	AdaBoost (Real)	92.86%	84.02%	96.22%	15.98%	1.94%	0%	0%
50%	BVQ (One Shot)	95.29%	91.09%	95.99%	8.91%	2.24%	0%	0%
	BVQ (5 Seeds)	95.12%	90.41%	96.20%	9.59%	2.11%	0%	0%
	K-NN	94.33%	87.43%	96.96%	12.57%	1.61%	0%	0%
	MetaCost	91.99%	80.80%	97.09%	19.20%	1.43%	0%	0%
	Weka J-48	91.99%	80.80%	97.09%	19.20%	1.43%	0%	0%
	AdaBoost (Gentle)	95.31%	82.67%	97.33%	9.20%	1.34%	8.15%	9.07%
	AdaBoost (Real)	92.24%	82.01%	96.53%	17.99%	1.74%	0%	0%
10%	BVQ (One Shot)	95.26%	91.58%	95.43%	8.42%	2.58%	0%	0%
	BVQ (5 Seeds)	95.02%	89.99%	96.34%	10.01%	2.02%	0%	0%
	K-NN	93.92%	86.03%	97.25%	13.97%	1.43%	0%	0%
	MetaCost	92.56%	83.02%	96.38%	16.98%	1.83%	0%	0%
	Weka J-48	91.31%	78.42%	97.65%	21.58%	1.11%	0%	0%
	AdaBoost (Gentle)	92.58%	87.45%	91.40%	11.14%	4.87%	1.41%	2.95%
	AdaBoost (Real)	92.24%	81.91%	96.66%	18.09%	1.67%	0%	0%

Table 4.5: Obtained results for the Memmingen dataset with 100%, 50% and 10% of building samples in the training set.

Memmingen								
Building Samples	Algorithm	OA	DR	R	FNR	FPR	UPR	TUR
100%	BVQ (One Shot)	96.22%	82.16%	89.93%	22.75%	1.42%	0%	0%
	BVQ (5 Seeds)	95.81%	85.45%	84.70%	22.04%	1.74%	0%	0%
	K-NN	89.68%	90.06%	58.49%	9.94%	10.38%	0%	0%
	MetaCost	95.82%	77.81%	90.97%	22.19%	1.25%	0%	0%
	Weka J-48	95.98%	79.10%	90.93%	20.89%	1.28%	0%	0%
	AdaBoost (Gentle)	95.49%	87.80%	79.48%	6.33%	3.68%	5.87%	10.20%
	AdaBoost (Real)	95.73%	78.09%	90.07%	21.91%	1.40%	0%	0%
50%	BVQ (One Shot)	95.98%	82.13%	88.29%	17.87%	1.75%	0%	0%
	BVQ (5 Seeds)	95.66%	83.32%	85.48%	16.68%	2.34%	0%	0%
	K-NN	90.01%	88.81%	59.58%	11.19%	9.79%	0%	0%
	MetaCost	95.97%	77.54%	92.40%	22.46%	1.04%	0%	0%
	Weka J-48	95.97%	79.10%	90.93%	22.48%	1.03%	0%	0%
	AdaBoost (Gentle)	96.18%	80.57%	83.34%	8.78%	2.62%	10.64%	8.91%
	AdaBoost (Real)	95.64%	74.40%	93.02%	25.60%	0.91%	0%	0%
10%	BVQ (One Shot)	95.98%	83.42%	87.27%	16.58%	1.98%	0%	0%
	BVQ (5 Seeds)	95.39%	84.69%	82.87%	15.31%	2.87%	0%	0%
	K-NN	93.07%	60.75%	85.48%	39.25%	1.68%	0%	0%
	MetaCost	95.89%	77.49%	91.84%	22.51%	1.12%	0%	0%
	Weka J-48	95.90%	77.52%	91.86%	22.48%	1.12%	0%	0%
	AdaBoost (Gentle)	96.79%	79.46%	89.52%	10.99%	1.51%	9.56%	11.78%
	AdaBoost (Real)	95.70%	73.60%	94.30%	26.40%	0.72%	0%	0%

4.5.1 Mannheim Datasets Results

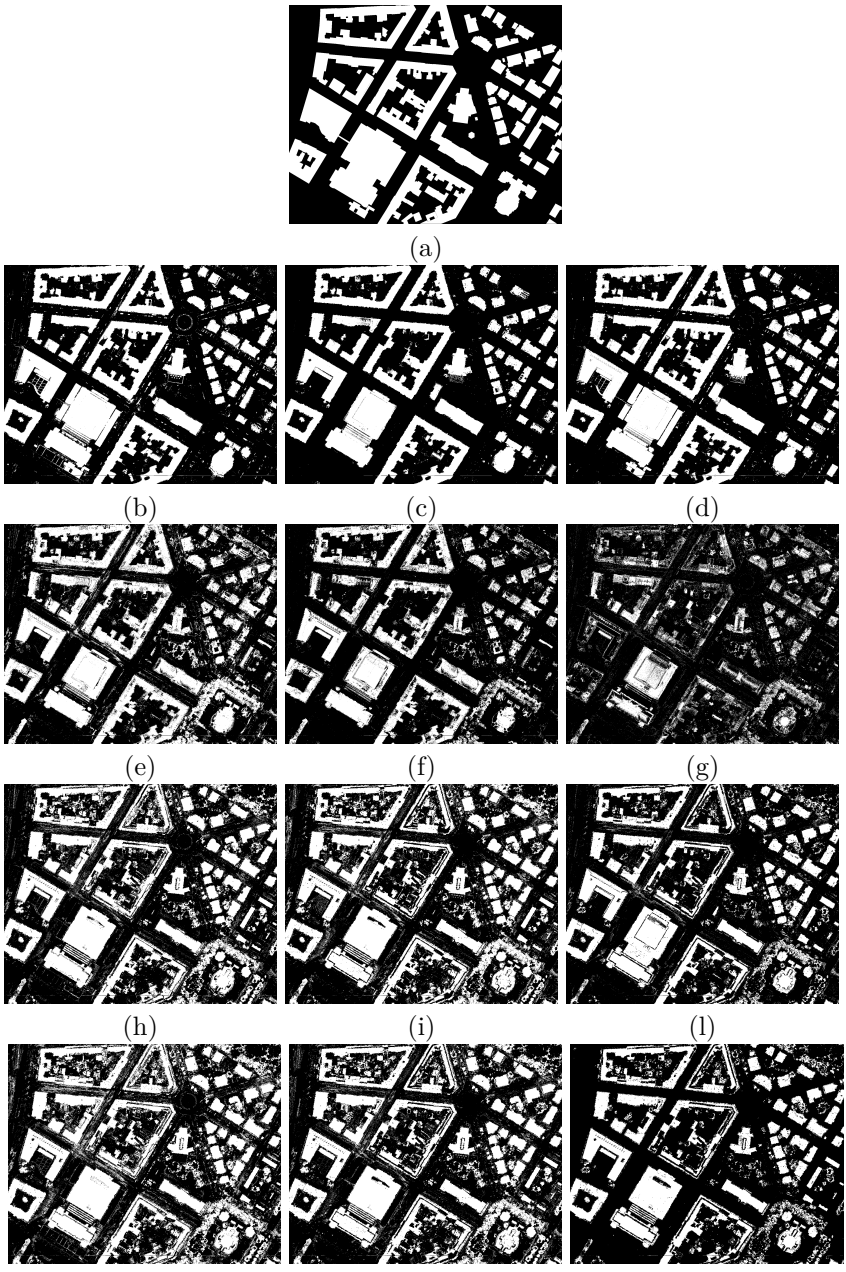
As shown in Table 4.3, BVQ outperforms all other algorithms on Mannheim1 whatever rate of imbalance is considered. As a matter of facts, considering the One Shot results, BVQ algorithm obtains an improvement in overall accuracy in the range of 10%-13% {92.96%,93.19%,93.03%} on average in relation to other algorithms. The algorithm that best match with BVQ, on average, on Mannheim1 dataset is AdaBoost Gentle that stands around the 83% on Overall Accuracy {82.93%, 83.10%, 82.55%}. Other algorithms achieve slightly lower performances, standing around 80%-81% of Overall Accuracy (Table 4.3). The results obtained by BVQ algorithm averaged over 5 different initialization seeds, are slightly worsen then the ones obtained by BVQ One Shot previously analysed {90.96%, 90.30%, 91.14%}, but are still better then those obtained by K-NN {84.71%,84.24%,72.771%}, AdaBoost Gentle, AdaBoost Real {88.75%,88.84,70.37%}, MetaCost {81.71%,79.32%,83.32%} and Weka J-48 {79.87%,80.60%,81.78%}. The increasing of the imbalance in Mannheim1 dataset demonstrated that, while the results achieved by BVQ algorithm are quite stable, other algorithms suffer strongly imbalance. This can be also appreciated by considering the DR metric, which measures the percentage of correctly classified samples of the building class. The improvement of BVQ is here more evident; BVQ returns a quite constant value for DR by increasing rate of imbalance while the other algorithms worsens significantly DR value. Considering Mannheim1_10 BVQ One Shot returns a DR value of 85.27% which is pretty much the same value obtained in Mannheim1_100 (85.21%). The same happens by considering BVQ Seeds; in Mannheim1_100 the DR value achieved is 79.27% while in the strongly imbalanced case is 79.55%. MetaCost is more affected by BVQ in terms of dataset imbalance; the DR values in Mannheim1_100 is 71.53% while in Mannheim_10 decrease of about 5%, standing at 66.73%. Considering the Non-Cost Sensitive algorithms the difference is more evident; K-NN suffers a DR value decrease of over 40%, moving the DR rate value from 74.41% to 32.40%. AdaBoost Gentle and Real versions return quite different results: AdaBoost Gentle suffers a decrease of the DR value of about 20% (64.09% to 44.20%) while AdaBoost Real version experiences a variation in the DR value of even 50% (72.08% to 20.10%).

Considering Mannheim2 dataset, results of BVQ and other algorithms (Table 4.4), in term of OA, are very similar and higher than Mannheim1; this is expected due to a greater class separability shown in Figure 4.1b. The situation slightly changes on Mannheim2_10, where the OA values of the non-Cost Sensitive algorithms decrease of about 0,5%-1%. The only exception is represented by AdaBoost Gentle algorithm that worsen its classification performances by more than 2.5 percentage points (95.18% to 92.58%). Considering the Cost Sen-

sitive algorithms (BVQ and MetaCost), the OA's values remain quite consistent with the ones obtained in Mannheim2_100 dataset, in particular, MetaCost algorithm marginally improves its performances (92% to 92.56%). As concerns DR, BVQ outperforms all others algorithms for each imbalance rate. As in Mannheim1 case, the DR value of the non-Cost Sensitive algorithms tends to worsen of about 2% with imbalance increasing; the only exception is represented by AdaBoost Gentle algorithm that has an improvement of its DR value (from 82.6% to 87.5%). This can be justified by considering the low number of unclassified positive samples (UPR=1.42%). These results suggest that BVQ works well even in the presence of noise due to manual labeling, which leads to a more difficult classification problem. Indeed, in Mannheim1, both the building class are less defined than in Mannheim2 and the two classes that overlap each other (see Figure 4.1). Comparing BVQ with the other algorithms, the different quality of the building detection, is evident in Figure 4.3, where classification results obtained by BVQ One Shot on Mannheim1 (Figures 4.3(b-d)) are more accurate than results obtained by the others algorithm on the same dataset (Figures 4.3(e-u)). It is noteworthy that all classifiers classify all samples belonging to test set. The only exception is represented by AdaBoost Gentle algorithm that has an high value of unclassified samples (see UPR and TPR in the Tables). Since OA does not take into account unclassified samples, the gap between the AdaBoost Gentle and the others algorithms is wider than the one represented by OA.

4.5.2 Memmingen Dataset Results

Table 4.5 shows that all the algorithms considered return quite similar values of the OA metric on the Memmingen dataset. Indeed, by considering Memmingen_100, the best OA is scored by BVQ One Shot that outperforms all other algorithms only by some tenths of a percentage point, obtaining an OA value of (96.22%). The results obtained by the other algorithms, in fact, are placed in the range between 95.5%-96% (except for K-NN), proving that the classification task represented by this case study is simple to solve for any classifier. This is evident also by considering the high class separability shown in Figure 4.1c. Things get more interesting by considering the strongly imbalanced case study (Memmingen_10). In this case, the best result is obtained by AdaBoost Gentle (96.79%), that returns a value of OA with an improvement of 0.81% with respect to BVQ One Shot (95.98%) and 1.4% by considering BVQ Seeds (95.39%). The results obtained by the other algorithms, show that the imbalance does not affect the OA value, in fact the performance highlighted by OA values remain relatively constant. However, is important to note that, in order to compare the accuracy of the AdaBoost Gentle algorithm with com-



(a) (b) (c) (d) (e) (f) (g) (h) (i) (l) (m) (n) (o)

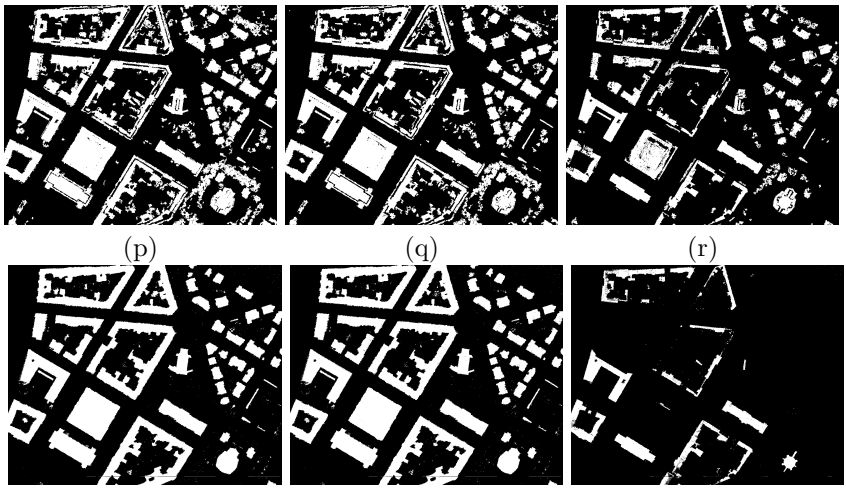
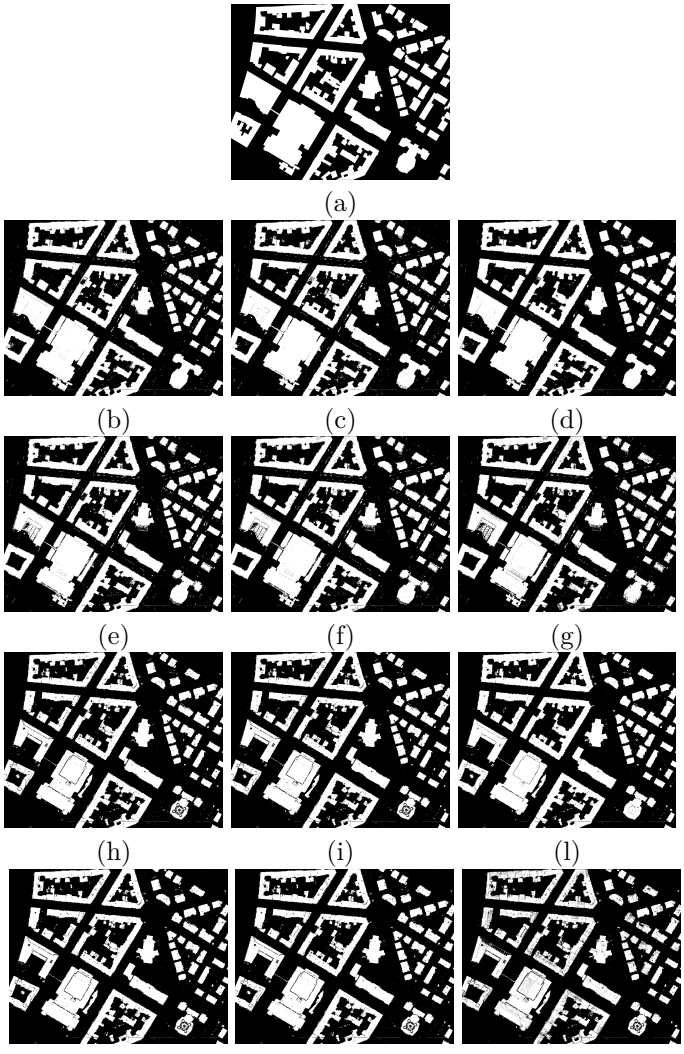


Figure 4.3: Evaluation of the Mannheim1 study area. Building and non-building areas are represented in white and black respectively. (a) Test image used with Mannheim1 and Mannheim2 case studies. (b-d) Best classification results obtained by BVQ One Shot algorithm on (b) Mannheim1_100, (c) Mannheim1_50, (d) Mannheim1_10. (e-g) Best classification results obtained by K-NN algorithm on (e) Mannheim1_100, (f) Mannheim1_50, (g) Mannheim1_10. (h-l) Best classification results obtained by Meta-Cost algorithm on (h) Mannheim1_100, (i) Mannheim1_50, (l) Mannheim1_10. (m-o) Best classification results obtained by Weka J-48 algorithm on (m) Mannheim1_100, (n) Mannheim1_50, (o) Mannheim1_10. (p-r) Best classification results obtained by AdaBoost Gentle algorithm on (p) Mannheim1_100, (q) Mannheim1_50, (r) Mannheim1_10. (s-u) Best classification results obtained by AdaBoost Real algorithm on (s) Mannheim1_100, (t) Mannheim1_50, (u) Mannheim1_10.



(a)

(b) (c) (d)

(e) (f) (g)

(h) (i) (l)

(m) (n) (o)

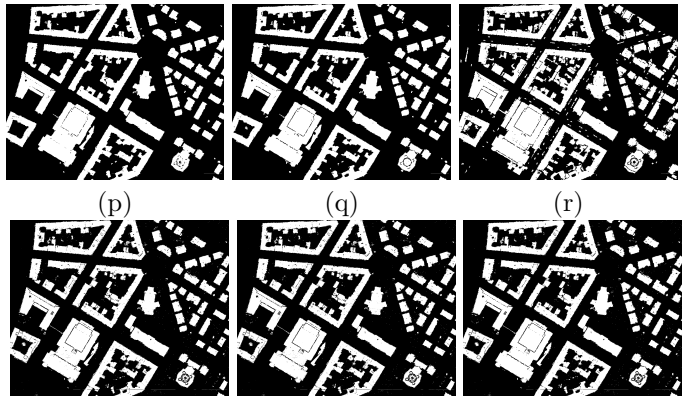
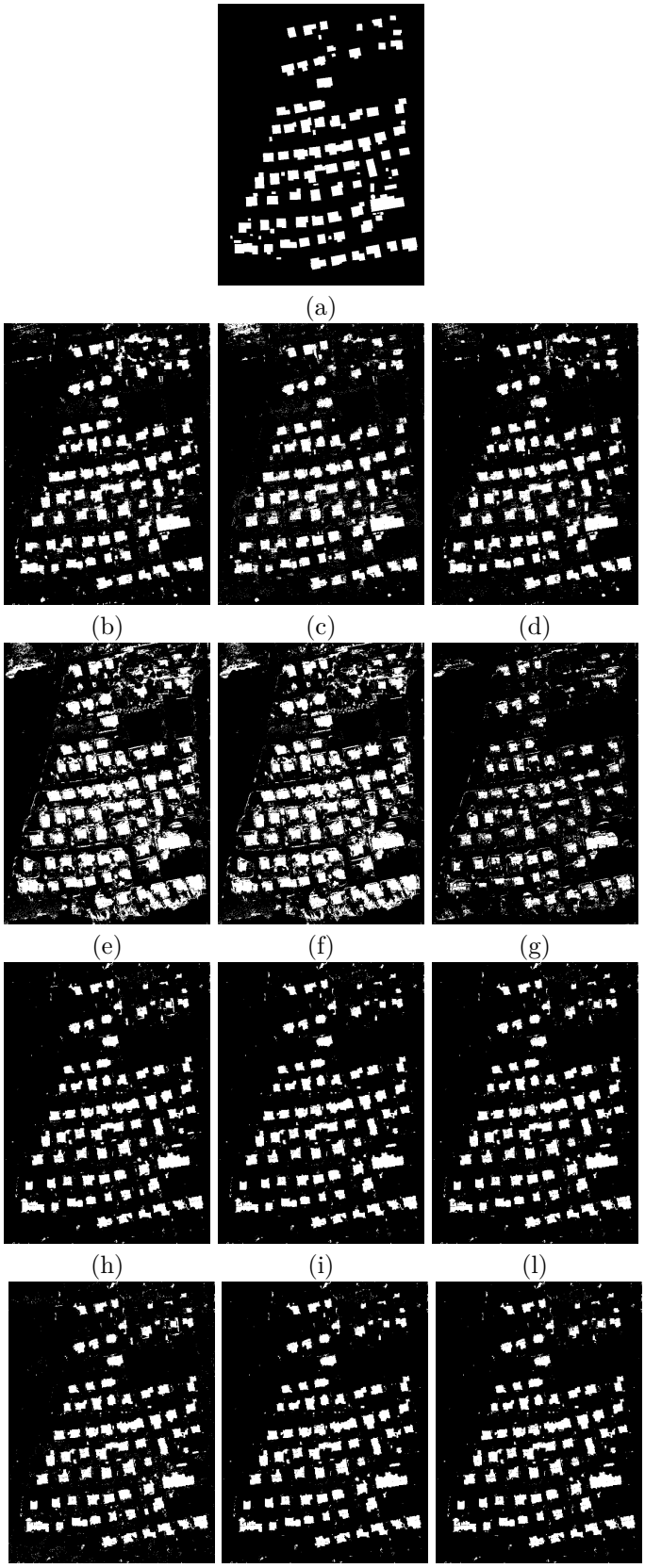


Figure 4.4: Evaluation of the Mannheim2 study area. Building and non-building areas are represented in white and black respectively. (a) Test image used with Mannheim2 case study. (b-d) Best classification results obtained by BVQ One Shot algorithm on (b) Mannheim2_100, (c) Mannheim2_50 and (d) Mannheim2_10. (e-g) Best classification results obtained by K-NN algorithm on (e) Mannheim2_100, (f) Mannheim2_50 and (g) Mannheim2_10. (h-l) Best classification results obtained by MetaCost algorithm on (h) Mannheim2_100, (i) Mannheim2_50 and (l) Mannheim2_10. (m-o) Best classification results obtained by Weka J-48 algorithm on (m) Mannheim2_100, (n) Mannheim2_50 and (o) Mannheim2_10. (p-r) Best classification results obtained by AdaBoost Gentle algorithm on (p) Mannheim2_100, (q) Mannheim2_50 and (r) Mannheim2_10. (s-u) Best classification results obtained by AdaBoost Real algorithm on (s) Mannheim2_100, (t) Mannheim2_50 and (u) Mannheim2_10.



(a)

(b) (c) (d)

(e) (f) (g)

(h) (i) (l)

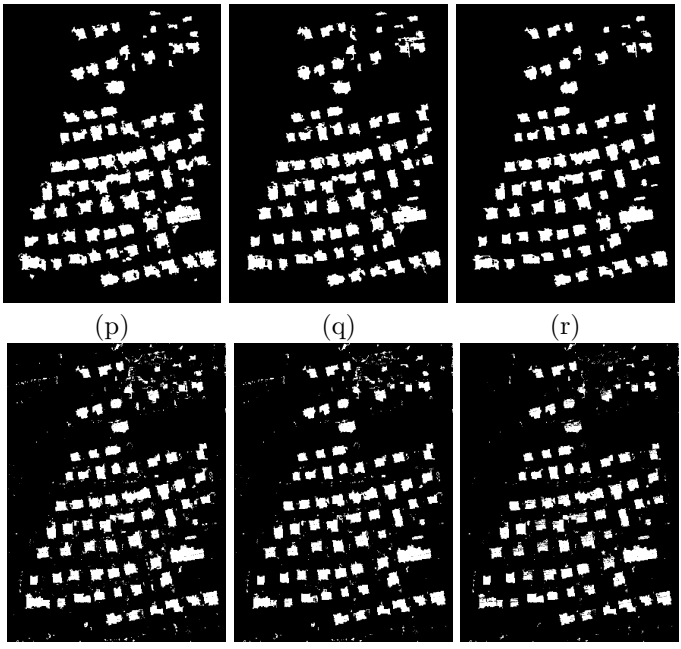


Figure 4.5: Evaluation of the Memmingen study area. Building and non-building areas are represented in white and black respectively. (a) Test image used with Memmingen case study. (b-d) Best classification results obtained by BVQ One Shot algorithm on (b) Memmingen_100, (c) Memmingen_50 and (d) Memmingen_10. (e-g) Best classification results obtained by K-NN algorithm on (e) Memmingen_100, (f) Memmingen_50 and (g) Memmingen_10. (h-l) Best classification results obtained by MetaCost algorithm on (h) Memmingen_100, (i) Memmingen_50 and (l) Memmingen_10. (m-o) Best classification results obtained by Weka J-48 algorithm on (m) Memmingen_100, (n) Memmingen_50 and (o) Memmingen_10. (p-r) Best classification results obtained by AdaBoost Gentle algorithm on (p) Memmingen_100, (q) Memmingen_50 and (r) Memmingen_10. (s-u) Best classification results obtained by AdaBoost Real algorithm on (s) Memmingen_100, (t) Memmingen_50 and (u) Memmingen_10.

petitors, unclassified samples should be also added to the denominator of the OA formula. For example, considering Memmingen_10, AdaBoost Gentle unclassifies 6080 buildings and 47551 samples of the non-building class. Since the number of correctly classified and misclassified samples are 388715 (TP=50549 and TN=338166) and 13907 (FP=5918 and FN=6989) respectively, the accuracy would drop from 96.79% to 85.38%. Considering the detection rate values obtained in Memmingen case study, it is evident that the same situation highlighted in Mannheim datasets it is clearly visible also in this case. The detection rate values obtained by BVQ algorithms are quite constant also in presence of an higher imbalance rate. The detection rate of BVQ One Shot remains around 83% (82.16% in Memmingen_100 and 83.42% in Memmingen_10) while BVQ Seeds stands around 85% (85.45% and 84.69% for Memmingen_100 and Memmingen_10 respectively). MetaCost algorithm also are less influenced by an higher value of the imbalance rate, maintaining the detection rate value around 77.5% for both Memmingen_100 and Memmingen_10 datasets (77.81% and 77.49% respectively). Different results are achieved by Non Cost-Sensitive algorithms. The detection rate values decrease of several percentage points considering K-NN, Weka J-48, AdaBoost Gentle and AdaBoost Real. These findings show the algorithms which take into accounts the imbalance of training sets, obtain better performances in strong imbalanced case studies. To better assess the difference which subsist between the results obtained by the considered algorithms, Figure 4.5 shows the graphical evaluation of the Memmingen study area. It is clear enough that, also in this case, BVQ algorithm has proven to be the most robust to the variation of the rate of imbalance.

Chapter 5

Conclusion and Future Works

In the present thesis, some aspects and problems that affects the Public Transport System in high congested urban areas has been faced and some solutions have been proposed. The first solution proposed is aimed to enhance the performances of Public Transport Systems by adopting an ontology-based framework tailored to support operators in the development of a system for monitoring performances of transport services. Among the existing standard data models for information systems for a PTS, this thesis referred to Transmodel as it represents the reference data model at European level and ensures a wider portability of the proposed solution. The approach is based on a formal, extensible and reusable representation of the knowledge involved in the transport domain, including business objectives, indicators and their formulas, and corresponding relations with Transmodel classes and packages. Although the definition of domain ontologies is typically time-consuming activity, in this case indicators are added in an incremental fashion. Indeed, at setup time only a subset of the ontological instances and relations must be defined, namely Transmodel classes and packages, and links to KPIOnto dimensional hierarchies. This information is not likely to change very frequently. On the other hand, extensibility of the ontology allows to define KPIs at need, by applying the functionalities described in the thesis, which are capable to verify the coherence of the provided information in order to keep the knowledge base always in a consistent state. As such, this also grants re-usability of the core of the ontology for multiple systems, each differing with respect to the set of Transmodel packages actually adopted and the set of defined KPIs. For what concerns logic-based reasoning, in this thesis a set of functionalities capable to manipulate mathematical expressions representing indicator formulas has been exploited. Such logic predicates are able to extend the traditional approaches to KPI management by enabling automatic recognition of equivalence between formulas, and are used also to keep the set of formulas mathematically consistent.

Although a full implementation of the framework is yet to come, at present the most significant components have already been developed, namely the ontology, which is publicly released and available at $http://w3id.org/kpionto$,

and logical functions in the reasoning framework.

The second solution proposed is the implementation of predictive Hybrid algorithms used to predict arrival time at bus stop in Urban Public Transport Systems. In modern society, where time represents one of the most precious resource, avoiding its waste plays a key role in the Quality of Life of people. In this thesis has been demonstrated that Simple Models cannot compete against Hybrid Models that combine historical information on past rides with real-time information obtained by applying Kalman Filtering model on-line GPS signals. In order to evaluate the efficiency of the proposed Hybrid Models, they have been applied to a real-world case study and, more precisely, they have been tested with the data obtained from Line 01 of Olbia's Public Transport System. In order to make the prediction process more usable, Urban Travel Time Prediction Software (UTTP) has been realized. The obtained results have demonstrated that Hybrid Models (ANN/SVM+Kalman Filtering model) outperforms Simple Models (ANN/SVM) and their performances can increase significantly the Quality of the proposed service. Some work can be made to extend the research performed on this thesis. For example some other algorithms can be compared to the ones presented here or other attributes can be added to dataset in order to increase performances of prediction algorithms.

The third aspect faced in this thesis focused on a Bayesian Vector Quantizer approach for building detection with multi-source aerial data. Experiments have proven that the described approach is a reliable and robust alternative method for building detection. In particular, the BVQ algorithm has shown good results even on imbalanced training sets, with a low number of samples belonging to the building class.

Several performed experiments were performed in order to evaluate the performances of BVQ by comparing them with K-NN, MetaCost, Weka J-48 and AdaBoost (Real and Gentle). However, AdaBoost and the other algorithms have never been tested on imbalanced building detection problems.

The experiments demonstrated that the BVQ classification method works better than the other algorithms considered with noisy datasets like Mannheim1. In the other cases (Mannheim2 and Memmingen) the difference in performance with complete training sets is minimal compared to that obtained by other algorithms with particular reference to those performed by the AdaBoost Gentle algorithm. Considering the cases of strongly imbalanced datasets, the difference in performance is more evident of course: for these datasets Non Cost-Sensitive algorithms tend to favour the non-building class that is more representative of the training sets. The BVQ algorithm, instead, tries to take into account the imbalance by considering the misclassification risk instead of the classification error.

Much work can be made to extend the present research, such as the develop-

ment of user interfaces to provide guidance to the execution of the tasks. Future work includes also further extensions of the ontological model for Transmodel, with the purpose to formally characterize all the information included in class diagrams in the official documentation, which may be of help to support more advanced reasoning tasks on the data model. Regarding the Building detection problem, a future development would consider a reduction methodology based on an LVQ classifier [228] in order to define the most discriminative features. It will be also considered the feature ranking or extraction from the original feature space. Regarding the feature ranking the BVQ-FR will be tested [229] while the extraction will be performed by using the BVQ-FE [230, 228].

Bibliography

[1] European Commission, "Traffic safety basic facts on main figures", *Directorate General for Transport*, June, 2016.

[2] United Nations, Department of Economic and Social Affairs, Population Division, *The World's Cities in 2016: Data Booklet*, 2016.

[3] Maria Lindholm and Sönke Behrends, "A holistic approach to challenges in urban freight transport planning", in *General Proceedings of the 12th World Conference on Transport Research Society*, 2010.

[4] Stephen S Lim, Theo Vos, Abraham D Flaxman, Goodarz Danaei, Kenji Shibuya, Heather Adair-Rohani, Mohammad A AlMazroa, Markus Amann, H Ross Anderson, Kathryn G Andrews, et al., "A comparative risk assessment of burden of disease and injury attributable to 67 risk factors and risk factor clusters in 21 regions, 1990-2010: a systematic analysis for the global burden of disease study 2010", *The lancet*, vol. 380, no. 9859, pp. 2224–2260, 2012.

[5] World Health Organization, "Burden of disease from ambient air pollution for 2012 — summary of results", 2015.

[6] World Health Organization et al., "Effects of air pollution on children's health and development: a review of the evidence", 2005.

[7] World Health Organization et al., "Review of evidence on health aspects of air pollution - revihaap project: Technical report", 2013.

[8] D. Stead, *The European Transport White Paper*, vol. Vol. 1, 2001.

[9] Jiawei Han, Jian Pei, and Micheline Kamber, *Data mining: concepts and techniques*, Elsevier, 2011.

[10] Surajit Chaudhuri and Umeshwar Dayal, "An overview of data warehousing and olap technology", *SIGMOD Rec.*, vol. 26, no. 1, pp. 65–74, March 1997.

[11] Panos Vassiliadis, Mokrane Bouzeghoub, and Christoph Quix, "Towards quality-oriented data warehouse usage and evolution", *Information Systems*, vol. 25, no. 2, pp. 89 – 115, 2000.

[12] Ricardo Jorge Santos and Jorge Bernardino, "Real-time data warehouse loading methodology", in *Proceedings of the 2008 International Symposium on Database Engineering & Applications*, New York, NY, USA, 2008, IDEAS '08, pp. 49–58, ACM.

[13] A. Kemper and T. Neumann, "Hyper: A hybrid oltp amp;olap main memory database system based on virtual memory snapshots", in *2011 IEEE 27th International Conference on Data Engineering*, April 2011, pp. 195–206.

[14] C.L. Philip Chen and Chun-Yang Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data", *Information Sciences*, vol. 275, pp. 314 – 347, 2014.

[15] Ibrahim Abaker Targio Hashem, Ibrar Yaqoob, Nor Badrul Anuar, Salimah Mokhtar, Abdullah Gani, and Samee Ullah Khan, "The rise of "big data" on cloud computing: Review and open research issues", *Information Systems*, vol. 47, pp. 98 – 115, 2015.

[16] Wei Fan and Albert Bifet, "Mining big data: Current status, and forecast to the future", *SIGKDD Explor. Newsl.*, vol. 14, no. 2, pp. 1–5, April 2013.

[17] X. Wu, X. Zhu, G. Q. Wu, and W. Ding, "Data mining with big data", *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 97–107, Jan 2014.

[18] Maryam Alavi and Dorothy E Leidner, "Knowledge management and knowledge management systems: Conceptual foundations and research issues", *MIS quarterly*, pp. 107–136, 2001.

[19] Maryam Alavi and Dorothy E. Leidner, "Knowledge management systems: Issues, challenges, and benefits", *Commun. AIS*, vol. 1, no. 2es, 1999.

[20] Syed Sibte Raza Abidi, "Knowledge management in healthcare: towards 'knowledge-driven'decision-support services", *International Journal of Medical Informatics*, vol. 63, no. 1, pp. 5–18, 2001.

[21] Daniel J Allsopp, Alan Harrison, and Colin Sheppard, "A database architecture for reusable commonkads agent specification components", *Knowledge-Based Systems*, vol. 15, no. 5, pp. 275 – 283, 2002.

[22] J.T Fernandez-Breis and R Martinez-Bejar, "A cooperative tool for facilitating knowledge management", *Expert Systems with Applications*, vol. 18, no. 4, pp. 315 – 330, 2000.

[23] Shu hsien Liao, "Knowledge management technologies and applications—literature review from 1995 to 2002", *Expert Systems with Applications*, vol. 25, no. 2, pp. 155 – 164, 2003.

[24] Karl M. Wiig, Robert de Hoog, and Rob van der Spek, "Supporting knowledge management: A selection of methods and techniques", *Expert Systems with Applications*, vol. 13, no. 1, pp. 15 – 27, 1997.

[25] Sang Bong Yoo and Yeongho Kim, "Web-based knowledge management for sharing product data in virtual enterprises", *International Journal of Production Economics*, vol. 75, no. 1, pp. 173 – 183, 2002, Information Technology/Information Systems in 21st Century Production.

[26] Dongkon Lee and Kyung-Ho Lee, "An approach to case-based system for conceptual ship design assistant", *Expert Systems with Applications*, vol. 16, no. 2, pp. 97 – 104, 1999.

[27] Qijia Tian, Jian Ma, and Ou Liu, "A hybrid knowledge and model system for r and d project selection", *Expert Systems with Applications*, vol. 23, no. 3, pp. 265 – 271, 2002.

[28] Hamid R. Nemati, David M. Steiger, Lakshmi S. Iyer, and Richard T. Herschel, "Knowledge warehouse: an architectural integration of knowledge management, decision support, artificial intelligence and data warehousing", *Decision Support Systems*, vol. 33, no. 2, pp. 143 – 161, 2002.

[29] Cecil Eng Huang Chua, Roger H.L. Chiang, and Ee-Peng Lim, "An intelligent middleware for linear correlation discovery", *Decision Support Systems*, vol. 32, no. 4, pp. 313 – 326, 2002.

[30] B.J. Hicks, S.J. Culley, R.D. Allen, and G. Mullineux, "A framework for the requirements of capturing, storing and reusing information and knowledge in engineering design", *International Journal of Information Management*, vol. 22, no. 4, pp. 263 – 280, 2002.

[31] P. Cunningham and A. Bonzano, "Knowledge engineering issues in developing a case-based reasoning application", *Knowledge-Based Systems*, vol. 12, no. 7, pp. 371 – 379, 1999.

[32] Alexander Sokolov and Fredrik Wulff, "Swingstations: a web-based client tool for the baltic environmental database", *Computers and Geosciences*, vol. 25, no. 7, pp. 863 – 871, 1999.

[33] Arne Koschel and Peter C. Lockemann, "Distributed events in active database systems: Letting the genie out of the bottle", *Data and Knowledge Engineering*, vol. 25, no. 1, pp. 11 – 28, 1998.

[34] Kai W. Wirtz, "Strategies for transforming fine scale knowledge to management usability", *Marine Pollution Bulletin*, vol. 43, no. 7, pp. 209 – 214, 2001.

[35] Gordon Stewart, "Supply-chain operations reference model (scor): the first cross-industry framework for integrated supply-chain management", *Logistics information management*, vol. 10, no. 2, pp. 62–67, 1997.

[36] Value Chain Group, "Vcc: Value reference model".

[37] Francisco M. del Rey Chamorro, Rajkumar Roy, Bert van Wegen, and Andy Steele, "A framework to create key performance indicators for knowledge management solutions", *Journal of Knowledge Management*, vol. 7, no. 2, pp. 46–62, 2003.

[38] M. Abe, J. J. Jeng, and Y. Li, "A tool framework for kpi application development", in *e-Business Engineering, 2007. ICEBE 2007. IEEE International Conference on*, Oct 2007, pp. 22–29.

[39] David Chen, "Enterprise interoperability framework.", in *EMOI-INTEROP*, 2006.

[40] Claudia Diamantini, Laura Genga, Domenico Potena, and Emanuele Storti, *Collaborative Building of an Ontology of Key Performance Indicators*, pp. 148–165, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.

[41] V. Sebastien, D. Sebastien, and N. Conruyt, "An ontology for musical performances analysis: Application to a collaborative platform dedicated to instrumental practice", in *2010 Fifth International Conference on Internet and Web Applications and Services*, May 2010, pp. 538–543.

[42] Michael Uschold and Michael Gruninger, "Ontologies and semantics for seamless connectivity", *SIGMOD Rec.*, vol. 33, no. 4, pp. 58–64, December 2004.

[43] George Anthony Gorry and Michael S Scott Morton, "A framework for management information systems", 1971.

[44] DJ Power, "Decision support systems: concepts and resources for managers", *Studies in Informatics and Control*, vol. 11, no. 4, pp. 349–350, 2002.

[45] Robert H Bonczek, Clyde W Holsapple, and Andrew B Whinston, *Foundations of decision support systems*, Academic Press, 2014.

[46] Danilo Ardagna, Elisabetta Di Nitto, Giuliano Casale, Dana Petcu, Parastoo Mohagheghi, Sébastien Mosser, Peter Matthews, Anke Gericke, Cyril Ballagny, Francesco D'Andria, Cosmin-Septimiu Nechifor, and Craig Sheridan, "Modaclouds: A model-driven approach for the design and execution of applications on multiple clouds", in *Proceedings of the 4th International Workshop on Modeling in Software Engineering*, Piscataway, NJ, USA, 2012, MiSE '12, pp. 50–56, IEEE Press.

[47] Brian L. Dos Santos and Martin L. Bariff, "A study of user interface aids for model-oriented decision support systems", *Management Science*, vol. 34, no. 4, pp. 461–468, 1988.

[48] Eric D. Carlson, "An approach for designing decision support systems", *SIGMIS Database*, vol. 10, no. 3, pp. 3–15, dec 1978.

[49] Chai Zhengmeng and Jiang Haoxiang, "A brief review on decision support systems and it's applications", in *2011 IEEE International Symposium on IT in Medicine and Education*, Dec 2011, vol. 2, pp. 401–405.

[50] Mark Gaynor, Margo Seltzer, Steve Moulton, and Jim Freedman, *A Dynamic, Data-Driven, Decision Support System for Emergency Medical Services*, pp. 703–711, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[51] M. K. Brohman, M. Parent, M. R. Pearce, and M. Wade, "The business intelligence value chain: data-driven decision support in a data warehouse environment: an exploratory study", in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, Jan 2000, pp. 10 pp. vol.1–.

[52] Gerardine DeSanctis and R. Brent Gallupe, "A foundation for the study of group decision support systems", *Management Science*, vol. 33, no. 5, pp. 589–609, 1987.

[53] Tung Bui and Matthias Jarke, "Communications requirements for group decision support systems", *Journal of Management Information Systems*, vol. 2, no. 4, pp. 8–20, 1986.

[54] E. Burton Swanson and Mary J. Culnan, "Document-based systems for management planning and control: A classification, survey, and assessment", *MIS Quarterly*, vol. 2, no. 4, pp. 31–46, 1978.

[55] Effy Oz, Jane Fedorowicz, and Tim Stapleton, "Improving quality, speed and confidence in decision-making", *Information and Management*, vol. 24, no. 2, pp. 71 – 82, 1993.

[56] David Stodolsky, "Steven l. alter: Decision support systems: Current practice and continuing challenges. reading, massachusetts: Addison-wesley publishing co., 1980, 316 pp.".

[57] Tim Berners-Lee, "Www: Past, present, and future", *Computer*, vol. 29, no. 10, pp. 69–77, 1996.

[58] DJ Power, "Web-based decision support systems", *DSstar, The On-Line Executive Journal for Data-Intensive Decision Support*, vol. 2, no. 33, 1998.

[59] Hemant Bhargava and Daniel Power, "Decision support systems and web technologies: a status report", *AMCIS 2001 Proceedings*, p. 46, 2001.

[60] Kittelson Associates Inc., Parsons Brinckerhoff, KFH Group Inc., Texas Transportation Institute, Arup, *Transit Capacity and Quality of Service Manual, 3rd Edition*, Washington D.C., 2013, Transportation Research Board of the National Academies.

[61] Laura Eboli and Gabriella Mazzulla, "Performance indicators for an objective measure of public transport service quality", *European Transport / Trasporti Europei*, vol. 51, 2012.

[62] Deepti S. Muley, Jonathan M. Bunker, and Luis Ferreira, "Evaluating transit quality of service for transit oriented development (tod)", in *30th Australasian Transport Research Forum (ATRF)*, Melbourne, Australia, September 2007.

[63] N.B. Hounsell, B.P. Shrestha, and A. Wong, "Data management and applications in a world-leading bus fleet", *Transportation Research Part C: Emerging Technologies*, vol. 22, pp. 76 – 87, 2012.

[64] Baozhen Yao, Ping Hu, Xiaohong Lu, Junjie Gao, and Mingheng Zhang, "Transit network design based on travel time reliability", *Transportation Research Part C: Emerging Technologies*, vol. 43, Part 3, pp. 233 – 248, 2014.

[65] J. Cao, K. Zeng, H. Wang, J. Cheng, F. Qiao, D. Wen, and Y. Gao, "Web-based traffic sentiment analysis: Methods and applications", *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 844–853, April 2014.

[66] United States Environmental Protection Agency, *Guide to sustainable transportation performance measures*, 2011.

[67] P Nijhout, R Wood, and L Moodley, "An example of public transport modelling with emme/2", in *20th South African Transport Conference*, 2001.

[68] Chhavi Dhingra, "Measuring public transport performance: Lessons for developing countries", *Sustainable Urban Transport Technical Document*, , no. 9, 2011.

[69] Kittleson Associate Inc., Urbitran Inc., LKC Consulting Service Inc., MORPACE International Inc., Queensland University of Technology and Yuko Nakanishi, *A Guidebook for Developing a Transit Performance-Measurement System*, 2003.

[70] CEN TC278, "Reference data model for public transport", *Comité Européen de Normalisation*, 2005.

[71] Roberto Gerin, "Transmodel: Standard europeo nel trasporto pubblico di persone - l'esperienza dell'ACT di Trieste", *European Transport / Trasporti Europei*, , no. 8-9, pp. 68–72, 1998.

[72] Categories Aristotle, "Internet classics archive", *On Sense and the Sensible*, 2009.

[73] Barbara Furletti, "Ontology-driven knowledge discovery", 2009.

[74] Rudolph Goclenius, *Lexicon philosophicum*, Becker, 1980.

[75] Barry Smith, "Ontology", in *Blackwell Guide to the Philosophy of Computing and Information*, Luciano Floridi, Ed., pp. 155–166. Oxford: Blackwell, 2003.

[76] George H. Mealy, "Another look at data", in *Proceedings of the November 14-16, 1967, Fall Joint Computer Conference*, New York, NY, USA, 1967, AFIPS '67 (Fall), pp. 525–534, ACM.

[77] Nicola Guarino and Christopher Welty, "A formal ontology of properties", *Knowledge Engineering and Knowledge Management Methods, Models, and Tools*, pp. 191–230, 2000.

[78] Valentina AM Tamma and Trevor JM Bench-Capon, "Supporting inheritance mechanisms in ontology representation", in *International Conference on Knowledge Engineering and Knowledge Management*. Springer, 2000, pp. 140–155.

[79] Paul Doran, Valentina Tamma, and Luigi Iannone, "Ontology module extraction for ontology reuse: An ontology engineering perspective", in

Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, New York, NY, USA, 2007, CIKM '07, pp. 61–70, ACM.

[80] Deborah L McGuinness, Frank van Harmelen, et al., "Owl web ontology language overview. w3c recommendation, 2004", *URL http://www. w3. org/tr/2004/rec-owl-features-20040210*, 2004.

[81] Gianmario Motta and Giovanni Pignatelli, *Business Process Knowledge Base*, BAI, 2008.

[82] Osvaldo Cairó Battistutti, José Sendra Salcedo, and J. Octavio Gutiérrez-García, "Crowdsourcing information for knowledge-based design of routes for unscheduled public transport trips", *J. Knowledge Management*, vol. 19, no. 3, pp. 626–640, 2015.

[83] Lu Jing, Xu Quchen, and Zhuang Yanping, *Transportation in Shanghai: A Decision Support System to Move towards Sustainability*, School of Engineering, Blekinge Institute of Technology., Karlskrona, Sweden, 2010.

[84] Farman Ali, Daehan Kwak, Pervez Khan, SM Riazul Islam, Kye Hyun Kim, and KS Kwak, "Fuzzy ontology-based sentiment analysis of transportation and city feature reviews for safe traveling", *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 33–48, 2017.

[85] Salima Mnif, Sarra Galoui, Sabeur Elkosantini, Saber Darmoul, and Lamjed Ben Said, "Ontology based performance evaluation of public transport systems", in *4th International Conference on Advanced Logistics and Transport*. 2015, pp. 205–210, IEEE.

[86] M. Houda and M. Khemaja and K. Oliveira and M. Abed, "A public transportation ontology to support user travel planning.", in *RCIS*, Pericles Loucopoulos and Jean-Louis Cavarero, Eds. 2010, pp. 127–136, IEEE.

[87] Junli Wang, Zhijun Ding, and Changjun Jiang, "An ontology-based public transport query system", in *Semantics, Knowledge and Grid, International Conference on*, Los Alamitos, CA, USA, 2005, p. 62, IEEE Computer Society.

[88] D. Lee and R. Meier, "Primary-context model and ontology: A combined approach for pervasive transportation services", in *Pervasive Computing and Communications Workshops, 2007. PerCom Workshops '07. Fifth Annual IEEE International Conference on*, March 2007, pp. 419–424.

[89] Káthia Marçal de Oliveira, Firas Bacha, Houda Mnasser, and Mourad Abed, "Transportation ontology definition and application for the content personalization of user interfaces", *Expert Systems with Applications*, vol. 40, no. 8, pp. 3145 – 3159, 2013.

[90] Chang-Shing Lee, Zhi-Wei Jian, and Lin-Kai Huang, "A fuzzy ontology and its application to news summarization", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 5, pp. 859–880, Oct 2005.

[91] Daniel Oberle, Stephan Grimm, and Steffen Staab, "An ontology for software", in *Handbook on ontologies*, pp. 383–402. Springer, 2009.

[92] Gaurav V Jain, SS Jain, and Manoranjan Parida, "Public transport ontology for passenger information retrieval", *International Journal of Transportation Engineering*, vol. 2, no. 2, pp. 131–144, 2015.

[93] Sabine Timpf, "Ontologies of wayfinding: a traveler's perspective", *Networks and spatial economics*, vol. 2, no. 1, pp. 9–33, 2002.

[94] Stijn Verstichel, Femke Ongenae, Leanneke Loeve, Frederik Vermeulen, Pieter Dings, Bart Dhoedt, Tom Dhaene, and Filip De Turck, "Efficient data integration in the railway domain through an ontology-based methodology", *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 617 – 643, 2011.

[95] Paloma Cáceres, Carlos E Cuesta, José María Cavero, Belén Vela, and Almudena Sierra-Alonso, "Towards knowledge modeling for sustainable transport", in *International Conference on Software Engineering and Formal Methods*. Springer, 2013, pp. 271–287.

[96] Viara Popova and Alexei Sharpanskykh, "Modeling organizational performance indicators", *Information Systems*, vol. 35, no. 4, pp. 505–527, 2010.

[97] Adela del Río-Ortega, Manuel Resinas, Cristina Cabanillas, and Antonio Ruiz-Cortés, "On the definition and design-time analysis of process performance indicators", *Information Systems*, vol. 38, no. 4, pp. 470–490, 2013.

[98] Jennifer Horkoff, Daniele Barone, Lei Jiang, Eric Yu, Daniel Amyot, Alex Borgida, and John Mylopoulos, "Strategic business modeling: representation and reasoning", *Software & Systems Modeling*, 2012.

[99] Claudia Diamantini, Domenico Potena, and Emanuele Storti, "Sempi: A semantic framework for the collaborative construction and maintenance

of a shared dictionary of performance indicators", *Future Generation Computer Systems*, vol. 54, pp. 352 – 365, 2016.

[100] Christoph Lange, "Ontologies and languages for representing mathematical knowledge on the semantic web", *Semantic Web*, vol. 4, no. 2, pp. 119–158, 2013.

[101] Dr.Claus Dohmen, "Modelling it systems for public transport companies: the domain model ittc", *Transportation Research Procedia*, vol. 25, no. Supplement C, pp. 1846 – 1864, 2017, World Conference on Transport Research - WCTR 2016 Shanghai. 10-15 July 2016.

[102] Glenn Lyons, Reg Harman, John Austin, and Alastair Duff, "Traveller information systems research: a review and recommendations for transport direct", 2001.

[103] Claudia Diamantini, Domenico Potena, and Emanuele Storti, "Extended drill-down operator: Digging into the structure of performance indicators", *Concurrency and Computation: Practice and Experience*, pp. n/a–n/a, 2015, cpe.3726.

[104] Claudia Diamantini, Alessandro Freddi, Sauro Longhi, Domenico Potena, and Emanuele Storti, "A goal-oriented, ontology-based methodology to support the design of aal environments", *Expert Systems with Applications*, vol. 64, pp. 117 – 131, 2016.

[105] Laura Cecilia Cham, *Understanding bus service reliability: a practical framework using AVL/APC data*, PhD thesis, Massachusetts Institute of Technology, 2006.

[106] Chen, Xumei and Yu, Lei and Zhang, Yushi and Guo, Jifu, "Analyzing urban bus service reliability at the stop, route, and network levels", *Transportation research part A: policy and practice*, vol. 43, no. 8, pp. 722–734, 2009.

[107] Ralph Kimball and Margy Ross, *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*, John Wiley & Sons, Inc., New York, NY, USA, 2nd edition, 2002.

[108] Stephen Buswell, Olga Caprotti, David P Carlisle, Michael C Dewar, Marc Gaetano, and Michael Kohlhase, "The open math standard", Tech. Rep., version 2.0. Technical report, The Open Math Society, 2004. http://www.openmath.org/standard/om20, 2004.

[109] Jérôme Euzenat, Pavel Shvaiko, et al., *Ontology matching*, vol. 18, Springer, 2007.

[110] Denny Vrandečić, *Ontology Evaluation*, pp. 293–313, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[111] Thomas R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing?", *International Journal of Human-Computer Studies*, vol. 43, no. 5, pp. 907 – 928, 1995.

[112] Leo Obrst, Werner Ceusters, Inderjeet Mani, Steve Ray, and Barry Smith, *The Evaluation of Ontologies*, pp. 139–158, Springer US, Boston, MA, 2007.

[113] Leon Sterling, Alan Bundy, Lawrence Byrd, Richard O'Keefe, and Bernard Silver, "Solving symbolic equations with press", *Journal of Symbolic Computation*, vol. 7, no. 1, pp. 71 – 84, 1989.

[114] S. Payne, "Study on key performance indicators for intelligent transport systems: final report in support of the implementation of the eu legislative framework on ITS (directive 2010/40/eu)", 2015.

[115] P Ryus, M Connor, S Corbett, A Rodenstein, L Wargelin, L Ferreira, Y Nakanishi, and K Blume, "Tcrp report 88: A guidebook for developing a transit performance-measurement system", *Transit Cooperative Research Program*, 2003.

[116] Ian Wallis Associated Ltd. and The TAS Partnership, *Improving bus service reliability*, September 2013.

[117] Jamie Houghton, John Reiners, and Colin Lim, "Intelligent transport: How cities can improve mobility", *IBM Institute for Business Value*, 2009.

[118] Wei-Hsun Lee, Shian-Shyong Tseng, and Sheng-Han Tsai, "A knowledge based real-time travel time prediction system for urban network", *Expert Systems with Applications*, vol. 36, no. 3, pp. 4239–4247, 2009.

[119] Chun-Hsin Wu, Jan-Ming Ho, and Der-Tsai Lee, "Travel-time prediction with support vector regression", *IEEE transactions on intelligent transportation systems*, vol. 5, no. 4, pp. 276–281, 2004.

[120] Wei-Hua Lin and Robert L Bertini, "Modeling schedule recovery processes in transit operations for bus arrival time prediction", *Journal of Advanced Transportation*, vol. 38, no. 3, pp. 347–365, 2004.

[121] RPS Padmanaban, Lelitha Vanajakshi, and Shankar C Subramanian, "Estimation of bus travel time incorporating dwell time for apts applications", in *Intelligent Vehicles Symposium, 2009 IEEE*. IEEE, 2009, pp. 955–959.

[122] Clark Glymour, David Madigan, Daryl Pregibon, and Padhraic Smyth, "Statistical themes and lessons for data mining", *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 11–28, Mar 1997.

[123] Seyed Mojtaba Tafaghod Sadat Zadeh, TONI ANWAR, and MINA BASIRAT, "A survey on application of artificial intelligence for bus arrival time prediction.", *Journal of Theoretical & Applied Information Technology*, vol. 46, no. 1, 2012.

[124] Matthew G Karlaftis and Eleni I Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights", *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387–399, 2011.

[125] Mei Chen, Xiaobo Liu, Jingxin Xia, and Steven I. Chien, "A dynamic bus-arrival time prediction model based on apc data", *Computer-Aided Civil and Infrastructure Engineering*, vol. 19, no. 5, pp. 364–376, 2004.

[126] Rudolph Emil Kalman et al., "A new approach to linear filtering and prediction problems", *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[127] Amer Shalaby and Ali Farhan, "Prediction model of bus arrival and departure times using avl and apc data", *Journal of Public Transportation*, vol. 7, no. 1, pp. 3, 2004.

[128] Dongjoo Park and Laurence R Rilett, "Forecasting freeway link travel times with a multilayer feedforward neural network", *Computer-Aided Civil and Infrastructure Engineering*, vol. 14, no. 5, pp. 357–367, 1999.

[129] Sanghoon Bae and Pushkin Kachroo, "Proactive travel time predictions under interrupted flow condition", in *Vehicle Navigation and Information Systems Conference, 1995. Proceedings. In conjunction with the Pacific Rim TransTech Conference. 6th International VNIS.'A Ride into the Future'*. IEEE, 1995, pp. 179–186.

[130] Lelitha Vanajakshi, Shankar C Subramanian, and R Sivanandan, "Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses", *IET intelligent transport systems*, vol. 3, no. 1, pp. 1–9, 2009.

[131] Chumchoke Nanthawichit, Takashi Nakatsuji, and Hironori Suzuki, "Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway", *Transportation Research Record: Journal of the Transportation Research Board*, , no. 1855, pp. 49–59, 2003.

[132] Yaochu Jin and Bernhard Sendhoff, "Pareto-based multiobjective machine learning: An overview and case studies", *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 3, pp. 397–415, 2008.

[133] Friedrich Recknagel, "Applications of machine learning to ecological modelling", *Ecological Modelling*, vol. 146, no. 1, pp. 303–310, 2001.

[134] JWC Van Lint, SP Hoogendoorn, and Henk J van Zuylen, "Accurate freeway travel time prediction with state-space neural networks under missing data", *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 5, pp. 347–369, 2005.

[135] Hao Liu, Henk van Zuylen, Hans van Lint, and Maria Salomons, "Predicting urban arterial travel time with state-space neural networks and kalman filters", *Transportation Research Record: Journal of the Transportation Research Board*, , no. 1968, pp. 99–108, 2006.

[136] Ranhee Jeong and R Rilett, "Bus arrival time prediction using artificial neural network model", in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*. IEEE, 2004, pp. 988–993.

[137] Taehyung Park and Sangkeon Lee, "A bayesian approach for estimating link travel time on urban arterial road network", *Computational Science and Its Applications–ICCSA 2004*, pp. 1017–1025, 2004.

[138] Mei Chen and Steven Chien, "Dynamic freeway travel-time prediction with probe vehicle data: Link based versus path based", *Transportation Research Record: Journal of the Transportation Research Board*, , no. 1768, pp. 157–161, 2001.

[139] Chandra Kuchipudi and Steven Chien, "Development of a hybrid model for dynamic travel-time prediction", *Transportation Research Record: Journal of the Transportation Research Board*, , no. 1855, pp. 22–31, 2003.

[140] Jiann-Shiou Yang, "Travel time prediction using the gps test vehicle and kalman filtering techniques", in *American Control Conference, 2005. Proceedings of the 2005*. IEEE, 2005, pp. 2128–2133.

[141] Z Wall and DJ Dailey, "An algorithm for predicting the arrival time of mass transit vehicles using automatic vehicle location data", in *in 78th Annual Meeting of the Transportation Research Board, National Research Council, Washington DC*. Citeseer, 1999.

[142] Dihua Sun, Hong Luo, Liping Fu, Weining Liu, Xiaoyong Liao, and Min Zhao, "Predicting bus arrival time on the basis of global positioning system data", *Transportation Research Record: Journal of the Transportation Research Board*, , no. 2034, pp. 62–72, 2007.

[143] Xiang Fei, Chung-Cheng Lu, and Ke Liu, "A bayesian dynamic linear model approach for real-time short-term freeway travel time prediction", *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1306–1318, 2011.

[144] Bin Yu, William HK Lam, and Mei Lam Tam, "Bus arrival time prediction at bus stop with multiple routes", *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1157–1170, 2011.

[145] Bo YU, Jing LU, Bin YU, and Zhongzhen YANG, "An adaptive bus arrival time prediction model", *Journal of the Eastern Asia Society for Transportation Studies*, vol. 8, pp. 1126–1136, 2010.

[146] Sonia Khetarpaul, SK Gupta, Shikhar Malhotra, and L Venkata Subramaniam, "Bus arrival time prediction using a modified amalgamation of fuzzy clustering and neural network on spatio-temporal data", in *Australasian Database Conference*. Springer, 2015, pp. 142–154.

[147] Cong Bai, Zhong-Ren Peng, Qing-Chang Lu, and Jian Sun, "Dynamic bus travel time prediction models on road with multiple bus routes", *Computational intelligence and neuroscience*, vol. 2015, pp. 63, 2015.

[148] M Zaki, I Ashour, M Zorkany, and B Hesham, "Online bus arrival time prediction using hybrid neural network and kalman filter techniques", *International Journal of Modern Engineering Research*, vol. 3, no. 4, pp. 2035–2041, 2013.

[149] "Weka".

[150] Nam H. Vu and Ata M. Khan, "Bus running time prediction using a statistical pattern recognition technique", *Transportation Planning and Technology*, vol. 33, no. 7, pp. 625–642, 2010.

[151] Simon Haykin, *Neural networks: a comprehensive foundation*, Prentice Hall PTR, 1994.

[152] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik, "A training algorithm for optimal margin classifiers", in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 144–152.

[153] Karl Pearson, "Liii. on lines and planes of closest fit to systems of points in space", *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901.

[154] K.a Khoshelham, C.b Nardinocchi, E.c Frontoni, A.c Mancini, and P.c Zingaretti, "Performance evaluation of automated approaches to building detection in multi-source aerial data", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 65, no. 1, pp. 123–133, 2010.

[155] Claudia Diamantini and Domenico Potena, "Bayes vector quantizer for class-imbalance problem.", in *SEBD*, Valeria De Antonellis, Silvana Castano, Barbara Catania, and Giovanna Guerrini, Eds. 2009, pp. 305–312, Edizioni Seneca.

[156] Tahir M.A., Kittler J., and Yan F., "Inverse random under sampling for class imbalance problem and its application to multi-label classification", *Pattern Recognition*, vol. 45, no. 10, pp. 3738–3750, 2012.

[157] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami, "Mining association rules between sets of items in large databases", *SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, June 1993.

[158] Mikhail V Kiselev, "Polyanalyst-a machine discovery system inferring functional programs.", in *KDD Workshop*, 1994, pp. 237–250.

[159] Mikhail V Kiselev, "Polyanalyst 2.0: combination of statistical data preprocessing and symbolic kdd technique", in *Proceedings of ECML-95 Workshop on Statistics, Machine Learning and Knowledge Discovery in Databases, Heraklion, Greece*, 1995, pp. 187–192.

[160] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth, "From data mining to knowledge discovery in databases", *AI magazine*, vol. 17, no. 3, pp. 37, 1996.

[161] Claudia Diamantini, Domenico Potena, and Emanuele Storti, *Data Semantics Meets Knowledge Discovery in Databases*, pp. 391–405, Springer International Publishing, 2018.

[162] John F Elder and Dean W Abbott, "A comparison of leading data mining tools", in *Fourth International Conference on Knowledge Discovery and Data Mining*, 1998, vol. 28.

[163] Michael Goebel and Le Gruenwald, "A survey of data mining and knowledge discovery software tools", *ACM SIGKDD explorations newsletter*, vol. 1, no. 1, pp. 20–33, 1999.

[164] Xue Z. Wang, *Data Mining and Knowledge Discovery: an Overview*, pp. 13–28, Springer London, London, 1999.

[165] H. Chauhan, V. Kumar, S. Pundir, and E. S. Pilli, "A comparative study of classification techniques for intrusion detection", in *2013 International Symposium on Computational and Business Intelligence*, Aug 2013, pp. 40–43.

[166] Fatos Elezi, Armin Sharafi, Alexander Mirson, Petra Wolf, Helmut Krcmar, and Udo Lindemann, "A knowledge discovery in databases (kdd) approach for extracting causes of iterations in engineering change orders", in *Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2011, pp. 1401–1410.

[167] Mira Abboud, Hala Naja, Mourad Oussalah, and Mohamad Dbouk, "Kdd extension tool for software architecture extraction", 2017.

[168] M. Collin, F. Flouvat, and N. Selmaoui-Folcher, "Patsi: Pattern mining of time series of satellite images in knime", in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, Dec 2016, pp. 1292–1295.

[169] S. Günnemann, H. Kremer, R. Musiol, R. Haag, and T. Seidl, "A subspace clustering extension for the knime data mining framework", in *2012 IEEE 12th International Conference on Data Mining Workshops*, Dec 2012, pp. 886–889.

[170] J. Chattratichat, J. Darlington, Y. Guo, S. Hedvall, M. Köhler, and J. Syed, *An architecture for distributed enterprise data mining*, pp. 573–582, Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[171] Srinivasan Parthasarathy and Ramesh Subramonian, "Facilitating data mining on a network of workstations", *Advances in Distributed Data Mining*, pp. 229–254, 1999.

[172] O. Rana, D. Walker, Maozhen Li, S. Lynden, and M. Ward, "Paddmas: parallel and distributed data mining application suite", in *Proceedings 14th International Parallel and Distributed Processing Symposium. IPDPS 2000*, 2000, pp. 387–392.

[173] A. Kumar, M. Kantardzic, P. Ramaswamy, and P. Sadeghian, "An extensible service oriented distributed data mining framework", in *2004 International Conference on Machine Learning and Applications, 2004. Proceedings.*, Dec 2004, pp. 256–263.

[174] Petar Ristoski and Heiko Paulheim, "Semantic web in data mining and knowledge discovery: A comprehensive survey", *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 36, pp. 1 – 22, 2016.

[175] Michael Zeller, Robert Grossman, Christoph Lingenfelder, Michael R. Berthold, Erik Marcade, Rick Pechter, Mike Hoskins, Wayne Thompson, and Rich Holada, "Open standards and cloud computing: Kdd-2009 panel report", in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2009, KDD '09, pp. 11–18, ACM.

[176] Robert Grossman and Yunhong Gu, "Data mining using high performance data clouds: Experimental studies using sector and sphere", in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2008, KDD '08, pp. 920–927, ACM.

[177] Yen-Ting Kuo, Andrew Lonie, Liz Sonenberg, and Kathy Paizis, "Domain ontology driven data mining: A medical case study", in *Proceedings of the 2007 International Workshop on Domain Driven Data Mining*, New York, NY, USA, 2007, DDDM '07, pp. 11–17, ACM.

[178] Khaled Khelif, Rose Dieng-Kuntz, and Pascal Barbry, "An ontology-based approach to support text mining and information retrieval in the biological domain.", *J. UCS*, vol. 13, no. 12, pp. 1881–1907, 2007.

[179] Claudia Diamantini, Domenico Potena, and Emanuele Storti, "Kddonto: An ontology for discovery and composition of kdd algorithms", 2009.

[180] Panče Panov, Larisa Soldatova, and Sašo Džeroski, *OntoDM-KDD: Ontology for Representing the Knowledge Discovery Process*, pp. 126–140, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[181] X.a Hu, Y.b Li, J.a c Shan, J.a Zhang, and Y.a Zhang, "Road centerline extraction in complex urban scenes from lidar data based on multiple features", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 7448–7456, 2014.

[182] Jiaojiao Tian, Shiyong Cui, and P. Reinartz, "Building change detection based on satellite stereo imagery and digital surface models", *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 52, no. 1, pp. 406–417, Jan 2014.

[183] Mourad Bouziani, Kalifa Goïta, and Dong-Chen He, "Automatic change detection of buildings in urban environment from very high spatial resolution images using existing geodatabase and prior knowledge", *{ISPRS}*

Journal of Photogrammetry and Remote Sensing, vol. 65, no. 1, pp. 143 – 153, 2010.

[184] J.A. Malpica, M.C. Alonso, F. Papí, A. Arozarena, and A.M. De Agirrea, "Change detection of buildings from satellite imagery and lidar data", *International Journal of Remote Sensing*, vol. 34, no. 5, pp. 1652–1675, 2013.

[185] G. Forlani, R. Roncella, and C. Nardinocchi, "Where is photogrammetry heading to? state of the art and trends", *Rendiconti Lincei*, vol. 26, pp. 85–96, 2015.

[186] Y.a Huang, B.a Yu, J.a Zhou, C.b Hu, W.b Tan, Z.a Hu, and J.a Wu, "Toward automatic estimation of urban green volume using airborne lidar data and high resolution remote sensing images", *Frontiers of Earth Science*, vol. 7, no. 1, pp. 43–54, 2013.

[187] B.a Yu, H.b Liu, J.a Wu, Y.a Hu, and L.a Zhang, "Automated derivation of urban building density information using airborne lidar data and object-based method", *Landscape and Urban Planning*, vol. 98, no. 3-4, pp. 210–219, 2010.

[188] L.a Cheng, J.b Gong, M.a Li, and Y.a Liu, "3d building model reconstruction from multi-view aerial imagery and lidar data", *Photogrammetric Engineering and Remote Sensing*, vol. 77, no. 2, pp. 125–139, 2011.

[189] S. Oude Elberink and G. Vosselman, "Quality analysis on 3d building models reconstructed from airborne laser scanning data", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 2, pp. 157–165, 2011.

[190] R. Wang, "3d building modeling using images and lidar: a review", *International Journal of Image and Data Fusion*, vol. 4, no. 4, pp. 273–292, 2013.

[191] J. Yan, K. Zhang, C. Zhang, S.-C. Chen, and G. Narasimhan, "Automatic construction of 3-d building model from airborne lidar data through 2-d snake algorithm", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 1, pp. 3–14, 2015.

[192] Gianfranco Forlani, Carla Nardinocchi, Marco Scaioni, and Primo Zingaretti, "Complete classification of raw lidar data and 3d reconstruction of buildings.", *Pattern Anal. Appl.*, vol. 8, no. 4, pp. 357–374, 2006.

[193] M.a Awrangjeb, C.b Zhang, and C.S.b Fraser, "Automatic extraction of building roofs using lidar data and multispectral imagery", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 83, pp. 1–18, 2013.

[194] M. Awrangjeb, C. Zhang, and C.S. Fraser, "Building detection in complex scenes thorough effective separation of buildings from trees", *Photogrammetric Engineering and Remote Sensing*, vol. 78, no. 7, pp. 729–745, 2012.

[195] Joachim Niemeyer, Franz Rottensteiner, and Uwe Soergel, "Contextual classification of lidar data and building object detection in urban areas", *{ISPRS} Journal of Photogrammetry and Remote Sensing*, vol. 87, pp. 152 – 165, 2014.

[196] D. Gonzalez-Aguilera, E. Crespo-Matellan, D. Hernandez-Lopez, and P. Rodriguez-Gonzalvez, "Automated urban analysis based on lidar-derived building models", *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 51, no. 3, pp. 1844–1851, March 2013.

[197] Y. Zeng, J. Zhang, G. Wang, and Z. Lin, "Urban land-use classification using integrated airborn laser scanning data and high resolution multispectral imagery", *Pecora 15/Land Satellite Information IV/ISPRS Commssion I/FIEOS*, 2002.

[198] S. Syed, P. Dare, and S. Jones, "Automatic classification of land cover features with high resolution imagery and lidar data: An object oriented approach", *Proceedings of SSC2005 Spatial Intelligence, Innovation and Praxis: The National Biennial Conference of the Spatial Sciences Institute*, 2005.

[199] M.C. Alonso and J.A. Malpica, "Classification of multispectral high-resolution satellite imagery using lidar elevation data", *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5359 LNCS, no. PART 2, pp. 85–94, 2008.

[200] Y. Ke, L.J. Quackenbush, and J. Im, "Synergistic use of quickbird multispectral imagery and lidar data for object-based forest species classification", *Remote Sensing of Environment*, vol. 114, no. 6, pp. 1141–1154, 2010.

[201] G. Zhou and X. Zhou, "Seamless fusion of lidar and aerial imagery for building extraction", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 7393–7407, 2014.

[202] M. Gerke and J. Xiao, "Fusion of airborne laserscanning point clouds and images for supervised and unsupervised scene classification", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 87, pp. 78–92, 2014.

[203] E.W.a Bork and J.G.b Su, "Integrating lidar data and multispectral imagery for enhanced classification of rangeland vegetation: A meta analysis", *Remote Sensing of Environment*, vol. 111, no. 1, pp. 11–24, 2007.

[204] L.a Guo, N.a b Chehata, C.b Mallet, and S.a Boukir, "Relevance of airborne lidar and multispectral image data for urban scene classification using random forests", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 66, no. 1, pp. 56–66, 2011.

[205] Carla Rebelo, Antonio Manuel Rodrigues, Jose Antonio Tenedorio, Jose Alberto Goncalves, and Joa Marnoto, "Building 3d city models: Testing and comparing laser scanning and low-cost uav data using foss technologies", in *Computational Science and Its Applications – ICCSA 2015*, Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Marina L. Gavrilova, Ana Maria Alves Coutinho Rocha, Carmelo Torre, David Taniar, and Bernady O. Apduhan, Eds., vol. 9157 of *Lecture Notes in Computer Science*, pp. 367–379. Springer International Publishing, 2015.

[206] A. Mancini, E. Frontoni, and P. Zingaretti, "A winner takes all mechanism for automatic object extraction from multi-source data", *2009 17th International Conference on Geoinformatics, Geoinformatics 2009*, 2009.

[207] E.S.a Malinverni, A.N.a Tassetti, A.b Mancini, P.b Zingaretti, E.b Frontoni, and A.a Bernardini, "Hybrid object-based approach for land use/land cover mapping using high spatial resolution imagery", *International Journal of Geographical Information Science*, vol. 25, no. 6, pp. 1025–1043, 2011.

[208] A.a Bernardini, E.b Frontoni, E.S.a Malinverni, A.b Mancini, A.N.a Tassetti, and P.b Zingaretti, "Pixel, object and hybrid classification comparisons", *Journal of Spatial Science*, vol. 55, no. 1, pp. 43–54, 2010.

[209] K Khoshelham, S Nedkov, and C Nardinocchi, "A comparison of bayesian and evidence-based fusion methods for automated building detection in aerial data", *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, pp. 1183–1188, 2008.

[210] Franz Rottensteiner, John Trinder, Simon Clode, and Kurt Kubik, "Using the dempster shafer method for the fusion of {LIDAR} data and multi-spectral images for building detection", *Information Fusion*, vol. 6, no. 4, pp. 283 – 300, 2005, Fusion of Remotely Sensed Data over Urban Areas.

[211] Mustafa Turker and Dilek Koc-San, "Building extraction from high-resolution optical spaceborne images using the integration of support

vector machine (svm) classification, hough transformation and perceptual grouping", *International Journal of Applied Earth Observation and Geoinformation*, vol. 34, pp. 58 – 69, 2015.

[212] Jixian Zhang, Xiangguo Lin, and Xiaogang Ning, "Svm-based classification of segmented airborne lidar point clouds in urban areas", *Remote Sensing*, vol. 5, no. 8, pp. 3749, 2013.

[213] Alberto Fernandez, Victoria Lopez, Mikel Galar, Maria Jose del Jesus, and Francisco Herrera, "Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches", *Knowledge-Based Systems*, vol. 42, pp. 97 – 110, 2013.

[214] M. Awrangjeb and C.S. Fraser, "Automatic segmentation of raw lidar data for extraction of building roofs", *Remote Sensing*, vol. 6, no. 5, pp. 3716–3751, 2014.

[215] Gary M. Weiss and Foster Provost, "Learning when training data are costly: The effect of class distribution on tree induction", *Journal of Artificial Intelligence Research*, vol. 19, no. 1, pp. 315–354, October 2003.

[216] A. Mellor, S. Boukir, A. Haywood, and S. Jones, "Exploring issues of training data imbalance and mislabelling on random forest performance for large area land cover classification using the ensemble margin", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 155–168, 2015.

[217] Claudia Diamantini and Domenico Potena, "Bayes vector quantizer for class-imbalance problem", *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 5, pp. 638–651, 2009.

[218] Claudia Diamantini and A. Spalvieri, "Quantizing for minimum average misclassification risk", *Neural Networks, IEEE Transactions on*, vol. 9, no. 1, pp. 174–182, Jan 1998.

[219] M. Bossard, J. Feranec, and J. Otahel, "CORINE land cover technical guide - addendum 2000", Tech. Rep. 40, European Environment Agency, May 2000.

[220] R.G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data", *Remote Sensing of Environment*, vol. 37, no. 1, pp. 35–46, 1991.

[221] E.N. Smirnov and A. Kaptein, "Theoretical and experimental study of a meta-typicalness approach for reliable classification", in *Proceedings*

on IEEE International Conference on Data Mining, ICDM. 2006, pp. 739–743, IEEE.

[222] Thomas M Cover and Peter E Hart, "Nearest neighbor pattern classification", *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.

[223] Pedro Domingos, "Metacost: A general method for making classifiers cost-sensitive", in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 155–164.

[224] J Ross Quinlan, *C4. 5: programs for machine learning*, Elsevier, 2014.

[225] Jerome Friedman, Trevor Hastie, and Robert Tibshirani, "Additive logistic regression: A statistical view of boosting", *The Annals of Statistics*, vol. 38, no. 2, pp. 337–374, April 2000.

[226] Robert E. Schapire and Yoram Singer, "Improved boosting algorithms using confidence-rated predictions", in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, New York, NY, USA, 1998, COLT' 98, pp. 80–91, ACM.

[227] Yoav Freund and Robert E. Schapire, "Game theory, on-line prediction and boosting", in *In Proceedings of the Ninth Annual Conference on Computational Learning Theory*. 1996, pp. 325–332, ACM Press.

[228] Claudia Diamantini, Alberto Gemelli, and Domenico Potena, "A geometric approach to feature ranking based upon results of effective decision boundary feature matrix", in *Feature Selection for Data and Pattern Recognition*, pp. 45–69. Springer Verlag, 2015.

[229] Claudia Diamantini, Alberto Gemelli, and Domenico Potena, "Feature ranking based on decision border", in *20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010*, 2010, pp. 609–612.

[230] Claudia Diamantini and Domenico Potena, "Bvq-based feature extraction: a computational analysis", in *Proceedings of the Fourteenth Italian Symposium on Advanced Database Systems, SEBD 2006, Portonovo (Ancona), 18-21 June 2006*, 2006, pp. 297–308.