UNIVERSITE DE LIMOGES

ECOLE DOCTORALE SCIENCE - TECHNOLOGIE

FACULTE DES SCIENCES & TECHNIQUES

Laboratoire XLIM

Thèse N°

Thèse

pour obtenir le grade de

DOCTEUR DE L‘‘UNIVERSITÉ DE LIMOGES

Discipline / Spécialité : Informatique

présentée et

soutenue par

Nikolaos SIDERIS

le 26 novembre 2019

Spatial Decision Support in Urban Environments using Machine Learning, 3D Geo-visualization and Semantic Integration of Multi-Source Data

Thèse dirigée par

Professeur Georges MIAOULIS et

Professeur Djamchid GHAZANFARPOUR

JURY :

Président :

Yves DUTHEN Professeur Université de Toulouse

Examinateurs :

Georges MIAOULIS Professeur Université de West Attica

Djamchid GHAZANFARPOUR Professeur Université de Limoges XLIM

Elli PETSA Professeur Université de West Attica

I dedicate this work to everyone

who inspired and supported me

during these years.

Acknowledgements

This document is my thesis at University of Limoges. This research area is dealing with semantic modelling, geodatabases and georeferenced urban data, 3D visualization and machine learning techniques. I would like to thank my research committee that supervised this work and everyone who supported this work.

In particular,

I would like to thank Dr. Georgios Miaoulis, Professor of the department of Informatics at University of West Attica, whose expertise was invaluable in the formulating of the research topic and methodology in particular and for his support to me on both academic and personal level.

I would like to thank Dr. Djamchid Ghazanfarpour, Professor of the department of Informatics at University of Limoges, for having supervised and supported my work.

I would also like to thank Thanos Voulodimos Assistant Professor and Georgios Bardis Lecturer, faculty members of department of Informatics and Computer Engineering at University of West Attica, for their valuable advices, helpful comments and excellent interaction in a very warm atmosphere of cooperation as co-encadrants in this thesis.

Table of Contents

Table of Contents

List of Tables

List of Figures

List of Graphs

Résumé

La quantité et la disponibilité sans cesse croissantes de données urbaines dérivées de sources variées posent de nombreux problèmes, notamment la consolidation, la visualisation et les perspectives d'exploitation maximales des données susmentionnées. Un problème prééminent qui affecte l'urbanisme est le choix du lieu approprié pour accueillir une activité particulière (service social ou commercial commun) ou l'utilisation correcte d'un bâtiment existant ou d'un espace vide.

Dans cette thèse, nous proposons une approche pour aborder les défis précédents rencontrés avec les techniques d'apprentissage automatique , le classifieur de forêts aléatoires comme méthode dominante dans un système qui combine et fusionne divers types de données provenant de sources différentes, et les code à l'aide d'un nouveau modèle sémantique. qui peut capturer et utiliser à la fois des informations géométriques de bas niveau et des informations sémantiques de niveau supérieur et les transmet ensuite au classifieur de forêts aléatoires. Les données sont également transmises à d'autres classificateurs et les résultats sont évalués pour confirmer la prévalence de la méthode proposée.

Les données extraites proviennent d'une multitude de sources, par exemple: fournisseurs de données ouvertes et organisations publiques s'occupant de planification urbaine. Lors de leur récupération et de leur inspection à différents niveaux (importation, conversion, géospatiale, par exemple), ils sont convertis de manière appropriée pour respecter les règles du modèle sémantique et les spécifications techniques des sous-systèmes correspondants. Des calculs géométriques et géographiques sont effectués et des informations sémantiques sont extraites.

Enfin, les informations des étapes précédentes, ainsi que les résultats des techniques d'apprentissage automatique et des méthodes multicritères, sont intégrés au système et visualisés dans un environnement Web frontal capable d'exécuter et de visualiser des requêtes spatiales, permettant ainsi la gestion de trois processus. objets géoréférencés dimensionnels, leur récupération, transformation et visualisation, en tant que système d'aide à la décision.

Abstract

The constantly increasing amount and availability of urban data derived from varying sources leads to an assortment of challenges that include, among others, the consolidation, visualization, and maximal exploitation prospects of the aforementioned data. A preeminent problem affecting urban planning is the appropriate choice of location to host a particular activity (either commercial or common welfare service) or the correct use of an existing building or empty space.

In this thesis we propose an approach to address the preceding challenges availed with machine learning techniques with the random forests classifier as its dominant method in a system that combines, blends and merges various types of data from different sources, encode them using a novel semantic model that can capture and utilize both low-level geometric information and higher level semantic information and subsequently feeds them to the random forests classifier. The data are also forwarded to alternative classifiers and the results are appraised to confirm the prevalence of the proposed method.

The data retrieved stem from a multitude of sources, e.g. open data providers and public organizations dealing with urban planning. Upon their retrieval and inspection at various levels (e.g. import, conversion, geospatial) they are appropriately converted to comply with the rules of the semantic model and the technical specifications of the corresponding subsystems. Geometrical and geographical calculations are performed and semantic information is extracted.

Finally, the information from individual earlier stages along with the results from the machine learning techniques and the multicriteria methods are integrated into the system and visualized in a front-end web based environment able to execute and visualize spatial queries, allow the management of three-dimensional georeferenced objects, their retrieval, transformation and visualization, as a decision support system.

Introduction

1. Introduction

1.1. General Introduction

The constantly increasing amount and availability of urban data derived from varying sources leads to an assortment of challenges that include, among others, the consolidation, visualization, and maximal exploitation prospects of the aforementioned data. A preeminent problem affecting urban planning is the appropriate choice of location to host a particular activity (either commercial or common welfare service) or the correct use of an existing building or empty space. In the present thesis we propose an approach to address the preceding challenges availed with machine learning techniques with the random forests classifier as its dominant method in a system that combines, blends and merges various types of data from different sources, encode them using a novel semantic model that can capture and utilize both low-level geometric information and higher level semantic information and subsequently feeds them to the random forests classifier. The data are also forwarded to alternative classifiers and the results are appraised to confirm the prevalence of the proposed method.

1.2. Problem Formulation and thesis objective

As was formally announced at the meeting of the World Planners Congress and the UN Habitat World Urban Forum in 2008, the part of the total Earth population living in cities is greater than the part that resides in a non-urban environment. In the decade that has elapsed since then, the importance as well as the complexity of urban planning have grown exponentially, yet the related intricacies are not always sufficiently acknowledged. Urban planning is an interdisciplinary process, highly demanding in terms of interconnection among the subsystems involved, as it spreads over a wide range of fields, including legal matters and legislation, political and social issues, capital investment, finance expenditures and others, while being computationally intensive due to the volume of participating data and inherently providing a very limited margin for errors and re-runs.

The availability of an increasing amount of heterogeneous data, stemming from a wide range of sensors installed throughout the city, lately coined as “urban big data”, appears to provide new streams of information to exploit in urban planning. Nevertheless, effectively leveraging such information is far from straightforward, since the involved multidisciplinary stakeholders do not necessarily possess the specialized knowledge and understanding of the concepts from the relevant different domains. Moreover, there is a lack of efficient computational

tools that would help translate these massive amounts of data into comprehensible, usable, and even actionable hints in the urban planning and development process.

Objective of the thesis is to ameliorate parts of the aforementioned problematic sections by developing a decision support system containing a 3D environment with the ability to visualize georeferenced urban data, navigate and execute spatial queries, and capable to exploit semantic information, apply machine learning techniques in urban data and utilize its results in a seamless manner.

1.3. Contribution Areas

The current work introduces a system that regards several scientific areas. Even though there are several approaches for each of the area separately, there is no approach in the literature that combines in the way we recommend all the features we mention. To the best of my knowledge there is not an implementation or approach of a decision support system that exploits the geographic and semantic information of urban data, uses machine learning techniques and subsequently visualize them in a web environment in the manner we propose.

Spatial Databases are databases optimized for storing and querying data representing objects in geometric space. They provide spatial indexing and efficient algorithms for spatial join functions. While the design and evolution of standard databases is oriented towards the efficient management and storage of various numeric and character data types, spatial databases require additional functionalities to address spatial data processing. Spatial measurements are required to calculate the length of a line, the area of a polygon or the distance between two geometries. Spatial functions have to be implemented along with geometry constructors to create or modify features and geometries, for example to generate a buffer around a geometry or an intersection of two distinct geometries. Predicates are also needed for querying spatial relationships with true/false results (e.g. do these polygons overlap or is there another polygon in a certain distance from a point of origin). In the present a spatial database is used that implements the semantic model introduced to facilitate the application of machine learning techniques to the data and endorse the creation of a decision support system.

Machine Learning is a scientific field that falls within the wider field of artificial intelligence. Its algorithms try to construct an approximate mathematical model based on some

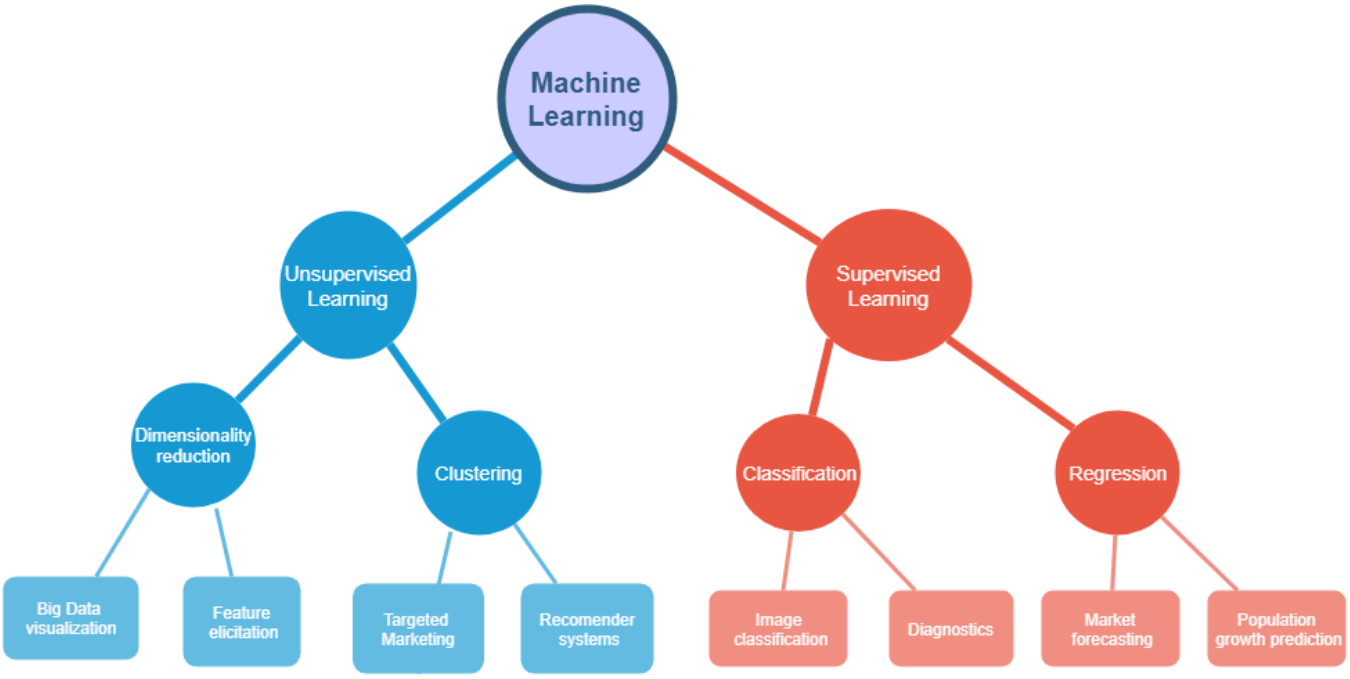


Figure 1 - Machine Learning areas

training data in order to conclude to a prediction or make a decision without explicit programming to perform the task. They are used in an extensive variety of applications, e.g. computer vision, email filtering, market forecasting, among others (Figure 1) and in general where it is practically impossible and infeasible to formulate a specific algorithm to complete the desired task. Therefore, the machine learning algorithms aim to exploit the training representative data and distinguish features of interest confiding in patterns and inference. The training examples however cannot cover all possible input combinations, hence the need for generalization, in order to be able to produce outcomes in cases not included in the training set. Experiments were conducted with various machine learning techniques and classifiers as well as with optimized versions of the latter followed by comparison and evaluation of the results and distinguishing of the predominant. The whole process will be presented in detail in a later chapter.

1.4. Thesis Organization

The thesis is organized in three parts. The first part contains the required theoretical background. More specifically, reference is made to developments and research in the field of 3d urban georeferenced scenes, spatial databases, semantic modeling, machine learning and multi criteria decision analysis. The last chapter of first part presents similar endeavors and state of the art in the respective fields.

The second part refers to the contribution of the current work to the relevant scientific

fields and the system implementation. An extensive description of the system, along with its technical features and capabilities are presented.

The third part involves the enrichment of the system with machine learning techniques, the experiments performed with several classifiers and their conclusions, while in the next chapter conclusions emerging from the whole of the system are discussed. Furthermore possible directions for future work are also presented.

Part I: Theoretical Background & State of the Art

2. GIS (Geographic Information Systems)

2.1. Introduction, definition and Global Geodetic System

The intense urbanization during the past recent decades has created complex urban structures, which are not sufficiently and clearly depicted by two-dimensional models. Thus, there is a need to identify a multi-layered and three-dimensional model in order to fully depict and record urban areas based on this three-dimensional reality (Figure 2). Technological developments are rapid and cutting-edge technological methods effectively support the optimization of multidimensional (nD) modeling and the pragmatic portrayal of reality in a number of applications.

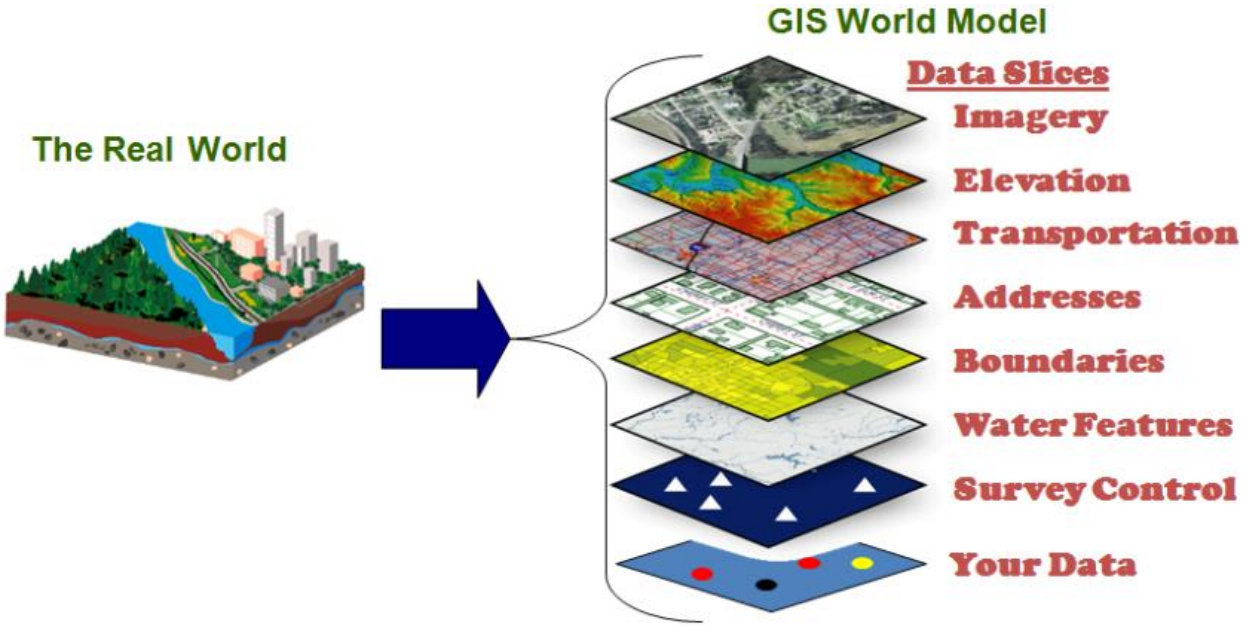


Figure 2 - GIS Layers

The modern tendency towards the creation of detailed three-dimensional models of urban structures which has been observed in recent years, has expanded to the field of Geographic Information Systems (GIS), broadening their main subject. In the field of GIS, a great research effort has been made over the last few years and important steps are being taken to create 3D city models, which represent the city with great precision, through a different approach that highlights not only its constituent parts and components, but also relationships and behaviors. Thus, evolution in three-dimensional city models incorporates in the geometric and topological model additional semantic information that models these relationships between the different elements of the city. One such model is the international standard CityGML, which

is considered the newest model for displaying real 3D information.

It is an object-oriented model, which presents 3D geometry, 3D topology, semantics and presentation / rendering at five levels of detail (Figure 3). CityGML not only presents the shape and graphic features of the city model but defines the semantics of its objects and

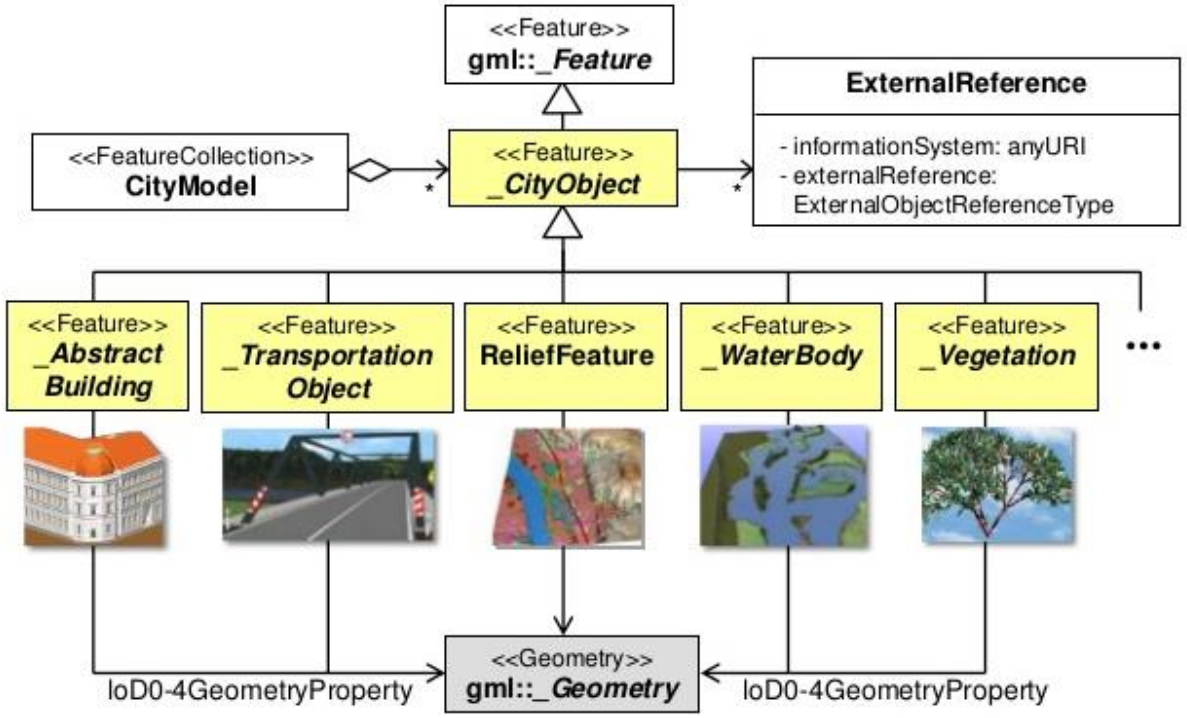


Figure 3 – Modeling in CityGML depicting the different layers of detail

visualizes its thematic properties, classifications and groupings.

Geographic Information Systems (GIS) have been in the epicenter of keen interest by the world during the last years. The need for logging, processing, storage of geospatial data has led to the creation of appropriate software and databases to perform these tasks. GIS technology includes hardware and software and is defined as a set of tools for collecting, storing, processing, analyzing, managing, retrieving and delivering real-world spatial data sets for serving specific purposes or decision making. The components that make up a GIS system are spatial data (digital map), logical operators (commands and functions), and a database that responds to various ways of retrieving information in order to answer questions concerning geographic space.

By the term spatial data we address the phenomena, observations or events related to space that can be encoded. In particular, spatial data in a GIS contains location information, topology and thematic features. They are resolved to points (pair of coordinates), lines (series

of pairs of coordinates), and polygons (closed path composed of a finite sequence of lines) and are structured at a hardware level and according to the user's perception, the way they are imported and how they are deployed in the database. The data may come from graphic information that includes existing maps, GIS files, processed or raw satellite images, rural data, or generated by non-graphic information including namespaces, statistical metrics, etc., and can be associated with correlation tables of graphical and non-graphic features. Each mapping attribute is an entry in an accompanying database and each entry may have multiple descriptive features. GIS has the ability to connect spatial graphic information with non-graphical information. The degree of connection gives the packet the ability to meet specific requirements. There is also the possibility of logical and numerical operations between maps.

The most developed GIS programs also include topology. Topology is a kind of cartographic logic, e.g. independent cartographic features "know" where they are, in relation to other characteristics, based on the geographic coordinates again. In non-digital maps there can be no topology. Many GIS users, typically those who want to use GIS for cartographic imaging, do not require topology. The most specialized users, those who want to pose queries on maps, compare between cartographic information levels or analyze market data usually require some level of topology.

2.2. World Geodetic System WGS '84

Large-scale GIS systems use the WGS '84 global geodetic system as point of reference and benchmark. WGS ‘84 is a system that has been defined based on the earth's mechanical properties and is the result of observations of various satellites using the Doppler measurement method. More specifically, we would say that it is a terrestrial reference system as the origin of the Cartesian coordinate system is the center of the Earth. The Z axis is parallel to the direction of the (conventional) earth pole (Bureau International de l' Heure, BIH). Axis X is defined as the intersection of the Greenwich meridian and the equator corresponding to the average Earth pole. The Y axis is defined to complement a clockwise rectangular system. So we understand that every point of the surface of the earth can be defined by the two coordinates, longitude and latitude (Figure 4).

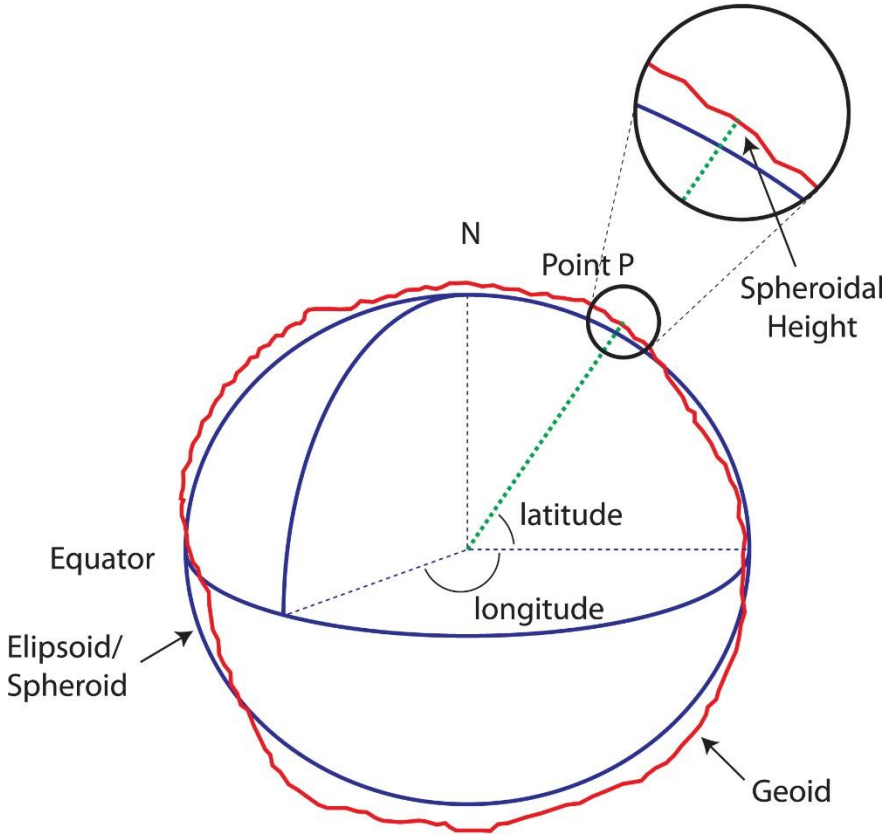


Figure 4 - WGS '84

The aforementioned system is used by cartography, geodesy and is considered the standard for global marine navigation. Also, the reference coordinates of this system are used in the Global Positioning System (GPS) developed by NASA and is widely known around the world.

2.3. Three dimensional representation of geospatial data

2D analysis of GIS is characterized by limitations in the visualization of specific situations such as urban planning, environmental audits, telecommunications, building design and landscaping. So we understand that the need for three-dimensional imagery is growing rapidly. The three-dimensional representation can be categorized in the following general categories.

In the first class we classify modelling of 3D objects using solid shapes. Bullets, cubes, and cylinders are used with a variety of parameters and functions such as junction, junction, and difference are used to capture a three-dimensional object. The main advantage of using

stereometry is the ease with which they are translated into images using PCs and the main disadvantage is that objects and their relationships can become very complex.

A second category of three-dimensional representation is the representation of the imprint, e.g. voxel. The voxel is a volume element (3D "pixel" which is represented as a three-dimensional cubic (or spherical) field). Each element may contain one or more data values. Voxels are mostly used to model contiguous phenomena such as geology, soil, etc. The advantages are numerous because the analysis is more detailed, but the main disadvantage of voxels is that high resolution data requires a large amount of space.

The third method for 3D data display is by using tetrahedrons (TEN) [1] imaging. A tetrahedron consists of four triangles that form a closed object in a three dimensional space with coordinates. It is the simplest 3D primitive (3-simplex) and it is relatively easy to create the proper functions that make it up. A tetrahedron object is also easy to use because it is well defined since the three points of each triangle are always at the same level. The significant drawback of this method is the use of multiple tetrahedra to construct an object.

The last three-dimensional object imaging method is boundary representation. The 3D object is represented by specific boundaries - elements such as vertex (0 Dimension), line (1D), polygon (2D), polyhedral (3D), which are organized and stored in data structures. The latter can be flat faces and straight edges such as border representations on a map or complex representations such as curved surfaces and edges. The main advantage of this method is that it represents to a large extent and with great precision objects of the real world. The boundaries of the objects are taken with measurements from the real world. A second advantage that can be mentioned is that most of the rendering engines are based on triangle representations with specific limits.

As a disadvantage we should say that boundary representations are not unique and that many objects may have the same boundaries, which forces us to use additional limitations and rules for modeling. So the process can be very complicated. For example, if we want to describe a triangle or a polygon (geometrically described), we must define constraints such as flatness, number of points and arcs, order of edges, relationship with adjacent neighbors etc.

2.4. Web GIS: Technologies and Its Applications

GIS technology has proven to be very useful in a number of areas, since spatial data can

be used in their proper form. Unfortunately, however, not everyone can access them in their raw form, which invalidates this usefulness. The solution to this problem is the Web GIS where the data are easily accessible by all users and can be made available without restrictions. Digital maps and point coordinates and landscapes are available from websites to the general public and entire platforms and applications are based on them as the application presented below.

With the development and growth of the Internet, the capabilities of GIS have also been developed. The web allows visual interaction with data such as a map. Maps published online are accessible to other users who may be able to update or evaluate them. A second feature offered by the internet is the easy access to geospatial data. Users can work from anywhere as long as they have access to the corresponding application site.

However, Web GIS also has flaws. Geospatial data requires large storage space and thus a long time to recover and display. Also, GIS technology is based on extensive graphic usage, which increases the cost of computer hardware that it manages, but also because of Internet connection speeds, can make graphic usage disproportionately slow for users. The latter, however, is an issue that tends to be eliminated with the expansion of the broadband internet access.

Web GIS uses the standard architecture of client / server with three levels. Client poses requests and the server processes them. The client program is a web browser and the server consists of a Web server, a GIS Web software and a database [2] (Figure 5).

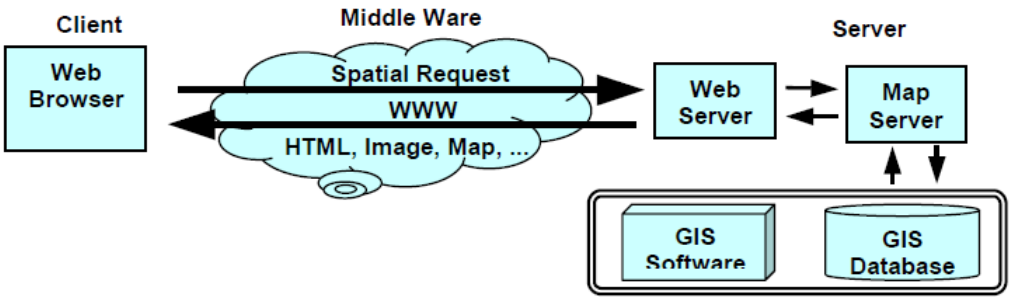


Figure 5 - How a typical Web GIS model works

The model depicted is widely used, where the GIS software is installed on the server, and the client connects to it and works on the interface that the server loads. The development of object-oriented programming has allowed faster execution of GIS software functions after using java classes, ActiveX elements and various plug-ins. Most work is done locally to the

client and the customer sees the results faster, since continuous communication with the server is not required. It should be recalled at this point that geospatial data are large in volume and difficult to manage.

2.4.1. Thin Client Architecture (Server Side Applications)

In the thin client architecture all data is processed on the server. The client sees only the interface the server offers it and all the workload is done on the server. Servers usually have great computing power and manage the entire GIS application. The client simply benefits of the server services.

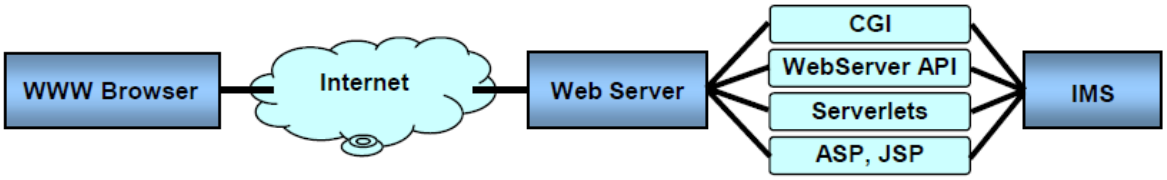


Figure 6 - Server side Applications

In Figure 6 the communication between the web browser, the Web server and the GIS server is represented. On the Web server side, we observe that it manages all functions such as CGI, Web Server Application Interface (API), Active Server Pages (ASP), Java Server Pages (JSP) and Java Servlet, while in the other side (client), the browser has no "jurisdiction".

The model has many advantages, first and foremost the fact that the end user is not forced to know the GIS architecture to utilize and benefit from its services. All the work is done on the server and its supervision is assigned to special application developers. Other advantages worth mentioning are: central control, ease of data rendering, low cost and integration capabilities.

The drawbacks of this architecture are presented below:

- Users may have different requests than those that the server can provide.
- Large amount of data managed by the server - Large database.
- Response time may be slow.
- Browsers who are not up to date may not be able to see the results sent by the server. Most vector-like data without additional browser programs cannot be displayed on the client.

2.4.2.Thick Client Architecture (Client Side Applications)

In the thick client architecture, unlike the thin client, the client acquires additional capabilities and decongestion to a degree occurs in the server. This architecture overcomes the disadvantage of not displaying vector data since they are now able to be processed locally after the browser functionality has been expanded (add-ons). The interface offered to the user has advanced from simply downloading documents to more interactive applications. The following figure (Figure 7) represents the new architecture.

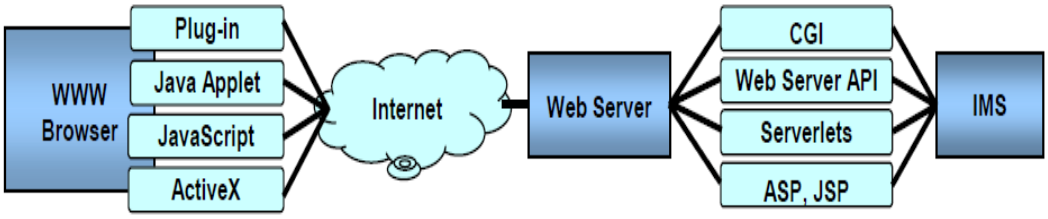


Figure 7 - Client Side applications

Important advantages of this model are:

- Documents and graphics templates are not required.
- Vector data can be used
- Image quality is not limited to GIF and JPEG
- Interfaces are modern and more functional.

The drawbacks of this model are:

- The database is partially stored in clients, so there exist data update and synchronization problems.
- Users must acquire additional software.
- The platform on the server may not be compatible with the client's browser.

2.4.3.Medium Client Architecture

To reduce the problems of the two previous architectures, an interim solution is proposed with the medium client architecture. This enables both client and server extensions to be used, and client computers may work more than the thin client architecture. The following figure (Figure 8) shows the comparison of the three architectures.

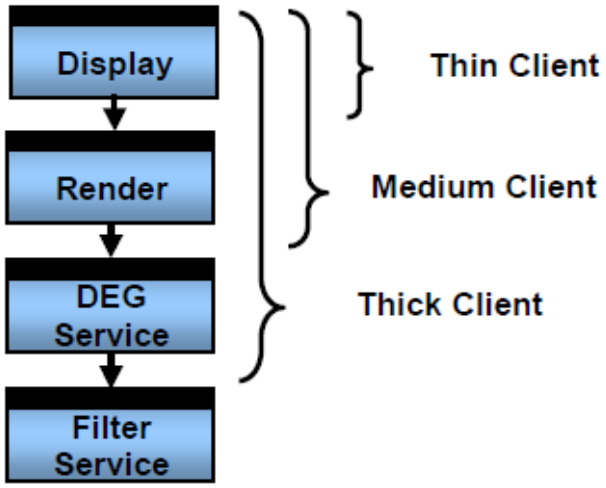


Figure 8 - Medium Client compared to Thin and Thick Client architectures

2.5. Chapter Conclusions

In this chapter the concepts relating to GIS (Geographics Information system) and their associated Web technologies as well as the issue of 3D Urban Georeferenced Scenes and three dimensional representation of geospatial data were analyzed, presented and illustrated.

3. 3D Urban Georeferenced Scenes

3.1. 3D City models

With the rapid development of GIS technology, the need for introducing semantic features into 3D city models has been born [3]. Semantic enrichment is considered particularly important because of the complex and multiple use of space within the multidimensional urban environment, while semantic modeling of cities requires appropriate data [4].

3D urban models represent spatial and geo-referenced urban data including land, buildings, vegetation as well as road and transport models. Generally, these models serve to present, investigate, analyze, and manage urban data. As a feature, 3D city models allow the visual incorporation of heterogeneous geo information into a single framework and, therefore, create and manage complex urban areas of information [5].

Moreover, modern tendencies focus on the semantic enrichment of distinct city objects or geometries that can be analyzed in their structural elements, integrating properties and relationships between them, and dealing with issues of spatial-semantic coherence, even for complex building models. Semantic modeling along with 3D geometry and topology of real objects can be implemented through the open model CityGML, which is a 3D semantic model that not only visualizes the shape and appearance of city models, their semantics objects and the depiction of their thematic properties but also their classifications [6]. However, two of the main concerns associated with 3D modeling and semantic integration, especially for 3D urban model applications are: (a) what are the optimal semantic representations within an overall city model or models of separate buildings, and at what level, and (b) what kind of semantic information is required for the satisfactory depiction of the different concepts of 3D city models [7].

In urban 3D models, different terms are used: Cybertown, Virtual City, or Digital City. 3D urban models are essentially digital models that include the graphical representation of buildings and other objects in 2.5 or 3D [8]. Three are the basic geospatial approaches used to create virtual 3D city models. In the first, conventional techniques such as vector map data, Digital Imaging Models (DEM) and aerial photographs, the second is based on high resolution satellite images with LASER scanning, while terrestrial images are used in the third using photogrammetry with Digital Surface Models (DSM) and texture mapping. The possible applications areas of 3D city models (Figure 9) as identified in [8] are:

- Urban planning.
- Commercial activity of the city.
- Public safety.
- Disaster management.
- Power supply and energy planning.
- Event management.
- Environmental management.
- Real estate market.
- Transport and navigation.
- Traffic management.
- Virtual tourism.

Despite the great improvement in technology over the past years, it remains difficult to find the funds and resources to begin the actual creation of 3D city models. It is obvious that 3D city models have a positive impact on a large number of governmental and administrative projects and processes: better communication, improvement in design and construction, better completion of projects, reduction of risks, etc, with positive overall result for citizens, state and business. The cost of creating and keeping up to date a 3D city model is a difficult question, which depends on a large number of factors. Most cities start with a small / partial model with a relatively low level of detail (Level of Detail 1 or 2), while upgrading sections of the city to a higher level (level 3 or 4) for the needs of specific projects in some areas. This is a way to gradually turn the city model into a more sophisticated, "Smart" 3D model, keeping costs at reasonable levels. During the 3D modeling process, relevant organizations seek and create standards based on open standardization structures (e.g. CityGML) and corresponding spatial databases [9].

3D modeling of the city is broader than 3D visualization of reality, which simply involves the geometry and imaging of entities. In the past, virtual 3D city models have been used, mainly for the visualization or graphic exploration of urban landscapes. In recent years, however, the ever-increasing number of new applications in urban planning, facility management, environmental and educational simulation, risk and security management, and personal navigation require additional information on city objects. In the fields of application of 3D modeling, the cadastre can also be used, which can exploit the potential of a city model as a

basic structure in Land Administration Systems (LASs). The concept of semantic modeling, in addition to geometry and imaging, which do not possess information about the meaning of objects, introduces the ontological structure that includes thematic levels, characteristics, and relations between them.

Beyond topology and geometry, therefore, the thematic semantics of 3D objects should also be established. In the semantic 3D model of a city, the related objects of the urban landscape are classified and their spatial and thematic properties are described, according to the definitions and functions of the objects that have been identified. As a result, a semantic 3D model is the basis for urban information modeling. For 3D urban models there are very few thematic semantic models. Buildings and land objects are the most important features for describing a three-dimensional city model. Based on this, the current version of CityGML, which includes thematic semantics as well as 3D geometry and topology, has included the surface of the ground and objects above the ground. Other semantic models have also been created and accepted as standards, such as the North American Data Model and Geology Science Markup Language (GeoSciML) for geological observations. Thus, semantic 3D urban models, apart from spatial and graphical elements, mainly include the ontological structure with their thematic classes, their properties and their interdependence. Therefore, the objects of space are decomposed into

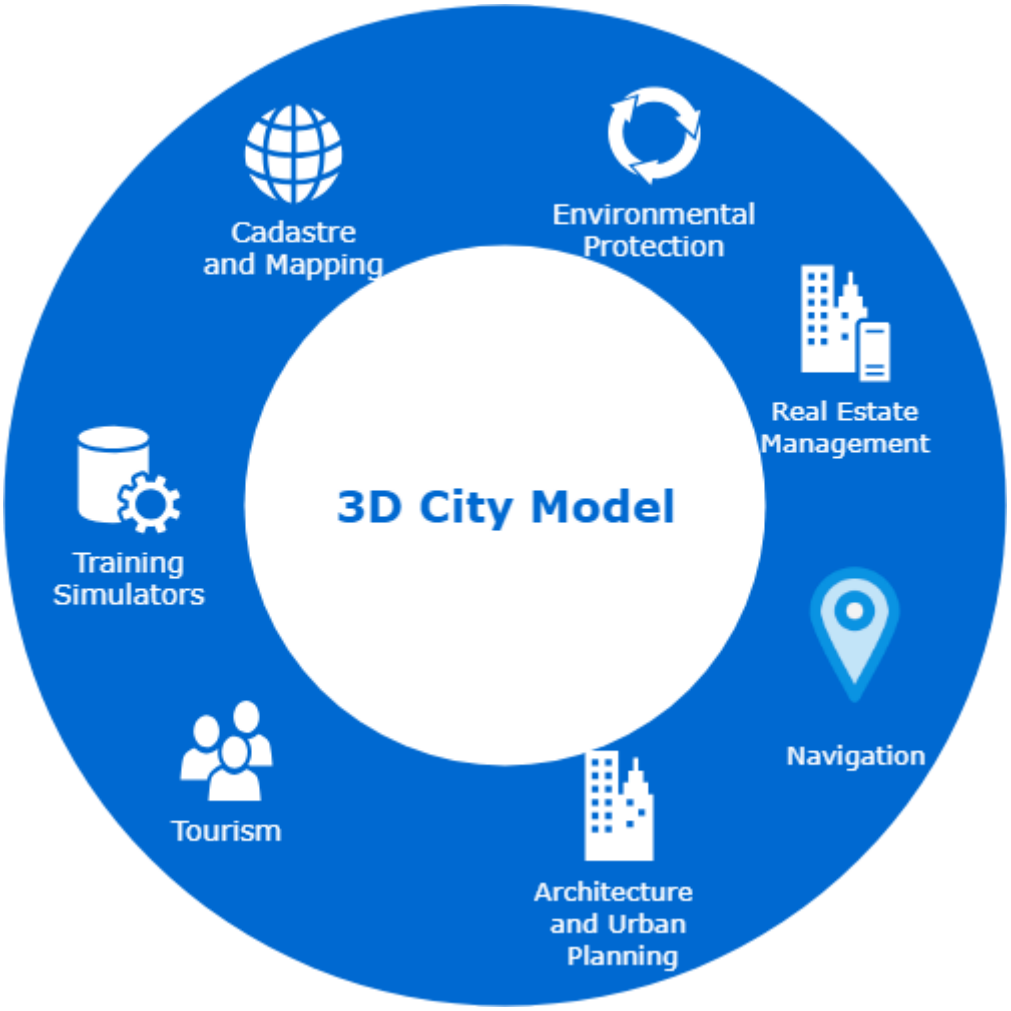


Figure 9 - Urban environment areas connected with 3D city models

the parts of which consist of logical criteria and structures, as they are observed in the real world.

The 3D semantic models of cities, constitute a new concept in city models, and combine spatial and graphical information with the ontological structure of urban space, its particular characteristics and the interactions of these characteristics. In these models, space objects decompose into parts of which they are based on reasonable criteria and it is obvious that the creation of such a semantic 3D city model requires the availability of appropriate 3D data.

This can be done either automatically or manually, but in any case a great deal of effort is required to create and maintain such a 3D city model. From an economic point of view, the profitability of a semantic model of a city depends on whether the data and especially its semantic information can be used by various users and in many applications. This also led to

the need to create a common model of information for different uses and applications, and this context includes the development of CityGML, which evolved through research into the integration of semantics, geometry and topology into 3D modeling. This is the essence and purpose of developing CityGML, to put a common definition of entities, attributes and relationships within a three-dimensional city model. By providing a basic model with entities covering many branches, the city model can be a central axis on which different applications are supported. By using these three-dimensional city models, the information that is needed to make strategic decisions on issues of concern for the proper functioning of the city can be drawn.

3.2. 3D City modeling topics

3D City model management: Because of the different sources used to develop 3D city models, it is necessary to standardize in describing, storing and exchanging between models. This is the reason for why designing and constructing CityGML [10] as a common information model for the visualization of 3D urban objects, which includes geometric, topological and semantic information. The vast number of data from 3D city models requires powerful databases, such as Oracle and PostgreSQL, which can be used to store and manage these data. Some databases support spatio-temporal data types or even CityGML formats (such as Oracle 11g), while managing even more data volume can be addressed with cloud computing.

The semantic information of 3D city models: in many applications, such as urban planning, facilities management and navigation, in addition to geometric models, semantic information is needed for city objects. It is therefore important to develop methods for modeling, imaging, managing, formulating spatial queries and analyzing the semantic information of 3D city models [11]. In addition, other related technologies with the 3D city models [12] must be sought and developed new applications, such as:

- Analysis methods based on 3D city models, such as visual analysis, noise analysis and optimal path analysis.
- Mobile phone usage: Many users would benefit from having access to a 3D city model in their mobile phone environment. This feature is particularly useful in cases of security, facility management, environmental quality and monitoring of municipal / community projects.
- Publication on the Internet: since accessibility is important in 3D city model, the use of these models should be done through web-based services. This feature,

due to the complexity of some 3D models, poses server performance issues and broadband network performance.

3.3. Urban Planning

Urban Planning, with its main object of shaping the city and its public space through space synthesis processes, has to take into account and combine many different spatial components that are involved in the creation, such as the architecture of the buildings and landscape, uses and functions of the site, economic viability of activities, environmental protection and development [13].

As early as the early 1990s, smart city appears to show the shift of urban development towards new technologies, innovation and globalization. The World Foundation for Smart Communities has supported the use of information technology to address new challenges in the global economic vision. However, the latest interest in smart cities has emerged through intense environmental concerns and the emergence of new online technologies such as smart phones, cloud computing, the semantic web as an extension of the current web etc. that promote the interconnections between real world users. A new spatial concept for cities is linked to multiple concepts as shown below [14]:

- Cyber cities, cyber government.
- Digital cities, digital city depictions, digital city transport and simulations.
- Intelligent cities, collective intelligence of the inhabitants, crowdsourcing, online collaboration, broadband access for innovation, urban capital, cooperative learning and innovation.
- Smart cities, mobile devices, sensors, embedded systems, smart gauges, smart environments, and equipment provided by city intelligence.

3.4. Smart Cities

The concept of a smart city from a technological point of view has some specific properties, as shown by the relevant literature, which are related to cyberspace, the digital environment, but also the recent achievements of IT, wireless networks and other technologies incorporated in the natural space of the city (Figure 10).

The emphasis on smart embedded devices creates innovative systems that combine

intensive-knowledge activities, and web based collective intelligence applications. It is anticipated that intelligent city solutions with the help of equipment and interfaces of mobile devices and sensors that will allow the collection and analysis of urban real-world data will improve the capacity of anticipating and managing urban flows and will give impetus to a collective intelligence of cities [14].

Figure 10 - Smart City infographic

How a city is defined as "smart" depends on the different perspectives involved in its development and management process. In any case, from a technological point of view, "intelligence" means understanding, learning and self-awareness. Indicatively, the following approaches for the smart city are mentioned:

- A city with satisfactory performance in a long-term course in the economy, governance, mobility and the living environment, built on the intelligent combination of its assets and its determinant activities, independent, with conscious citizens.
- A city that monitors and integrates the conditions of all its major infrastructure, including roads, bridges, tunnels, subways, airports, ports, communications, water, power, and even large buildings, which can optimize its resources, design prevention measures and activities, and improve safety issues while maximizing its services for the benefit of its citizens.

- A city linking physical infrastructure, information technology infrastructure, social infrastructure and business infrastructure to harness the collective intelligence of the city.
- A city that tries to make itself "smarter" (more effective, sustainable and fair).
- A city that combines ICT and Web 2.0 technology with other organizational, design and research efforts to alleviate and speed up bureaucracy processes and to help identify new, innovative solutions for the complexity of city management, with the aim of improving its sustainability.
- The use of smart IT technologies to make their important parts of infrastructure and city services (including city administration, education, health care, public safety, real estate, transport and utilities) more intelligently, interconnected and efficient [15].

A necessary step towards implementing a smart city is data mining and analysis on data traced through the dynamics of a city. The trajectories of people, vehicles and other moving objects can provide important information about the overall mobility of the city. To this aim, the Global Positioning System (GPS) along with other related tracking technologies, facilitates internal and external navigation in a smart city.

3.5. Chapter Conclusions

In this chapter 3D city models and urban planning topics and challenges were presented. Special mention was made to smart cities and perspectives of intelligent infrastructures.

4. Decision Support Systems

4.1. Introduction

The decision-making process in modern organizations and businesses is often an extremely complex and inadequately structured process, usually assigned to a group of people who represent different functions. During such a process, decision-makers are asked to propose and consider a variety of alternatives, taking into account their expected short and long-term effects on the organism. Effective handling of the contentious issues requires a thorough discussion on different approaches and mutual cooperation between the individuals involved. It is also known that decision makers adopt and propose action plans based on their position in the organization and the individual objectives they are called upon to serve. Furthermore, the relevance and value of the proposed action plans differ, and often arises a need to develop defending arguments on them. This is due to possible different interpretations of the problem and on the other hand, due to the competitive or even conflicting interests, objectives, priorities and constraints involved in such a process [16]. In practice, working groups that are called upon to make a decision have to address the problem with inadequate or excessive information, and the time needed to acquire or process the necessary knowledge is often prohibitive. Moreover, the significance and value of existing codified knowledge varies according to the role and objectives of each decision-maker. Therefore, it is necessary to establish common points of reference for representing the problem, assessing the current situation and defining the objectives to be achieved. Consequently, it is necessary to scatter the communication barriers and to provide techniques for structured analysis of decisions, systematically guiding the drafting, timing and content of relevant discussions. It is also necessary to increase and exploit the flow of knowledge. In particular, the knowledge inherent in all the resources available to an organization, namely people, structure, culture and processes [17].

Decision Support Systems aim to facilitate communication between decision-makers and support the expression of the views and positions of decision makers in a way that is commonly accepted and understood, while providing the necessary technical infrastructure to support this communication [18]. This is done through a series of techniques for formulating and gradually building the decision.

Furthermore, they are a combined approach to manage decision-making with IT tools and techniques [19]. More specifically, they are the result of the development of two fields of

studies, that on decision-making of an organization or business by Simon, Cyert, March and other researchers at the Carnegie Institute of Technology in the late 1950s, and the technical works by Gerrity, Ness and other researchers at the Massachusetts Institute of Technology followed by the development of interactive computer systems in the early 1960s [20]. A widely accepted definition describes Decision Support Systems as a computer software which accepts as input data a large number of events and methods to convert them into comparisons, graphs and directions in a sense that facilitates and expands the capabilities of the decision maker. [21]. Such systems support decision-making by helping to organize and manage knowledge in problems which fall to the structured or semi-structured categories. They may provide one or more of the following types of support [18]:

- Indicating the need for a decision,
- Recognizing the problems that need to be solved,
- Problem solving
- Facilitating or extending the ability of users to process knowledge,
- Offering of advice, instructions, estimations, expectations, facts, and plannings
- Stimulating the perception, imagination and creativity of decision-makers
- Guiding or facilitating interaction between decision makers

4.2. Decision Making

A decision-making process includes the determination of the objectives of the decision, the alternatives that serve the objectives to be achieved, and the criteria for evaluating available options [22]. The most widely known and used model of decision-making comes from Simon and constitutes the widely accepted formalism of all the actions required to make a decision [23]. This model separates the decision-making process into three successive stages: intelligence, design and selection.

First of all, at the intelligence stage, it is recognized that a problem should be resolved or an opportunity has been presented and should therefore be investigated. More specifically, this stage involves clarifying the exact status of the organization and its surroundings,

recognition of the general problems or opportunities and the gathering of the data necessary for the problem or opportunity in question. For that purpose it is necessary to collect the relevant explicit or implicit knowledge of the decision makers concerning the subject under consideration.

In the second stage, (design), decision-makers consider the problem's data as a whole in the decision-making environment and choose the method and the criteria on the basis of which the final decision will be taken. This stage usually involves a series of discussions through which group members present their views. The third stage is the choice of a solution using the criteria or pre-defined conditions. For this purpose, a variety of methods can be used which come from the fields of Operational Research and Multicriteria Decision.

Based on the above model, efficient and effective decision-making should be based on continuous reassessment of the problem parameters and the correction of any errors identified during the process of evaluation. It is also important to record any errors and / or omissions in the process so that they will not be repeated in future cases. For this reason, decision making is treated as an iterative process with feedback loops between the three stages. It should be noted that the implementation of the above procedure in practice may vary from case to case with respect to the emphasis assigned to each step, which depends on the urgency of each situation, the availability of the necessary data, the importance of the decision for the organization, etc.

4.3. The Structure of a Decision Support System

In general, the design of a Decision Support System concerns the development of Databases and tools for managing them, which provide access to internal and external data, information and knowledge, models for analysis and / or decision making and interfaces that allow for interactive searches, reports and graphical representations related to the decision [24].

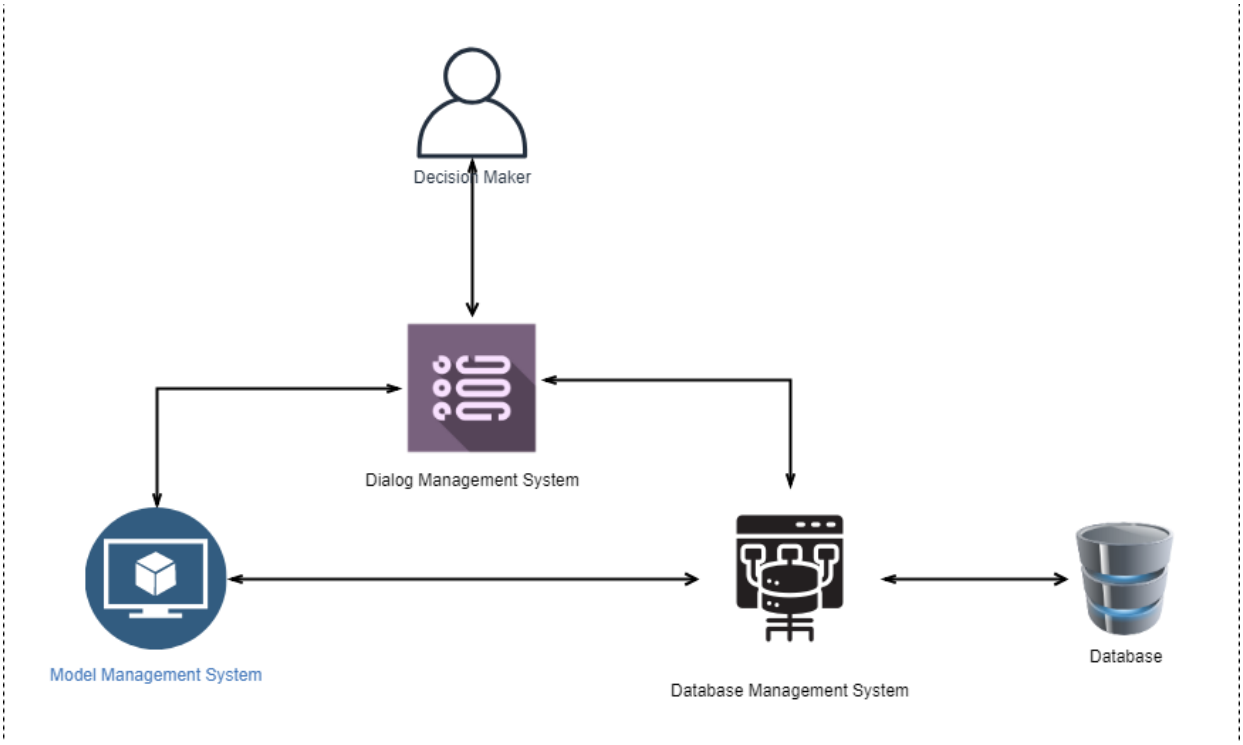


Figure 11 - Diagram of the basic modules of a Decision Support System

Figure 11 shows a diagram of the basic modules that constitute such a system. This is made up of the Dialog Management System, the Model Management System, the Database Management System, and the Database.

More specifically, the Dialogue Management Subsystem is responsible for presenting the information outputs of the Database Management System and the Model Management System to the decision maker and vice versa for the introduction of requirements and user decisions as inputs to them. Since the Management and Dialogue Management System is the one that allows the user to communicate with the system, it is considered to be the most important part of the system, because it determines how easy and efficient the system management is, and therefore the exploitation of the capabilities it provides. For this reason, it should be designed to represent knowledge and control system functions through suitably designed user interfaces. It should also be user-friendly and help with the features it supports.

The Model Management Subsystem is that part of the system that manages decision models and supports the decision-maker with relevant analysis methods and evaluation algorithms. The main function of the Model Management System is the management of mathematical and financial models for the analysis, processing and evaluation of decision problems.

The Database Management Subsystem is that part of the system that manages the database. Through this, the user gets access to the information they need to form and analyze the decision. The Database Management System must be capable of managing internal and external data of the organization. Furthermore, the Database Management System must be capable of informing the user about the available data forms and how to access them. As far as the Database is concerned, this is a collection of record files and files that are organized to serve a particular purpose. For example, a Database may contain a set of financial files and / or data for customers and suppliers that contain decision-making information.

4.4. Applications of Decision Support Systems

The Area of Decision Support Systems has demonstrated a multitude of applications for the needs of organizations after the first wide-use approaches of the 1970s [19]. Part of these applications generally approach the problem of decision-making, either through alternative theoretical approaches or by using different technology. Others have been developed to be applied to specialized problems and work environments.

During the 1980s, applications were developed that targeted specific categories of users. The most important development in the field of Decision Support Systems was the transition from systems that supported decision making by an individual user to systems that allowed the communication and collaboration of a user group. More specifically, Executive Information Systems (EIS) have been reported to offer support to business executives, Group Decision Support Systems (GDSS), and Organizational Decision Support Systems - ODSS). Group Decision Support Systems support group decision-making with technologies that can be distinguished according to the time, location and level of support of the group of decision makers. In particular, a Group Decision Support System can allow the synchronous or asynchronous, remote or in the same communication of the members of the decision-maker group. It can also support group members at an individual or collective level.

In recent years, following the evolution of relevant technology and information systems, intelligent models have been developed that aim to support solutions in urban design problems (e.g. better energy consumption of buildings) and aim to utilize data from sensors, recorders and actuators of individual systems to guide the decision maker in developing short-term action plans. This adds intelligence to these systems, since they enable the decision maker to administer real-time acquired data.

4.5. Chapter Conclusions

This chapter included an analysis of Decision Support Systems and its purposes, the process of Decision making and described the structures and subsystems that constitute such a system.

5. Spatial Databases

5.1. Overview

As a result and natural consequence of the rapid development in the geographic systems area and the continuous increase of the amount of data needed to be managed by them, along with the corresponding developments in the field of relational databases, it became evident the need for the creation and the use of spatial databases .

A spatial database is a database that is enhanced to store and access spatial data or data that defines a geometric space. Alongside traditional data (text, numbers) spatial databases can store data as coordinates, points, lines, polygons and topology. Some spatial databases handle more complex data like three-dimensional objects, topological coverage and linear networks (Figure 12).

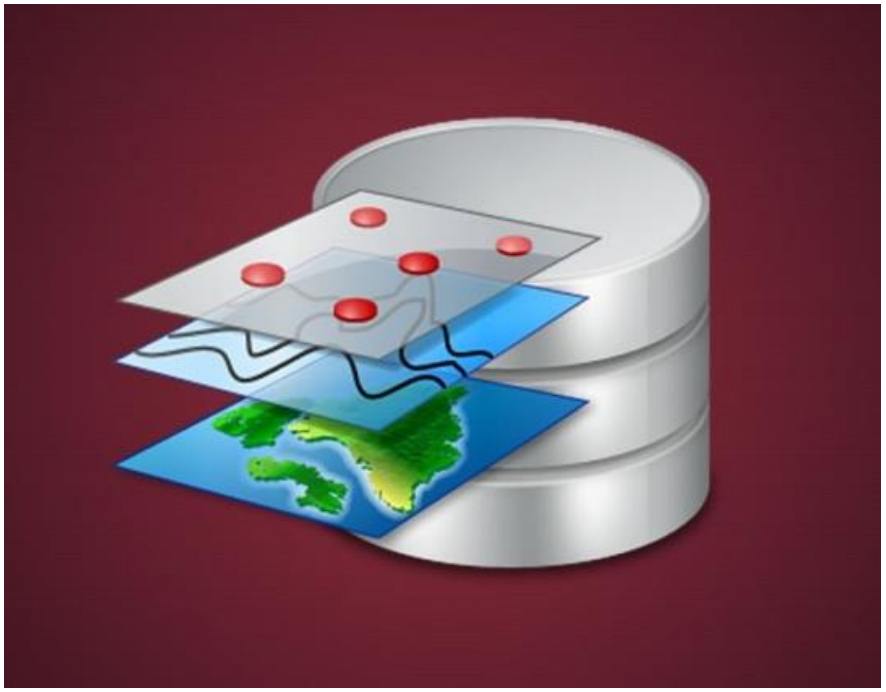


Figure 12 - Spatial databases combining handling multitude of data

5.2. Data Representation

To better understand the particularity of spatial databases, in the figure below (Figure 13) the diversity of the data and features to be stored can be seen. In GIS terminology, real-world features are called spatial entities. The total space is treated as a set of subspaces. Each subspace hosts a data category and differs from others based on the thematic dimension. Vector

and raster are two different ways of representing spatial data. However, the distinction between vector and raster data types is not unique to every GIS.

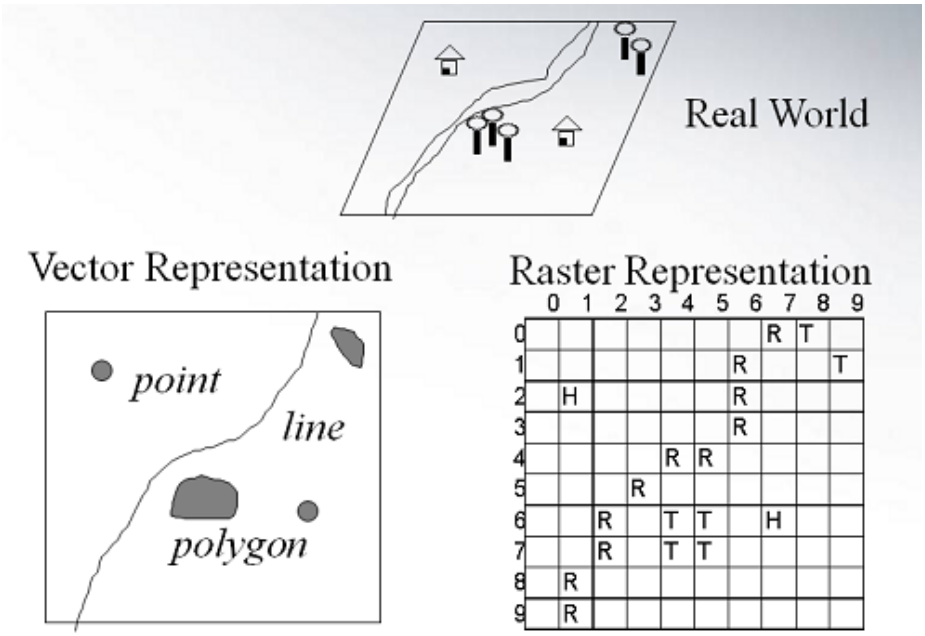


Figure 13 - From the real world to ...maps

For raster data the entire area of the map is subdivided into a grid of rudimental non-overlapping cells. Each cell is uniquely identified by a thematic feature (e.g. land use). To achieve that, each cell contains the dominant value of that cell which correlates to the nature of whatever is present at the corresponding location on the ground. Raster data can be thought of as a matrix of values. The cells can also be multi-dimensional. Raster datasets are intrinsic to most spatial analysis.

Grid according to the field of application and the data for representation may be either regular or irregular (Figure 14).

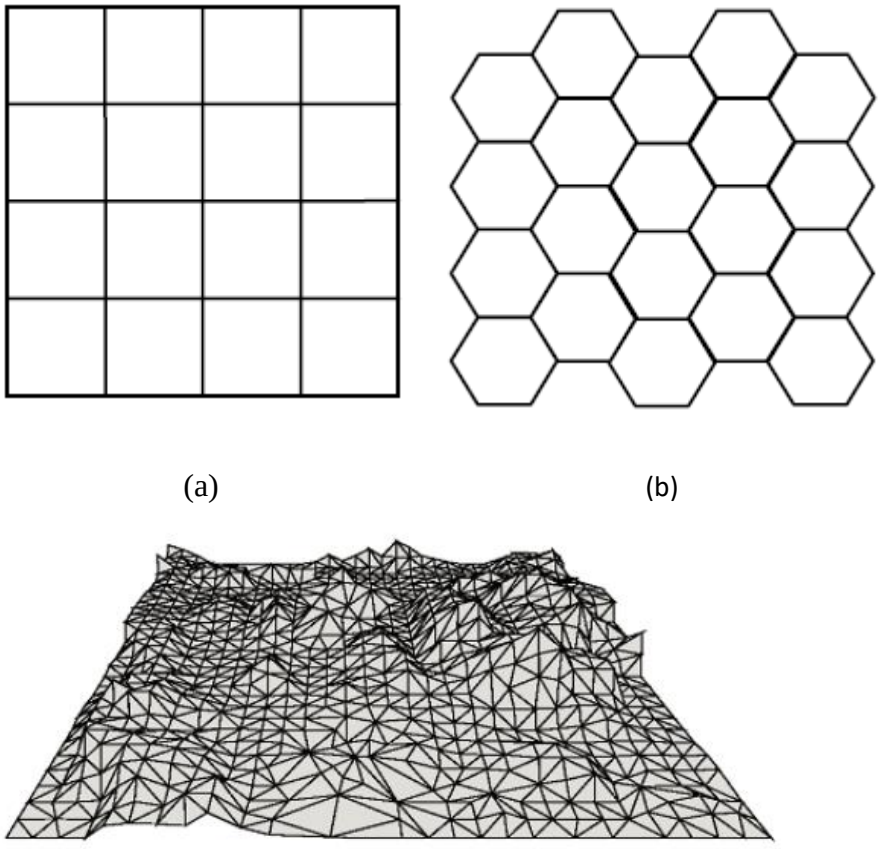


Figure 14 - Regular grid (a), (b) and irregular grid (TIN) (c)

Vector data representation is a coordinate-based data model that represents geographic features as points, lines, and polygons (areas). Each point feature is represented as a single coordinate pair, while line and polygon features are represented as ordered lists of vertices. Attributes are associated with each vector feature, as opposed to a raster data model, which associates attributes with grid cells.

A vector point is a zero-dimensional geometry that occupies a single location in coordinate space. When features are too small to be represented as polygons, points are used.

Vector lines can be described by a sequence of coordinate pairs defining the points through which the line is drawn and connect vertices with paths in a particular order. They usually represent features that are linear in nature, such as roads or rivers, or they can also depict artificial divisions such as regional borders or administrative boundaries.

A vector polygon feature is created when a set of vertices are joined in a particular order and closed (the first and last points joined to make a complete enclosure) (Figure 15). In order to

create a polygon, the first and last coordinate pair has to be the same and all other pairs must be unique. Polygons represent features that have a two-dimensional area. Examples of polygons are buildings, agricultural fields and discrete administrative areas.

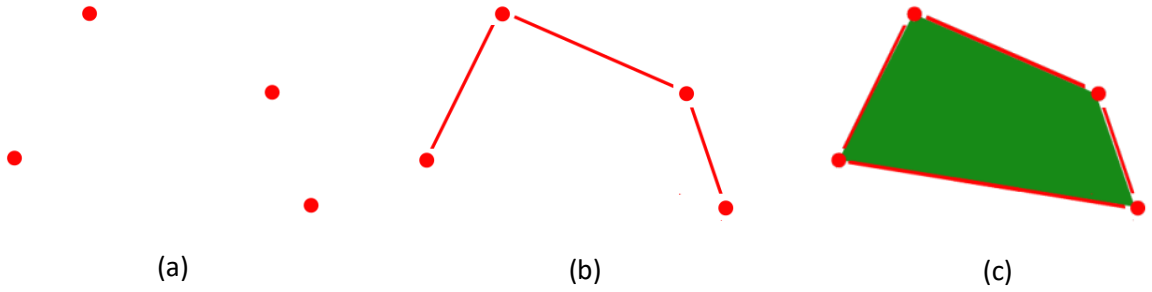


Figure 15 - Vector points (a) , line (b) and polygon (c)

5.3. Spatial Data Types

There are two types of spatial data. The geometry data type supports planar, or Euclidean (flat-earth), data and the geography data type, which stores ellipsoidal (round-earth) data, such as GPS latitude and longitude coordinates.

The geometry and geography data types are divided into instantiable and non-instantiable data types. As the figure indicates, the instantiable types of the geometry and geography data types are Point, MultiPoint, LineString, Polygon, MultiPolygon, MultiLineString and GeometryCollection. The subtypes for geometry and geography types are divided into simple and collection types (Figure 16). Simple types include:

- Point
- LineString
- Polygon

Collection types include:

- MultiPoint
- MultiLineString
- MultiPolygon
- GeometryCollection

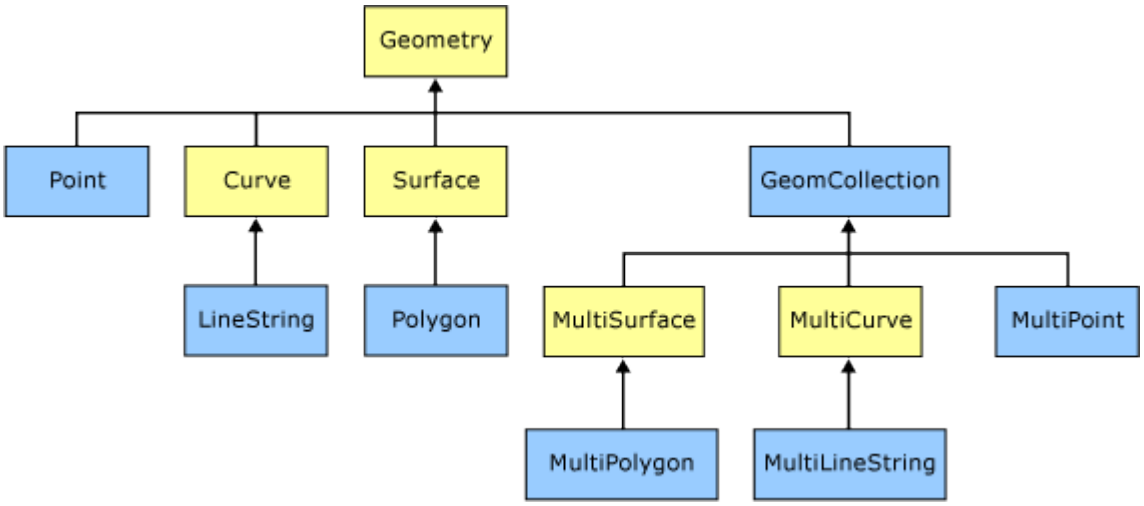


Figure 16 - Hierarchy of OGC geometric shapes

A particular reference should be made for the differences between the geometry and geography data types. The two types of spatial data usually and in most cases behave quite similarly, but there are some key differences in how the data is stored and manipulated. The connecting edge between two vertices in a geometry type is a straight line. However, the connecting edge between two vertices in a geography type is a short great elliptic arc between the two vertices. A great ellipse is the intersection of the ellipsoid with a plane through its center and a great elliptic arc is an arc segment on the great ellipse.

In the planar, or flat-earth, system, measurements of distances and areas are given in the same unit of measurement as coordinates. Using the geometry data type, the distance between two points is always their Euclidean distance and is measured in units, regardless of the Spatial Reference System or units used.

In the ellipsoidal, or round-earth system, coordinates are given in degrees of latitude and longitude. However, lengths and areas are usually measured in meters and square meters, though the measurement may depend on the spatial reference identifier (SRID) of the geography instance. The most common unit of measurement for the geography data type is meters.

5.4. Spatial data methods and functions

One of the main reasons spatial databases are used in combination with GIS systems is the inherent support of basic spatial analysis methods and functions. Basically, it is addressed at a viewpoint of practical use of spatial databases by experts and all types of users. They provide

relatively easy spatial analysis capabilities.

The basic methods are mostly used to specify properties of a single geometry. Representative examples could be the discovery of the dimensions of an existing geometry (point, line or surface), or finding the geometry type and its SRID.

Spatial analysis functions make calculations based on geometry. The majority of the functions return new geometries, while some are applied to pairs. The following table shows the capital ones. They are implemented through functions (Table 1).

Distance (a,b)	Shortest distance between shapes
Length(g)	Line length / ring perimeter
Area(g)	Surface area
Intersection(a,b)	Intersection of geometries (a AND b)
Union(a,b)	Union of geometries (a OR b)
Difference(a,b)	Difference of geometries (a AND (NOT b))
Centroid(g)	Calculates and returns the centroid of the geometry. In most cases it coincides with the mass center.
ConvexHull(g)	Returns the convex geometry enclosure

Table 1 - Fundamental spatial analysis functions

By using these features the spatial database can internally and in a transparent way perform extremely complex geometrical calculations and queries across multiple geometries.

5.5. Topological relations

Topology can be formally defined as "the study of qualitative properties of certain objects (called topological spaces) that are invariant to any continuous deformation of space. The topology simplifies analysis functions, audits relationships between entities based on their positions in space. The correlation types between entities may be topological (e.g. meet, overlap, intersection),

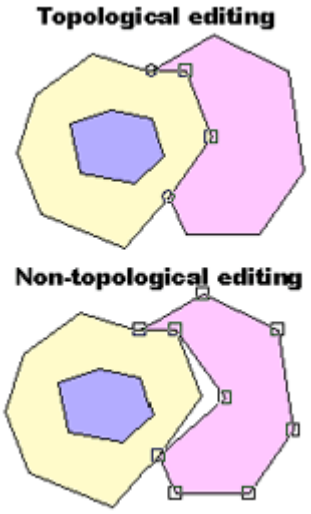


Figure 17 - Topological editing

directional (north, east, top, right), or measuring associated (near, away from etc.) (Figure 17).

A very interesting method for the classification of topological relations was proposed by Egenhofer in 1993, which was subsequently embraced by most geospatial database systems.

5.6. Spatial indexes

Common database systems use indexes for a faster and more efficient search and access of data. These indexes, however, are not particularly fit for spatial queries due to the nature of spatial data. Instead, spatial databases use a unique index called a spatial index to speed up database performance. Spatial indexing is very much required because a system should be able to retrieve data from a large collection of objects without really searching the whole bunch. It also supports relationships between connecting objects from different classes in a better manner than just filtering (Figure 18).

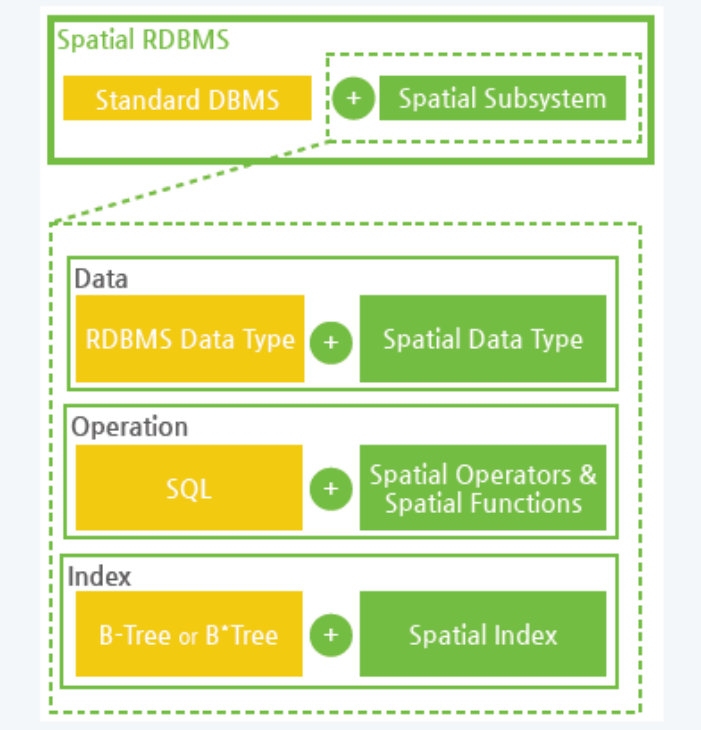


Figure 18 - Spatial DBMS concept

5.7. Chapter Conclusions

The fifth Chapter focused on Spatial Databases, the way they represent spatial data types, and the reasons that make them prevail over the other alternatives are presented, as their ability for topological relations and spatial indexes was discussed.

6. Semantic Modeling

6.1. The concept of semantics

Semantics is the study of meaning. In the context of the science of linguistics, semantics is used to describe the meaning of the words, terms and phrases people use to communicate. This concept can be extended to computer science, describing the meaning of words, terms and phrases in programming languages. The techniques used in computer science make this type of definition much narrower and more precise than for human language and therefore, by extension, in information sciences, the term semantics describes the technical relationship between data and its context or ontology. In the context of the web, the term semantics describes the relationship of their data and metadata. The aim is to make both understandable by machines through formatting and structuring them in standardized ways. Semantic content is already on the Web, but machines are not able to uncover it or use it consistently. Better understanding of semantic will enable professionals of several related fields to make better use of existing web technology and achieve better interoperability.

Linguistic science has two other areas of research next to semantics: Syntax and pragmatism [25]. The syntax describes the rules by which terms and words can be constructed in sentences and phrases. It is possible to create syntactically correct proposals that do not hold semantic essence. Noam Chomsky created the phrase "revolutionary new ideas appear infrequently, colorless green ideas sleep furiously" [26] as an example that a syntactically correct sentence does not equate with meaning. Several interpretations of this proposal have been undertaken to show that it could make sense in specific contexts. Especially when talking conversely (with the addition of a frame or metadata), "colorless" can be interpreted as "dull" and "green" as "young" or "fresh". By giving a brief introduction to the reader, it would be possible to unfold some meaning from the otherwise silly proposal. This indicates that the frame of any element has a profound effect on the relevant meaning. This is the third pillar of studies in Linguistics: Pragmatics. It is the relationship between the term or the word and the observer. This highly interesting aspect of Linguistics has not been explored in vastness in information technology which may be one reason why semantics do not hold ample diffusion to many practical aspects of the Web.

6.2. Ontology

Ontology is the study of the nature of existence and its relationships. It is a branch of philosophy that analyzes what exists or can be assumed to exist, how these entities can be grouped and linked together. Typically relationships are grouped and subdivided into hierarchies according to similarities and differences. In computer and information science, ontologies are an official representation of knowledge of different domains [27]. Ontologies can be used to describe the domain in an official way. Relationships between domains can also be described in ontologies. Ontologies are formal, clear specifications of the common concepts that provide a vocabulary of

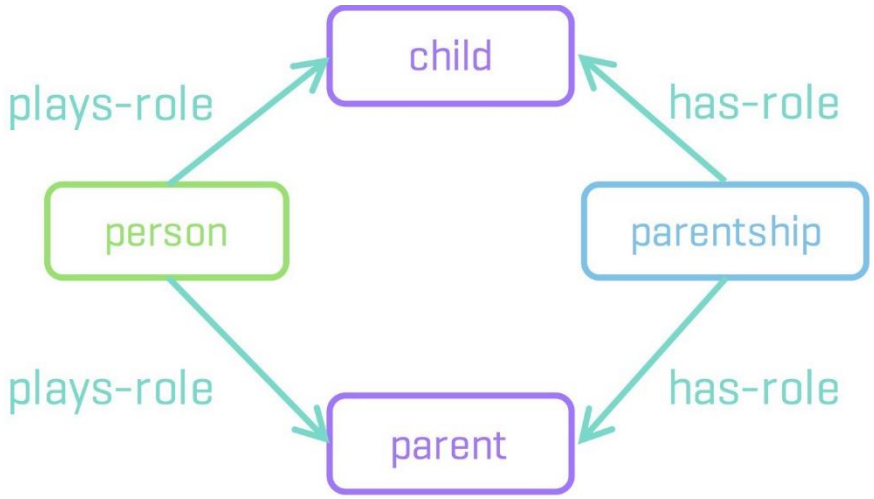


Figure 19 – Ontology example

a certain semantic meaning. Vocabulary can be used to model a domain with a defined syntax, describing the type of objects, properties, and relationships. A short explanatory example is presented in Figure 19.

Ontologies can be described officially using different templates and languages, for example Owl (OWL). For the content of this article we will not go into more detail, but first we will have an overview of the technologies already in use.

6.3. Geospatial Semantics

Geospatial Semantics comprises a research area spanning over several fields, involving spatial databases, GIS, Semantic Web, Artificial Intelligence (AI) and functions of cognitive science [28]. The aforementioned field of semantics uses a variety of methods ranging from top-down knowledge engineering to data mining. It also employs techniques like logical deduction and knowledge engineering and integrates knowledge engineering with the specificities of GIS like spatial reference systems and spatial reasoning induction [29].

It also extends methods derived from cognitive science, such as semantic similarity and analogy reasoning, for example, to allow the retrieval of semantic geographic information [30]. Often, geospatial semantics combines work on conceptual modeling and geo-ontologies with spatial statistics, e.g. for the study of land cover [31].

The semantic interpretations of geographic information may vary significantly, which often causes misinterpretations when using and combining data and services on the Web. An example is web services that provide sensor data, e.g. from meteorological stations. In order to simulate the spreading of a toxic gas, two different services can be sought for the wind direction measurement. Both services can be syntactically comparable as returning a string of wind direction as output, along with an integer ranging from 0 to 360 °. However, both services may have contradictory semantic interpretations from what the return values indicate: wind blows to or from. Thus, by sending both values to an evacuation simulation conducted by a Web Processing Service (WPS) will produce misleading results [32]. In addition to the challenges arising from the integration of heterogeneous data and the combination of services, the intercomparison of models of data plays another crucial role [33] . Finally, time and therefore the change effect is another challenge to be considered. Most concepts are not static but evolve over time or are dynamically redefined. For long-term preservation and maintenance of data and ontologies this leads to research challenges such as how to handle semantic aging [34].

By analogy with Kuhn's distinction [35] between modeling versus encoding on the Semantic Web, We can distinguish between two distinct scientific thinking methodologies in geospatial semantics. The design task of semantic modeling addresses the problem of how geographic information should be modeled in an ontology, for example what relationships and classes are useful in order to discover, record and explore the essence of spatiotemporal and geographic phenomena. Examples include work on geotechnical ontology [36] [37] and standardization of spatial logic [38]. Spatial relations facilitate the querying and localization of complex geometrical objects such as cities or buildings, with respect to other references, such as countries and roads [39]. This research strand involves spatial replications and functions in Geographic Information Systems (GIS) [40] as well as integrity constraints in spatial databases.

Another strand concerns querying based on semantics, integration and interoperability of geographic information. We must keep in mind that geodata and its respective models are characterized by immense heterogeneity as they expand over the areas not only of geography, cli

atology, geology, ecology and oceanography but also of economics, transportation research and so on. Therefore the integration and sharing of the resulting georeferenced information requires methods to ensure semantic interoperability [41]. Furthermore, depending on the circumstances we need different levels of abstraction, detail or scaling in the representation of the information, which may be inherently unclear and uncertain [42], thus leading to another cause of interoperability problems. The automatic discovery of geospatial objects of interest in non-georeferenced data sources followed by their semantic connection and linking is still a very difficult issue. Other work following similar guidelines explore the role of semantic similarity for spatial queries [43].

6.4. Semantic interoperability

Standard semantic descriptions of geo-services promised to automate service discovery. Describing the semantics of geo-field services inputs and outputs is critical to the geo-service discovery. The term "semantics" here refers to the concept of expression in a language [28]. Expressions can be single symbols (the words of a language) or symbol combinations. The conceptual or meaning triangle defines the interaction between symbols or words, concepts and objects of the world (Figure 20).

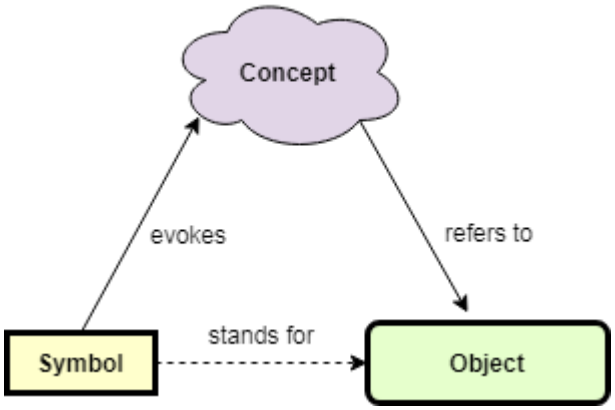


Figure 20 - The Meaning Triangle

The triangle shows that the relationship between a word and a thing is indirect, and words cannot fully grasp the real meaning of a thing. The right connection is achieved only when a factor interprets the word by referring to a corresponding concept in a context by choosing the desired interpretation and rejecting others (we use the term factor to emphasize that discovery services are used by either humans or software on their behalf).

Conceptualization is the description of a reality that is considered and organized by a factor,

regardless of the vocabulary used and the actual appearance of a particular situation [44]. We meet several definitions of ontology, "an ontology is a specification of an idea" [45], or “An ontology is a specific artificial object designed to express the intended meaning of a vocabulary in terms of the nature and structure of the entities to which it refers” [44]. An ontology usually contains two different parts: names for important concepts and knowledge / constraints in the field [46].

Ontologies can be categorized according to the level of detail and their degree of dependence on a particular task or view [47]. The level of detail can be classified by the ontological precision, from catalog to axiomatized theory [44]. The dependence on a particular task or point of view distinguishes between top-level, domain, task, and application ontologies. In order to reconcile the ontologies of requested and provided geo-services at the application level, there must be an agreement between GIS and environmental models on both fundamental basic and general concepts.

6.5. Layer-Based Structure of Ontologies

The upper ontology, the ontology of measurement theory, the core ontology of geo-services and the Descriptions and Situations (D&S) ontology [42] can be structured in four layers (Figure 21).

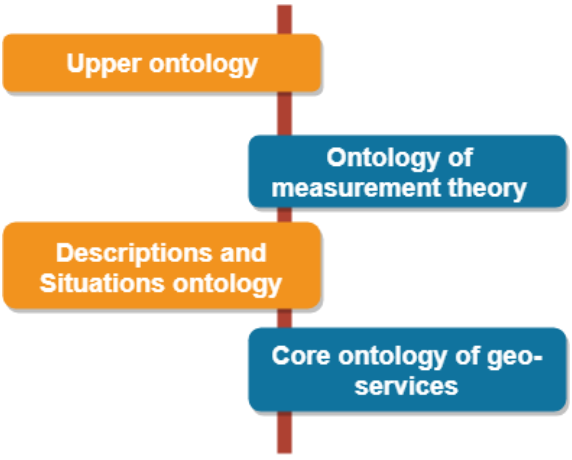


Figure 21 – Ontological Structures

The role of the Descriptions and Situations (D & S) ontology is to fill the conceptual gap between upper ontology and ontology of measurement theory on one hand, and core ontology on the other. To uniquely identify concepts and relationships in these ontologies, the labels "uont",

"mth", "das" and "cogs" are used respectively for upper ontology, ontology of measurement theory, ontology of descriptions and situations and core ontology of geo-services. The following subsections explain the concepts and relationships in the ontology of measurement theory.

6.5.1. The Ontology of Measurement Theory

Each entity includes some properties that exist as long as the entity exists [48]. In terms of conceptual beliefs, these attributes are a set of states for modeling the physical system that can be observed in every direction. Geospatial data can be used to capture and display features such as temperature, population density or soil type, which can subsequently serve as input or output for geo-services. The features of a field, including the type of measurement and the unit of measurement, are an important part of the description of input and output semantics in a geo-service.

6.5.2. Scales of Measurement and Measurement Units

The results of observations are recorded as magnitudes on a specific measurement scale. The attributes of data are typically classified into four scales: nominal, ordinal, interval and ratio [49]. For example, features such as flow rate, wind speed, penetration rate, and physical distance are expressed in a ratio scale. Features such as temperature, latitude, longitude and daytime are expressed on interval scale. These measurement scales differ in terms of what numerical operators can be applied. For example, it is possible to divide, subtract or sum up two values with ratio scales, while it is only possible to sum up or subtract two values with spacing scales. Characteristics measured in ratio or interval scales are categorized as quantitative characteristics.

Features such as drainage class or erosion potential are usually on an ordinal scale, often

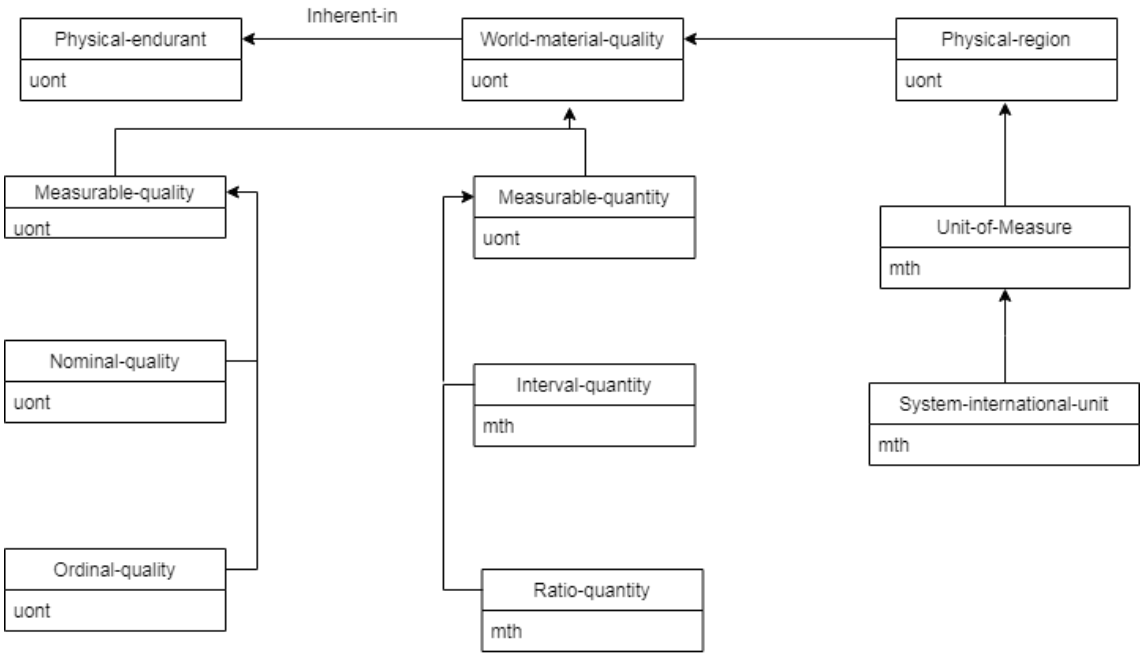


Figure 22 - Ontology of measurement theory. The alignments depict the relationship between the concepts, while the arrows depict the subsumption relationships (superclass / subclasses)

coded with numbers (e.g. 1 = good, 2 = moderate, 3 = bad). Other properties such as land cover, soil type, soil texture, and rock type are on a nominal scale (e.g. 1 = rocky, 2 = clay). Ordinal and nominal values cannot be used in numerical expressions and are therefore classified as qualitative (ordinal quality and nominal quality (Figure 22)).

The unit of measurement is a feature used to describe the semantics of a field's properties. Magnitudes of quantitative characteristics, can be compared with units of measurement (e.g. kg/m2). The units of measurement are described in the ontology of a measurement theory. In that sense, the concept of unit of measurement is the formal description of the unit of measurement (Figure 22). KilogramPerSquareMeter is an individual of the unit of measurement concept, used as units of measurement for quantitative characteristics.

6.5.3. Core Ontology of Geo - Services

An ontology containing geo-service concepts depicts the properties and capabilities of geo-services. The Web-Ontology Working Group at the World Wide Web Consortium has created an ontology of service concepts that a corresponding service manufacturer is supplied with a core set of markup language structures to describe the properties and capabilities of a Web service [50]. However, OWL-S seems to lack a formal semantic framework. Some of the missing

semantics are given informally in the text of the document [51]. One particular limitation is that for each Service, only one ServiceModel is expected to hold. This makes it impossible to evaluate the relationship between a ServiceModel that is required by an applicant and the subject of the provider system [51].

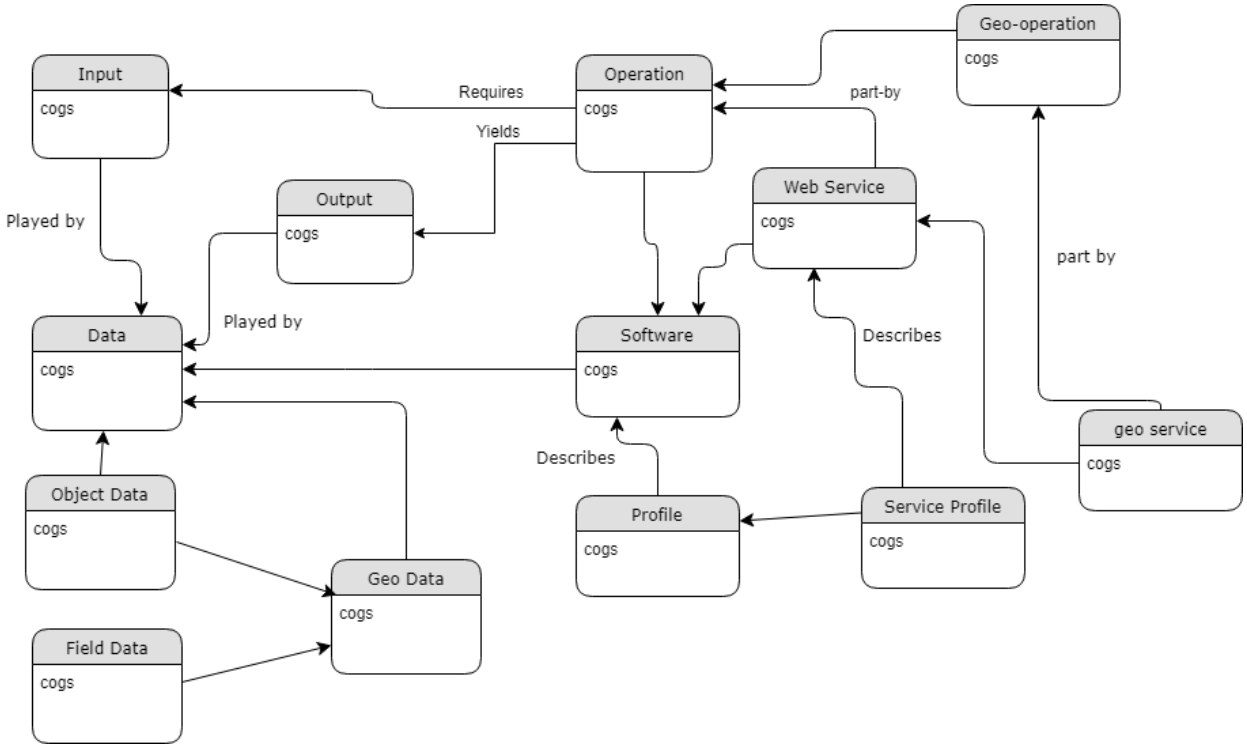


Figure 23 - The concepts and relationships for describing geo-services

To overcome these constraints, the basic geo-service ontology must include concepts such as geo-service, geometry, and service profile (Figure 23). The similarity between a geo-service requested and the geo-services provided can be determined by achieving the degree of similarity between these concepts [35].

6.6. Chapter Conclusions

The sixth Chapter deals with the topics of semantic modeling and ontologies. Special mention was made to geospatial semantics, the need for semantic interoperability and the layer-based structure of ontologies.

7. Machine Learning

7.1. Introduction

Machine learning abides as one of the fields of artificial intelligence. The scientific field that seeks to simulate the conceptual function of humans to gain knowledge of the environment using a machine is machine learning. For a human or an animal, the learning process is innate. The objective of machine learning is to accomplish making a machine capable of efficiently receiving information from the environment and to subsequently proceed in understanding of the data in some way. The ultimate goal is to develop machines capable of making decisions or predicting cases based on data input. This can be accomplished by employing Machine Learning Classifiers. In general classification is the process of predicting the class of given data. Classification predictive modeling is the task of approximating a mapping function from input variables to discrete output variables. A classifier utilizes some training data to understand how given input variables relate to each class. The most complex and most efficient in terms of yielding the best results Machine Learning Classifiers applied and evaluated in the experiment of the current thesis are: Support Vector Machines and Random Forests and they will be presented briefly.

7.2. Support Vector Machines

The Support Vector Machines [52] constitute the implementation of a machine learning methodology based on Statistical Learning Theory [53]. Support vector machines share similar architecture with neural networks but also bear many radical differences. They possess dissimilar construction methods. The evolution of neural networks proceeded heuristically, with applications and extensive experimentation preceding theory. In contrast, for the development of SVMs, a robust theory was first founded and developed and then followed implementations and experiments.

Support Vector Machines can be used for classification or regression. When used for classification, they aspire to find the optimal boundary hyperplane (in the case of two-dimensional data the hyperplane is reduced to a line) that separates the classes. The Support Vector Machines construct a hyperplane or a set of hyperplanes in a high or infinite dimension space and aim to select the optimal hyperplane that best separates the points in the input variable space by their class.

The hyperplane that prevails is the one with equal and maximum distance from the closest

representatives of each class, namely the *support vectors*. The distance between the hyperplane and these support vectors creates what is referred as a margin (Figure 24). To calculate the margin the perpendicular distance from the hyperplane to the closest data points is used.

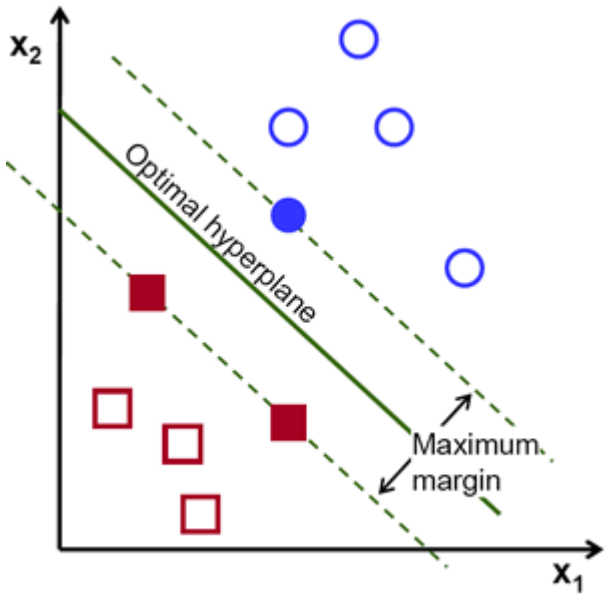


Figure 24 - Optimal Hyperplane in two dimensions

Only these points contribute to defining the hyperplane and in the construction of the classifier. These points are the support vectors and they support or define the hyperplane. The Maximal-Margin hyperplane is the best or optimal hyperplane that can separate the classes while preserving the largest margin. To attain that hyperplane, an optimization procedure that maximizes the margin is used that learns from training examples and can be expressed through the following expression:

$$\min_{w,b_0} L(w,b_0) = \frac{1}{2}\|w\|^2 \quad \text{subject to} \quad t_i(w^T x + b_0) \geq 1 \; \forall i \qquad (1)$$

Where $L(w,b_0)$ is the function we want to minimize with some constraints and t_i is the label of each class. For example in the most commonly used case of two classes $t_i \in [-1,1]$.

To provide a brief mathematic background on hyperplanes, the equation of a line is:

$$y = ax + b$$

Which is equivalent to:

$$y - ax - b = 0$$

While the equation of a hyperplane is defined by:

$$\boldsymbol{w}^T\boldsymbol{x} = 0$$

Which consists of the inner product of 2 vectors.

Given two vectors $w\begin{pmatrix}-b\\-a\\1\end{pmatrix}$ and $x\begin{pmatrix}1\\x\\y\end{pmatrix}$, we get

$$\boldsymbol{w}^T\boldsymbol{x} = -b \times (1) + (-a) \times x + 1 \times y$$
$$\boldsymbol{w}^T\boldsymbol{x} = y - ax - b$$

Or using another (dot) notation for the inner product

$$\boldsymbol{w} \cdot \boldsymbol{x} = y - ax - b \qquad (2)$$

Which leads us to the fact that there are many ways to express the same relation and gives us the notation for the hyperplane with which we will continue for the following reasons: This notation makes it easier to work in more than two dimensions and also the vector **w** is always normal to the hyperplane by definition.

However in the example above we used vectors that have 3 dimensions, but if for simplicity use two dimensional vectors $\boldsymbol{w}'(-a, 1)$ and $\boldsymbol{x}'(x, y)$ we get

$$\boldsymbol{w}' \cdot \boldsymbol{x}' = (-a) \times x + 1 \times y$$
$$\boldsymbol{w}' \cdot \boldsymbol{x}' = y - ax \qquad (3)$$

Combining (2) and (3) and adding *b* on both sides we get

$$\boldsymbol{w}' \cdot \boldsymbol{x}' + b = \boldsymbol{w} \cdot \boldsymbol{x} \qquad (4)$$

So we are looking for our classifier hyperplane $\boldsymbol{H_0}$ (linear separator in the case of two dimensions) separating our dataset and also satisfying

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

Where $\boldsymbol{w}$ is a weight vector, $\boldsymbol{x}$ is the input vector and *b* is bias. In essence we aim to determine 2 other hyperplanes $\boldsymbol{H_1}$ and $\boldsymbol{H_2}$ (equidistant from $\boldsymbol{H_0}$) having the following equations respectively $\boldsymbol{w} \cdot \boldsymbol{x} + b = \delta$ and $\cdot \boldsymbol{x} + b = -\delta$. We can assume without generality loss that δ =1. Basically that means that we are looking for the values of **w** and *b* So we want to select the hyperplanes that

have no data points between them thus they have to comply with the following constraints:

$$\boldsymbol{w \cdot x_i + b \geq 1 \; when \; y_i = 1} \; (\boldsymbol{x_i \; belongs \; to \; class \; 1}) \qquad (5)$$

$$\boldsymbol{w \cdot x_i + b \leq -1 \; when \; y_i = -1} \; (\boldsymbol{x_i \; belongs \; to \; class \; -1}) \qquad (6)$$

Multiplying both sides with $\boldsymbol{y_i}$ on (5) and (6) and replacing it on the right side with 1 and -1 respectively, we can combine the equations both equations to

$$y_i \; (\boldsymbol{w \cdot x_i} + b) \geq 1 \qquad \forall \; i \; \epsilon \; [1, n] \qquad (7)$$

Giving us a constraint. For the next step we have to try and maximize the margin (distance between the 2 hyperplanes with respect to the constraints). To calculate the distance we take a point in the first hyperplane and with the help of w which is perpendicular to the first hyperplane we try to find its projection to the second hyperplane which is:

$$m = \frac{2}{\|\boldsymbol{w}\|}$$

Summarizing we have to solve

$$\text{Minimize in } (\boldsymbol{w}, b)$$
$$\|\boldsymbol{w}\|$$
$$subject \; to \; y_i(\boldsymbol{w \cdot x_i} + b) \geq 1 \; \forall \; i \; \epsilon \; [1, n]$$

which is an optimization problem with actually *n* constraints and can be reformulated as follows:

$$\underset{w,b}{\text{minimize}} \; \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$subject \; to \; y_i(\boldsymbol{w \cdot x_i}) + b - 1 \geq 0 \; \forall \; i \; \epsilon \; [1, n]$$

The factor $\frac{1}{2}$ we added and squaring the norm helps us overpass some critical difficulties because it removes the square root and transforms the problem to convex quadratic optimization. To ascertain the fact that we calculate the global minimum and not some local, we will use some properties from convex functions and specifically the theorem that states that a minimum of a convex function is always global. Still we have to calculate the minimum of our multivariate function with constraints. We have to prove that the hessian matrix is semi definite and in order to achieve that we have to find the partial derivatives of each variable. Furthermore we will use Lagrange multipliers. The method of Lagrange multiplier solves optimization problems with

equality constraints which is not the case here but can still be used in our case provided it complies with some additional conditions (KKT conditions). To elaborate:

We declare a function

$$f(\boldsymbol{w}) = \frac{1}{2}\|\boldsymbol{w}\|^2$$

and *n* constraint functions:

$$g_i(\boldsymbol{w}, b) = y_i(\boldsymbol{w} \cdot \boldsymbol{x_i}) + b - 1, \qquad \forall\, i \,\epsilon\, [1, n]$$

Lagrange perceived that the minimum of a function *f* under the constraint of *g* is found when their gradient points are in the same direction ($\nabla f(x) = a\nabla g(x)$). This leads us to introduce the Lagrangian function:

$$L(\boldsymbol{w}, b, a) = f(\boldsymbol{w}) - \sum_{i=1}^{n} a_i g_i(\boldsymbol{w}, b)$$

$$L(\boldsymbol{w}, b, a) = \frac{1}{2}\,\|\boldsymbol{w}\|^2 - \sum_{i=1}^{n} a_i(y_i(\boldsymbol{w} \cdot \boldsymbol{x_i}) + b - 1) \qquad (8)$$

The reason for rewriting *f* now becomes clearer, as for convex functions we know that strong duality holds (Slater's theorem) and this minimization problem can be considered as a maximization problem. Also notice we introduce a Lagrange multiplier a_i for each constraint function. As we mentioned above, part of the solution comprises calculating the partial derivatives of *L* with respect to **w** and *b*. solving for the gradient of the Lagrangian ($\nabla L(\boldsymbol{w}, b, a) = 0$), results in finding the points where the gradient of *f* and *g* are parallels (hence the minimum).

$$\nabla_{\boldsymbol{w}} L = \boldsymbol{w} - \sum_{i=1}^{n} a_i y_i \boldsymbol{x}_i = 0$$

$$\boldsymbol{w} = \sum_{i=1}^{n} a_i y_i \boldsymbol{x}_i$$

and

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} a_i y_i = 0$$

Substituting **w** to our original Lagrangian (8) we define a new function

$$W(a,b) = \frac{1}{2}\left(\sum_{i=1}^{n} a_i y_i \boldsymbol{x}_i\right) \cdot \left(\sum_{j=1}^{n} a_j y_j \boldsymbol{x}_j\right) - \sum_{i=1}^{n} a_i \left[y_i \left((\sum_{j=1}^{n} a_j y_j \boldsymbol{x}_j) \cdot \boldsymbol{x}_i + b \right) - 1 \right]$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \, \boldsymbol{x}_i \cdot \boldsymbol{x}_j - \sum_{i=1}^{n} a_i \, y_i \left((\sum_{j=1}^{n} a_j y_j \boldsymbol{x}_j) \cdot \boldsymbol{x}_i + b \right) + \sum_{i=1}^{n} a_i$$

$$= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \, \boldsymbol{x}_i \cdot \boldsymbol{x}_j - \sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \, \boldsymbol{x}_i \cdot \boldsymbol{x}_j - b\sum_{i=1}^{n} a_i y_i + \sum_{i=1}^{n} a_i$$

$$\sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \, \boldsymbol{x}_i \cdot \boldsymbol{x}_j - b\sum_{i=1}^{n} a_i y_i$$

However we know that

$$\sum_{i=1}^{n} a_i y_i = 0$$

So we get the Wolfe Lagrangian function

$$W(a) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} a_i a_j y_i y_j \, \boldsymbol{x}_i \cdot \boldsymbol{x}_j \qquad (9)$$

$$subject\ to\ \ a_i \geq 0 \ \forall\ i \in [1, n]$$

From our original problem we eliminated the dependencies to **w** and *b*. Due to the inequality constraints our solution must satisfy the Karush-Kuhn-Tucker (KKT) conditions (Stationarity, primal feasibility, dual feasibility and complementary slackness). So now we can solve our last equation because the only unknowns are the multipliers and subsequently calculate **w** and b using relations

$$\boldsymbol{w} = \sum_{i=1}^{n} a_i y_i \boldsymbol{x}_i$$

And

$$b = y_i - \boldsymbol{w} \cdot \boldsymbol{x_i}$$

If we examine more closely (9), we can deduce some useful insight. If the input vectors $\boldsymbol{x}_i, \boldsymbol{x}_j$ are dissimilar ore completely alike then their dot product is 0 and they do not contribute to *W*. If the two vectors make similar predictions then the value of $a_i a_j y_i y_j \boldsymbol{x}_i \cdot \boldsymbol{x}_j$ will be positive resulting in decreasing the value of W (since it is subtracted from the first term, so the contribution of similar feature vectors is decreased. On the contrary when vectors with similar feature make opposite predictions the sum is maximized, because these are the critical ones, the one that define or support the hyperplane.

However, most realistic problems have an advanced degree of complexity and their training examples are fuzzy and not linearly separable in the feature space they are defined, thus leading to inability of locating an appropriate separating hyperplane. To provide countermeasures for potential noise in the data, we can introduce a slack variable that allow the classifier to make some mistakes.

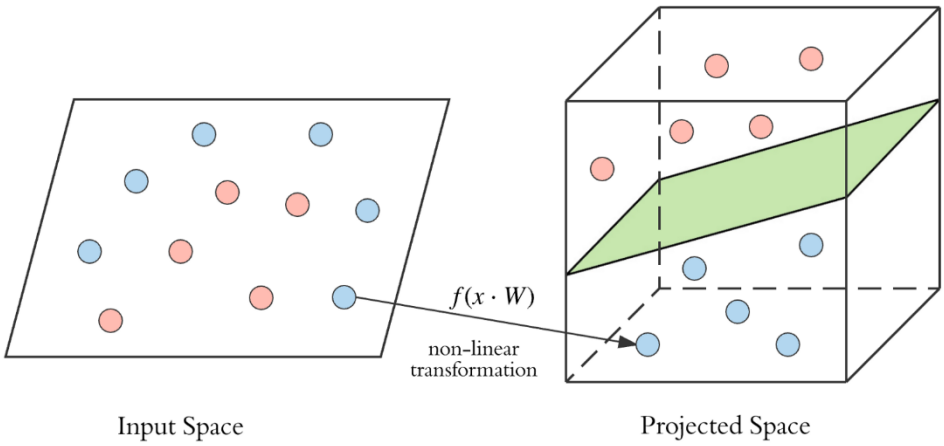


Figure 25 - The kernel trick

To overcome the problem of linear inseparability, a technique known as the kernel trick is used (Figure 25). The main idea behind this is mapping the data to a higher dimension to regain linear separability. The kernel trick exploits the observation that in (9) only the dot product of input vectors is used. If we transform using a transformation function φ, the instead of calculating $(\varphi(x_i) \cdot \varphi(x_j))$, which is computationally expensive and time consuming and involves defining φ explicitly, we can define a kernel function K such that $K(x_i, x_j) = (\varphi(x_i) \cdot \varphi(x_j))$, then we do not have to compute φ but only the inner products in the transformed space.

For example if we consider two points in two dimensional space $p = (p_1,p_2)$ and $q = (q_1, q_2)$ and a φ a mapping function which maps two dimensional point to three dimensional space as follows

$$\varphi(x) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$$

Then let *K(p,q)* a kernel function which performs a dot product between two points as

$$K(p,q) = \varphi(p) \cdot \varphi(q) = (p_1^2, p_2^2, \sqrt{2}p_1p_2) \cdot (q_1^2, q_2^2, \sqrt{2}q_1q_2)$$
$$= p_1q_1 + p_2q_2 + 2p_1q_1p_2q_2$$
$$= (p_1q_1 + p_2q_2)^2 = (p \cdot q)^2 = \langle p, q\rangle^2 = \langle \varphi(p), \varphi(q)\rangle$$

The above shows how a dot product in three dimensional space can be achieved through a dot product on two dimensional space. For nonlinear transformation we can also use apart from the polynomial kernels described above ($K(x,y) = (x \cdot y + 1)^p$ in general form), radial basis function (10) or sigmoid. The latter will be thoroughly tested in our experiments through their hyperparameters as will be explained in the representative chapter.

$$RBF\ (Gaussian)\ kernel\ K(x_i, x_j) = e^{-\frac{||x_i - x_j||^2}{2\sigma^2}} \qquad (10)$$

To summarize, we saw that the SVMs present significant advantages over Neural Networks (NN). They address the drawback frequently observed in NN, that of suffering from multiple local minima, yielding a solution global and unique. Moreover SVMs have a simple geometric interpretation and their computational complexity does not depend on the dimensionality of the input space. ANNs avail empirical risk minimization, while SVMs on the other hand use structural risk minimization. Finally SVMs often outperform ANNs in practice because they are less prone to overfitting, which is one of the biggest problem with ANNs.

7.3. Random Forests

The random forest classifier belongs to the broader field of ensemble learning (methods that provide and generate a number of classifiers and aggregate their results). The two best known methods are boosting and bagging of classification trees. The key point of the boosting technique is that consecutive trees add extra weight to the points incorrectly predicted by previous predictors. Upon completion a weighted vote is used for prediction. On the contrary, for bagging (bootstrap aggregating) subsequent trees do not depend on previous trees and each one is

independently constructed using a bootstrap sample of the data set. Finally for the prediction a simple majority vote is utilized.

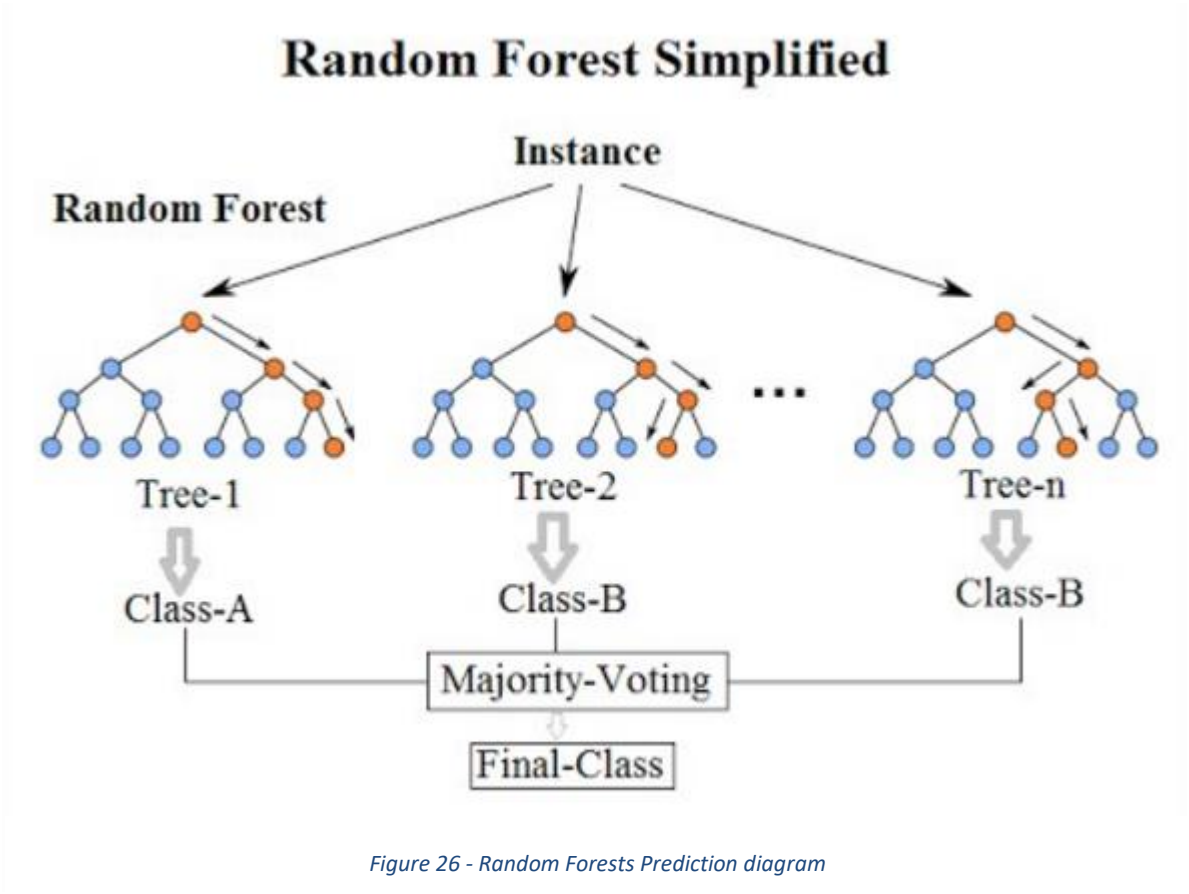


Figure 26 - Random Forests Prediction diagram

Random Forests provide an extra mantle of randomness to bagging. Apart from the creation process (each tree uses a different bootstrap sample of the data), random forests use different tree construction method. In a standard tree, each node is separated using the best separation score considering all predictor variables. In a random forest, each node is divided using the best score between a subset of the predictors randomly selected on this node. This strategy may initially look odd but it is proven to perform very well compared to many other classifiers, including Support Vector Machines and neural networks. Furthermore, due to that strategy they are resistant against overfitting [54]. Random forest algorithm consists of 3 steps (Figure 26):

1. Drawing n bootstrap samples from the original dataset (n refers to the numbers of trees to be constructed).

2. For each of the bootstrap samples, grow a classification tree after modifying it as follows: in each node, rather than choosing the best segregation between all predictors, randomly sample m predictors and select the best separation between them

3. The predictions for new data can be implemented by aggregating the predictions of the n trees (majority vote).

The appropriate number of trees to be constructed, as well as the number of predictors to be sampled, has also been subjected for studying by many experts, but most researchers including the creator [54] suggest empirical tests according to the dataset characteristics.

We can acquire an estimation of the error rate, based on training data, using the following procedure: Firstly, for each iteration using the bootstrap data, predict the data not in the bootstrap sample (Breiman refers to them as "out-of-bag", or OOB, data) using the tree grown with the original sample. For the second part, we aggregate the OOB predictions. Depending on the implementation of the algorithm, the percentage of times that some data will be out of the bag differs, so we have to aggregate them and then calculate the error rate. This is an indicative way of calculating the error rate called OOB error and generally provides appreciable and accurate estimation of error rate [55]. In the experiments we conducted, we did not use only this error estimation method, but also other metrics.

7.4. Chapter Conclusions

The seventh Chapter concerned Machine Learning and presented the mathematical background and theory behind Support Vector Machines and their geometric interpretation. Furthermore the Random Forests classifier was also discussed.

8. Multi Criteria Decision Analysis (MCDA)

8.1. Preference modeling

MCDA techniques involve ways of mapping and modeling preferences. They attempt to capture the informal, subjective and often non-measurable notion of user (or decision maker) preference, and accordingly for meaning of comparison as perceived by humans. The concept of preference refers to a set of alternative actions, options or candidates (Figure 27).

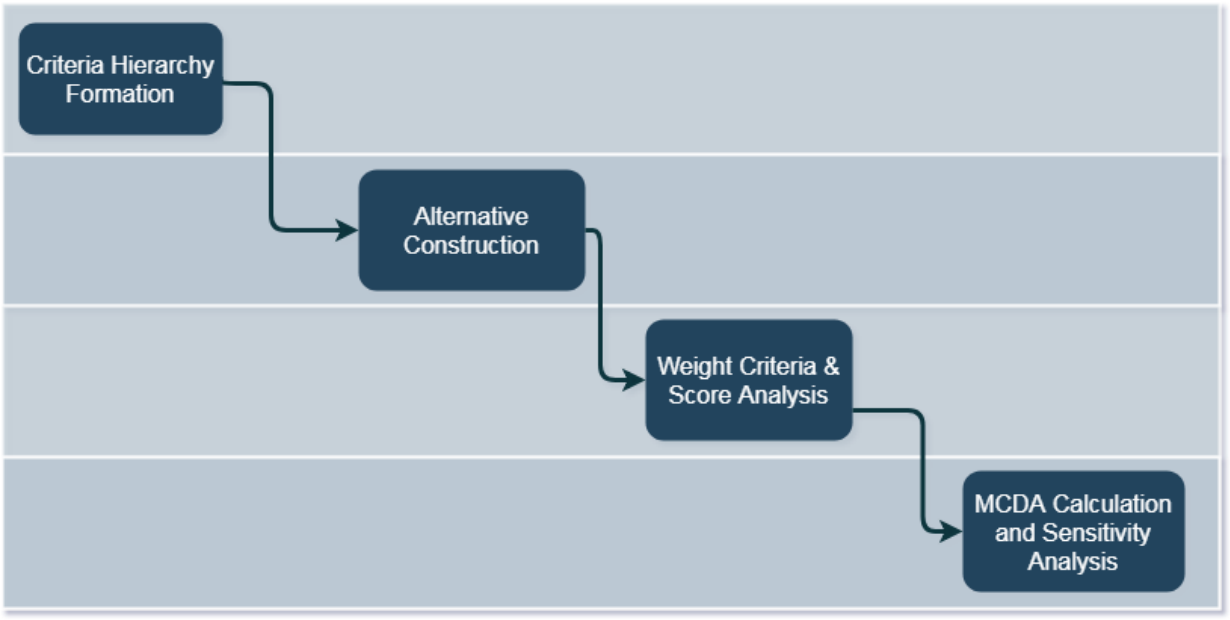


Figure 27 - MCDA typical flowchart

The aforementioned set must comply with certain formalistic rules [56], therefore it is considered finite, in the sense that it does not contain infinite alternatives and assumed to be stabled and not diversify during decision making. Between two options we may:

- Prefer one against the other
- Be indifferent to either selected
- Be unable to choose

So if we consider a set of alternative options A

$$A = \{a_i : i = 1, \dots, n\}$$

We can define the following relations :

- Preference P denoted as : $a_i \, P \, a_k$

$$P = \{\{a_i, a_k\} : a_i, a_k \in A\}$$

- Indifference I : $a_i \, I \, a_k$

$$I = \{\{a_i, a_k\} : a_i, a_k \in A\}$$

- Incomparability R : $a_i \, R \, a_k$

$$R = \{\{a_i, a_k\} : a_i, a_k \in A\}$$

We also can define a superiority relation based on *P* and *I*:

$$a_i \, S \, a_k \iff a_i \, P \, a_k \vee \; a_i \, I \, a_k$$

We must also clarify the difference between indifference and incomparability, the first represents an equal preference of the DM for each action, whilst the latter represents absence of preference expression. Between two actions only one of the three relations may apply, so

$$P \cap I = \emptyset \qquad (11)$$

$$I \cap R = \emptyset, \qquad (12)$$

$$P \cap R = \emptyset \qquad (13)$$

The prerequisites for an valid preference structure of classic preference modelling is that the indifference relation must be reflexive ($\forall \, a_i \in A : a_i \, I \, a_i$) and symmetric ($\forall \, a_i, \, a_k \in A : a_i \, I \, a_k \Rightarrow a_k \, I \, a_i$), while the preference relation is asymmetric ($\forall \, a_i, \, a_k \in A : a_i \, P \, a_k \Rightarrow a_k \not{P} \, a_i$) and finally the incomparability relation must also be symmetric ($\forall \, a_i, \, a_k \in A : a_i \, R \, a_k \Rightarrow a_k \, R \, a_i$).

Preference can be modeled as a function that depicts each action as a number, $f_p : A \rightarrow \mathbb{R}$. Therefore it is evident that $a_i \, I \, a_k \Rightarrow f_p(a_i) = f_p(a_k)$ and $a_i \, P \, a_k \Rightarrow f_p(a_i) > f_p(a_k)$. As a consequence there are no non-comparable actions, since they can be potentially modeled and depicted as numbers, which also gives us

$$\forall \, a_i, a_k \in A :$$
$$(a_i \, P \, a_k) \wedge \neg \, (a_k \, P \, a_i) \; \wedge \neg \, (a_i \, I \, a_k)$$
$$\vee$$
$$\neg (a_i \, P \, a_k) \wedge \; (a_k \, P \, a_i) \; \wedge \neg \, (a_i \, I \, a_k)$$
$$\vee$$

$$\neg(a_i \; P \; a_k) \wedge \neg \; (a_k \; P \; a_i) \; \wedge \; (a_i \; I \; a_k)$$

8.2. SMART (Simple Multi-Attribute Rating Technique)

SMART is a multi-criteria decision method, which can be characterized as compensatory [57] and is established on the principle that each alternative consists of criteria having specific values and each criterion has weights that can describe its significance in relation to the other criteria. Based on this rating, an assessment is made to every alternative in order to find the optimal solution and to that end, a linear additive model is adopted to predict the value of each option. The particular method's compelling advantage is the simplicity of the questions and answers upon which the Decision Maker interacts, which influence directly and facilitate the understanding of the Decision Maker. The analysis and steps of the process are transparent which in its turn provides a better understanding of the problem and is admissible by the Decision Maker. Its key steps include the identification of the alternatives to be evaluated and then the identification of the relevant dimensions of value for evaluation of the alternatives. If the goals are too many they have to be combined or the least important omitted. Subsequently, the dimensions are ranked and rated in order of importance by the DM (hence the need for limiting their number) and the weights used for the latter normalized. Finally the location of each alternative evaluated on each dimension is measured and utility values for each alternative is calculated. That multi linear function model can be expressed as:

$$U_j = \sum\nolimits_k \cdot w_k u_{jk}$$

where U_j stands for the utility value of alternative j, w_k is the normalized weight for objective k, and u_{jk} is the scaled value for alternative j on dimension k. It is evident that the decision selection aspires for the alternative with maximum value (U_j). The function can be used also for ranking alternatives.

8.3. ELECTRE (Elimination Et Choix Traduisant la REalité)

The method belongs to the outranking methods and is based to pairwise comparisons of options which implies that every alternative is compared to all others. In combination with the use of thresholds they are the two fundamental characteristics of the method. Let $g_j, j = 1,2, \dots n$ a set of defined criteria and A set of alternatives, the traditional preference modeling assumes that for two alternatives $(a, b) \; \epsilon \; A$:

$$aPb\ (a\ is\ preferred\ to\ b) \Leftrightarrow g(a) > g(b)$$
$$aIb\ (a\ is\ indifferent\ to\ b) \Leftrightarrow g(a) > g(b)$$

the above two relations hold.

Conversely ELECTRE methods and specifically ELECTRE III include the concept of an indifference threshold, *q*, reforming the preference relations as follows:

$$aPb\ (a\ is\ preferred\ to\ b) \Leftrightarrow g(a) > g(b) + q$$
$$aIb\ (a\ is\ indifferent\ to\ b) \Leftrightarrow |g(a) - g(b)| \leq q$$

The introduction of the threshold expresses the confidence of the DM for the preference in an alternative during a realistic pairwise comparison. However, we must take into account the fact that there are cases or instances where the preference or the superiority of an alternative or a specific criterion is not clear, and the point where the DM differentiates from indifference to preference is unclear. Therefore, there are foreceful reasons for introducing an intermediate buffer zone among indifference and strict preference to illustrate this difficulty of the DM for an apparent choice. This zone is referred as a weak preference, and can be also modeled as a binary relation introducing a new "preference" threshold, *p*. That gives us a double threshold model and a binary relation *Q* to measure the weak preference:

$$aPb\ (a\ is\ strongly\ preferred\ to\ b) \Leftrightarrow g(a) - g(b) > p$$
$$aQb\ (a\ is\ weakly\ preferred\ to\ b) \Leftrightarrow q < g(a) - g(b) \leq p$$
$$aIb\ (a\ is\ indifferent\ to\ b) \Leftrightarrow |g(a) - g(b)| \leq q$$

The thresholds obviously strongly influence the binary relations, making the selection of appropriate ones complicated. However in the more realistic decision-making situations there are good reasons for selecting non-zero values for p and q [58].

In this process we must contemplate the probability that p and q may not necessarily be constant, but derived from a function and therefore become variable. Even though the use of constant thresholds obviously leads to a simplification of the process and the complexity of the method, if a criterion has high values and therefore may create larger indifference and preference thresholds, it could be reasonable to explore the use of variable thresholds.

The method exploits the thresholds to try and establish an outranking relation *S*. the binary relation *aSb* means that according to the DM preferences "a is as good as b" or "a is not worse

than b". All possible pairs of a and b are then examined to check the validity of the aforementioned assessment (*aSb).* That leads us to the following possible combinations:

$$aSb \wedge \neg (bSa)$$
$$\neg (aSb) \wedge bSa$$
$$aSb \wedge bSa$$
$$\neg (aSb) \wedge \neg (bSa)$$

The third assertion stands for indifference and the fourth for incomparability. In order to decide whether to accept the relation *aSb*, two principles are taken under consideration:

1. The majority principle, which considers whether the majority of the criteria, according to the relative weight of each one, supports this relationship thus creating the concordance matrix and
2. The non-discordance principle. It checks whether among the minority of the criteria voting against the assertion, exists strong oppositions.

To provide a better insight in the implementation of the above principles, we should consider the outranking relations for each of the *n* criteria:

$aS_j b$ or "a is as good as b with respect to the j^{th} criterion ", *j=1,2,…n.*

The j^{th} criterion is concordant with *aSb* iff(if and only if) $aS_j b$ which entails that $g_j(a) \geq g_j(b) - q_j$. That is because even if $g_j(a)$ has a smaller value than $g_j(b)$, if their difference is smaller than q_j, then it does not contradict the assertion.

The j^{th} criterion is discordant with *aSb* iff $bP_j a$ or $g_j(b) \geq g_j(a) + p_j$. That is because we have a strong preference for b over a for criterion j and therefore the criterion is not in concordance with the assertion.

Based on the above we can measure the strength of the assertion *aSb.* We must first establish a measure of concordance using the concordance index C (a,b) for every pair of alternatives $(a, b) \in A$. Let k_j represent the weight for criterion j. Sometimes the weights are referenced as importance coefficients [59]. We can define the following relation:

$$C(a,b) = \frac{1}{k}\sum_{j=1}^{n} k_j c_j\,(a,b), \quad where\ k = \sum_{j=1}^{n} k_j$$

And

$$c_j(a,b) = \begin{cases} 1, & \text{if } g_j(a) + q_j \geq g_j(b) \\ 0, & \text{if } g_j(a) + p_j \leq g_j(b) \\ \dfrac{g_j(a) + p_j - g_j(b)}{p_j - q_j}, & otherwise \end{cases}, \qquad j = 1,2,\dots n$$

For the discordance principle we must calculate the veto threshold, v_j, to imply the possibility of refusing the assertion if $g_j(b) > g_j(a) + v_j$ for any criterion j. The discordance criterion $d_j(a,b)$ is given by:

$$d_j(a,b) = \begin{cases} 0, & \text{if } g_j(a) + p_j \geq g_j(b) \\ 1, & \text{if } g_j(a) + v_j \leq g_j(b) \\ \dfrac{g_j(b) - g_j(a) - p_j}{v_j - p_j}, & otherwise \end{cases}, \qquad j = 1,2,\dots n$$

We now have defined both concordance and discordance for each pair of alternatives $(a,b) \in A$. We can combine them to produce a credibility degree for the assertion under discussion and define for every pair of $(a,b) \in A$ the following formula:

$$S(a,b) = \begin{cases} C(a,b), & if\ d_j(a,b) \leq C(a,b)\ \forall j \\ C(a,b) \cdot \prod_{j \in J(a,b)} \dfrac{1 - d_j(a,b)}{1 - C(a,b)} & where\ J(a,b)\ is\ the\ set\ of\ criteria \\ & such\ that\ d_j(a,b) > C(a,b) \end{cases}$$

If the measure of concordance transcends the one of discordance then no modification needs to happen, whereas if the opposite is the case then we have to reconsider the assertion and adjust the value according to formula. It is worth mentioning that if discordance is 1 for a criterion, then obviously we have no confidence in the hypothesis that *aSb* and therefore S(a,b) = 0.

8.4. PROMETHEE (Preference Ranking Organization METHod for Enriched Evaluation)

A method based on evaluation per criterion. It introduces the concept of unicriterion and global flows. The participation and involvement of the DM remains strong. The method follows three main steps:

- Preference degree calculation for each ordered pair of alternatives on each criterion
- Unicriterion flows calculation
- Global flows calculation

The method supports ranking which will occur based on the global flows calculation. For the first step we must calculate the preference degree which is a score that indicates how an alternative is preferred over another according to the DM. High preference degrees implies strong preference and if there is no preference then the degree is approximately close to zero. The preference degree is performed pairwise by measuring the difference of between the evaluations of the two alternatives. The preference function measuring the differences can be selected various possible choices, such as the linear or the Gaussian function. Preference of A over B does not lead us to a safe conclusion for the preference of B over A. After the calculations a preference matrix is created, but when the number of alternatives is large, procuring reliable results is complicated.

To that end we denote a set of actions or alternatives to be ranked as $A = \{a_1, a_2, \dots, a_n\}$ and let $F = \{f_1, f_2, \dots, f_m\}$ be the set of criteria. The preference degree P_{ij}^k , sometimes also noted as $P_k(a_i, a_j)$, is computed for every ordered pair of $(a_i, a_j) \in A$ and indicates how strongly a_i is preferred over a_j based on criterion f_k. Let also p and q be the the indifference and preference thresholds respectively and for the linear preference function we have [60]:

$$P_{ij}^k = \begin{cases} 0, & if \ \ f_k(a_i) - f_k(a_j) \ \le q \\ \dfrac{[\, f_k(a_i) - f_k(a_j) - \ q \,]}{[\, p - q \,]}, & if \quad q < f_k(a_i) - f_k(a_j) < p \\ 1, & if \ \ f_k(a_i) - f_k(a_j) \ge p \end{cases}$$

While if we choose the Gaussian preference function then we have

$$P_{ij}^k = \begin{cases} 1 - e^{\frac{-((f_k(a_i) - f_k(a_j))^2}{2s^2}}, & if\ f_k(a_i) - f_k(a_j) \ge 0 \\ 0 & otherwise \end{cases}$$

Uniformly, P_{ji}^k expresses how $a_j$is preferred over a_i according to the DM. P_{ji}^k and P_{ij}^k comply to the condition $0 \le P_{ij}^k + P_{ij}^k \le 1$.

After calculating all the unicriterion preference degrees we can compute the global preference degree π_{ij} , including to the equation the weights affiliated with each criterion. Let $w_k$be the weight associated to criterion f_k. If the weights respect the formal condition $\sum_{k=1}^{q} w_k = 1$ then

we have:

$$\pi(a_i, \alpha_j) = \pi_{ij} = \sum_{k=1}^{q} w_j \cdot P_{ij}^k$$

The above formula captures the global preference of a_i over α_j according to the whole set of criteria. The global preference degree varies between 0 and 1 and is subject to constraint $0 \leq \pi_{ij} + \pi_{ji} \leq 1$. The global preference degree can lead us to some conclusions. If both π_{ij}, π_{ji} lie around zero this can be translated as indifference, if both have about the same value not equal to zero then this situation can be translated as incomparability whereas if there is a great difference between the two degrees $|\pi_{ij} - \pi_{ji}| \gg 0$ then we have a preference between the two actions. The above conclusions are not strictly defined but depend on the DM for proper interpretation. This is how the preference matrix Π is compiled and *Π(i,j)* depicts π_{ij}.

Consequently we must summarize the preference degrees calculating the positive flows, the negative flows and the net flows, which are scores that measure how an alternative is preferred by and over all other options.

The flows epitomize the preference degree to a single score for each alternative. Let $\Phi^+(a_i)$ and $\Phi^-(a_i)$ denote the positive and negative flows of alternative a_i respectively. They are given by the following formulae:

$$\Phi^+(a_i) = \frac{\sum_{j=1}^{n} \pi_{ij}}{n-1}$$
$$\Phi^-(a_i) = \frac{\sum_{j=1}^{n} \pi_{ji}}{n-1}$$

The positive flow is the sum of all corresponding degrees divided by their number minus one, as an action cannot compare to itself and therefore is the mean preference degree of alternative a_i over all others. Similarly the negative flow represents the mean preference of all other actions over a_i. The net flow is the epitome of the two flows and can be expressed as:

$$\Phi(a_i) = \Phi^+(a_i) - \Phi^-(a_i)$$

There are two methods to calculate ranking noted as PROMETHEE I and PROMETHEE II. The first provides partial ranking using the positive and negative flows and can in occasions lead to incomparability when there is no preference or indifference between two actions.

PROMETHEE II produces complete ranking as it uses only net flows which are axiomatically transitive.

In our system, due to the large volume of data, we preferred the PROMETHEE II, in order to exploit the complete ranking the method provides.

8.5. Chapter Conclusions

In the eighth Chapter the fundamentals of Multi Criteria Decision Analysis were discussed. The operating principles of preference modelling and of the methods SMART, ELECTRE and PROMETHEE used in experiments in this thesis were also presented.

9. Related Work

9.1. Machine Learning for Urban Planning

The work in [61] attempts to apply a machine learning mechanism for large scale evaluation of the qualities of the urban environment. The characteristics used for the learning mechanism vectors are based on the construction and quality of the building façade and the continuity of the street wall as obtained by the relevant street view images using machine vision techniques. The training examples are images labeled by experts and the evaluation results are compared to the public's opinion of the corresponding buildings, as obtained through an in-situ survey. The authors acknowledge the limited capabilities of the method due to the inherent problems of the source images (perspective, trees, deficiencies imperceptive by machine, etc.) and the possible inconsistency between the experts' and the public's evaluations. These problems, enhanced by the high complexity of the problem addressed, are reflected in the results, demonstrating low precision (<50%) and average recall (72-85%) capability for both observed qualities.

In [62] a procedure to mine points from social networks and feed them to machine learning techniques to estimate aggregated land use is presented. The researchers recognize the problems we debate related to the origin of data and its validity and reliability. However they deal only with 2d where the mined data consist only of points. The focus is on comparing the results of the machine learning algorithms with those of the census and the ground truth used is another proprietary data set, which does not exclude the existence of errors. The research is macroscopic, performed at a regional level without concentration on city infrastructure and relies on an already available software package (weka) without no additional customization or further development.

In [63] an architecture is proposed to exploit IoT based city sensors dividing each task to low, mid and high level and assigning it to a separate stage of the architecture. The city sensors used include smart home sensors, vehicular networking, weather and water sensors, among others, recording what could be classified as Big Data. The low levels are responsible for data gathering, the intermediate levels perform the task of communications between sensors and framework and the data management and processing whereas the higher level deals with data interpretation. There is no clear reference to the machine learning mechanisms used, beyond the automatic classification carried out by the ready-made system. In some experiments the data is

small (10-15 vehicles). The decision support system is not presented thoroughly neither has any visualization.

In [64] the urban planning problem of road network expansion and alteration based on existing traffic flow information is addressed. The work infers potentially useful road linkages between city zones that could alleviate traffic flow and, subsequently, improve quality of life and productivity. Alternatively, the load of certain zones participating in high traffic flow yet revealed to be indirectly and, thus, inefficiently connected, may be reduced and transferred to other zones aiming to a more efficient distribution. Data is collected through bluetooth sensors deployed across the urban area. The proposed model is claimed to be also applicable to new housing, construction, or economic activity, under the limiting condition that these processes will be adequately captured as correlations between zones to guide new routing of the traffic network or load redistribution.

9.2. Visualization of Urban Environments

A conceptual framework for urban or regional development design is presented in [65]. The authors' proposal relies on multifractal modelling in compliance with a number of urban planning principles. A multifractal Sierpinski carpet representing a hierarchical nesting of central places (i.e. urban centres) serves as the theoretical reference model. Fractalopolis GIS-based software [66] is employed to support the application of the concept in relevant case studies. The approach concentrates on the issues of urban development in the sense of expansion/contraction and building density. Moreover, it relies on the presence of relevant data in the form of six shapefiles including buildings (represented by polygons), public transport stations (represented by points), shops and services (represented by points), leisure facilities and green areas (represented by points), non-developable areas (represented by polygons) and the current number of housing units in each local community. The rich semantic content considered as input (e.g. the kind of each service and the frequency of its use) allows for efficient quantitative evaluation of the computed plan and adequate 2D visualization of the plan itself and its efficiency, whereas the approach is synthetic, in the sense that it is producing integrated development plans yet not supporting queries on the efficiency of specific locations and uses.

The work in [67] attempts to visualize the potential sprawling of urban areas. A virtual environment accepts as input the geographical orientation and topography as well as growth of buildings. Urban growth consists of new buildings generated and checked against environmental

factors and attractiveness of location whereas a communal social behavior is programmed to govern the overall building generation. The idea is pertinent to urban planning and relevant decision making and the authors claim similarity of resulting patterns with real urban forms. However, semantic information with respect to buildings is neither exploited nor generated in the process whereas it is admitted that the rules employed in the virtual urban environment generation mechanism are not derived from actual urban development experience.

The work in [68] concentrates on incorporating semantic information to produce visually appealing 3D models. The latter is achieved by maintaining planar shapes when originally present even in imperfect form, adopting straight building outlines and focusing on detailed building representation while allowing for less detailed surroundings. The semantic content is assigned by previously trained machine learning mechanisms and it is exploited to improve the image recognition and 3D reconstruction process. The achieved accuracy is balanced with a compact and visually appealing 3D reconstruction. The results are also acknowledged to be of decision making interest to certain stakeholders like real-estate agents, however the effort towards any decision support functionality is limited to the care for the aesthetic level of the 3D outcome.

In [69] the emphasis is on the efficiency of the visualization due to the large scale of urban data. Similar to the current work, the visualization is applied on a virtual globe. The work is supportive of a framework for problem solving in urban science presented in (The Framework for Problem Solving Environments in Urban Science). The latter presents a higher level proposal for such a system, integrating and exploiting current capabilities including GIS, heterogeneous data aggregation and efficient visualization. While the proposed framework is wide in scope, the presented implementations of it are limited, not implementing a large part of its functionality. In comparison, the work herein enhances the proposed framework with decision support powered by machine learning techniques while offering an implementation covering the proposed functionality in its entirety.

9.3. Semantic Information Exploitation

In the field of semantic exploitation of urban scenes we have several different approaches. [70] uses multiple sources to reconstruct complete urban environments and enhance the scenes with semantic information. However, in most real scenarios, it is extremely difficult and improbable to acquire or possess that amount of data. Furthermore, as part of evaluation for

the proposed method, simulated cities constructed with pseudo-random synthetic data were used.

In [71] segmentation mechanisms combining GIS and VHR images is used to semantically classify buildings. However, the lack of appropriate real world samples creates imbalanced data sets which influences the classification results. The categorization of buildings and their variations is limited and constrained in the sense of the amount of the semantic information they administer.

In [72] a method is proposed so that generated meshes from multi view imagery that present some advantages over LIDAR can be semantically classified. The aforementioned work is formulated in the photogrammetry domain and differs from our proposal in goal and formulation.

In [73] is explained how CityGML works. It aims to describe a whole city, so it is quite extensive but does not deepen on specific features. The central entity is an abstract building that has geometric characteristics but is not rich in non-geometric information. In addition, there are various levels of accuracy (Lod) that are not affected by the extra information we add.

In [74] an interesting description is provided in the part of the process of enriching a model with semantic information. However, it deals with the fragmentation of footprints in buildings and matching them with existing in databases, which again goes beyond the scope of the current paper.

9.4. Chapter Conclusions

In the ninth Chapter the related work of other researchers in the relevant fields of Machine Learning for urban planning, visualization of urban environments and semantic information exploitation were examined and differences to the current thesis were debated.

Part II: Thesis Contribution to Semantic Querying, Navigation and Spatial Decision Making of 3D Urban Scenes using Machine Learning

10. Thesis Contribution to Semantic Querying, Navigation and Spatial Decision Making of 3D Urban Scenes

10.1. Urban planning challenges

The diverse sources of data that can facilitate urban planning stakeholders, decision makers and other participant actors offer the opportunity to develop and experiment with actual mechanisms for semantic modeling and decision support in realistic operating conditions. This leads to the first aspect of the problem, that of efficient and accessible merging and presentation of several features in an integrated environment. Urban planning requires comprehension of infrastructure and its surrounding environment both in terms of low level physical characteristics (such as geometric features of buildings and their arrangement) as well as higher level concepts (e.g. standardization and categorization of buildings and their uses, land use modes, usage of road networks).

A major issue faced by urban planning experts is that of standardization. The need for establishing standards in the field is so immense and suggested by numerous researchers and experts that international organizations have been created for the sole purpose of managing this issue,[75] [76]. All these data are generated by different sources, encoded in different formats and, usually, not exploitable in their initial form. Furthermore, it is not always possible to convert the data to an exploitable form, while, even when the conversion is feasible, the hazard of data alteration or corruption during conversion is always present. Most often, it is extremely difficult to verify the correctness of the process, partly due to the huge volume of data, since it renders a human-performed visual control practically impossible, but also due to copyright issues. As a result, access to the raw data is not unobstructed, thus greatly impeding the process of verification of the outcome.

Another issue is the origin of the data in regard to the authority of the provider itself, the validity of the data along with their age. The validity of the data is directly related to the entity that collected or implemented them, the methods and technical means used to collect the data, as well as the amount of time elapsed since collection. It is often the case that there is no information concerning the issues above, but even when it is provided, the data may have been rendered obsolete due to their age. A city is subject to constant changes, as shops and businesses open and close, roads are converted to bidirectional or unidirectional, new buildings are built and other demolished, to name a

few examples. Data older than a certain limit are likely to be obsolete and this time limit cannot be precisely defined as it depends on many precarious and uncertain factors.

Since we examine urban data, it is undoubtful that some of the data sources will originate from open data, which are themselves a subject of study and controversy as their use still presents some predicaments. Further, the emphasis on open data is given on the data itself and not on how it is used or exploited. As a result, most providers focus on providing data rather than providing the means to exploit the data, or a system that facilitates data manipulation and queries [77]. Finally, some types of data cannot become publicly available for legal reasons, which may severely impact on the exploitability of different, but in some way linked, types of data.

10.2. Open Data

Open data as defined by the open definition is “data that can be freely used, re-used and redistributed by anyone - subject only, at most, to the requirement to attribute and share alike”. Use of open data may involve various advantages that lie in economic areas (e.g. providing economic growth and stimulation of competitiveness), political and social areas (e.g by providing more transparency and democratic accountability, and improving the participation and self-empowerment of citizens) or operational areas by improving administrative processes.

However, the exploitation of open data in an environment or application other than that of its provider or distributor is an endeavor far from trivial and several difficulties are encountered in the process. Interoperability is one of the largest complications. As mentioned earlier, there are no commonly accepted standards in creating and distributing open data so each organization depending on the technological means available ends up coding and sharing the data in a different way. In addition, organizations dealing with open data place the emphasis on data itself rather than on disposal or on tools to exploit them [78].

In addition, as is the case with most data sets the quality of the data is not unquestionably guaranteed and it needs the intervention of human acuity so as the information can be utilized for certain objectives. Data can simply be noisy or simply incorrect or present other deficiencies. There may also be insufficient information about the data in the sense of metadata, ie information on the date of acquisition of data.

Directly associated with the quality of the data is the process of updating and keeping them up to date which is often not performed. There are data sets subject to changes over time, such as urban, which is the interest of this sector.

As a consequence, the data set could be outdated or obsolete and therefore inappropriate for use, but due to lack of disposal information we may not be aware. Most organizations providing open data emphasize on the data itself themselves more and more on how to dispose of or provide tools for their exploitation.

10.3. Evolving technologies, emerging applications and its contribution to urban planning

Urban planning is a problem towards the resolution of which, in recent decades, developments in various scientific fields have contributed enormously. Special mention should be made in the contribution of fields of computer science where research has resulted in software products consequently used for mapping, modeling, storing and analyzing information.

However, the same thing cannot be said for machine learning, one of the most rapidly advancing fields in computer science the last years. The great strides, enabled mainly by the advent of deep learning, have brought about revolutionary changes in a variety of fields, such as computer vision, [79] and [80], robotics [81] , text analysis [82] and [83], financial market analysis [84] and [85], biology [86] and medicine [87], physical sciences e.g. physics [88] and chemistry [89], and recommender systems in various domains [90]. On the contrary, the use of artificial intelligence in urban planning and development has been far more limited. State of the art artificial intelligence methods can now be appropriately adapted, fine-tuned and used to exploit the aforementioned increasingly available heterogeneous “urban big data”. Results of such intelligent data analysis can then be used as input to the decision making process by urban planning stakeholder.

10.4. Decision Support and Contribution

In this thesis we present a system that can fuse various types of data from different sources, encode them using a novel semantic model that can capture and utilize both low-level geometric information and higher level semantic information. Among the open data providers and sources, there are public organizations dealing with urban planning (e.g. Estate Property Agency).

One of the main problems affecting urban planning is the appropriate choice of location to host a particular activity (either commercial activity or common welfare service) or the correct use of an existing building or empty space. The most frequently asked questions posed by stakeholders concern finding a suitable site for the construction for example of a new school or the construction of a new hospital, while discussion is made on the methods and implementation procedures bearing in mind the public interest [91]. Experts need to take into account a variety of factors, such as population distribution and composition, transport coverage and of course the cost, availability of buildings and spaces and much more. Similar problems are encountered in finding a fitting site for a specific commercial use (e.g. finding a place suitable to open a restaurant or deciding on the suitability of a particular site). Such problems are the focal point of our work.

In particular, the proposed work yields the core of a decision support system, which, in turn, dictates the need to maximize the degree of automation. In this thesis, the formulated problem (suitability of a building or space for a specific use) is treated as a classification problem. We propose the use of random forests classifier, because they tend to be invariant to monotonic transformations of the input variables, and are robust to outlying observations, which are often encountered in the discussed urban data. We also scrutinize the effectiveness of a wide range of machine learning classifiers, such as Support Vector Machines, Feedforward Neural Networks, Naïve Bayes, and other.

In addition to big data management and intelligent decision support, the proposed system also offers a visual interactive environment using current visual techniques (Figures 28 and 29).

Figure 29 - Visualization of a geoquery by our system

The inherently large volume of urban data and their type, mostly comprising three-dimensional geometries, makes them practically impossible to conceive in their raw form and strongly suggests their rendering and visualization, a challenging but essential process towards their full exploitation.

In problems of such kind, an important factor towards attaining the best possible solution is human intuition and pertinent visualization intensifies human perception facilitating the

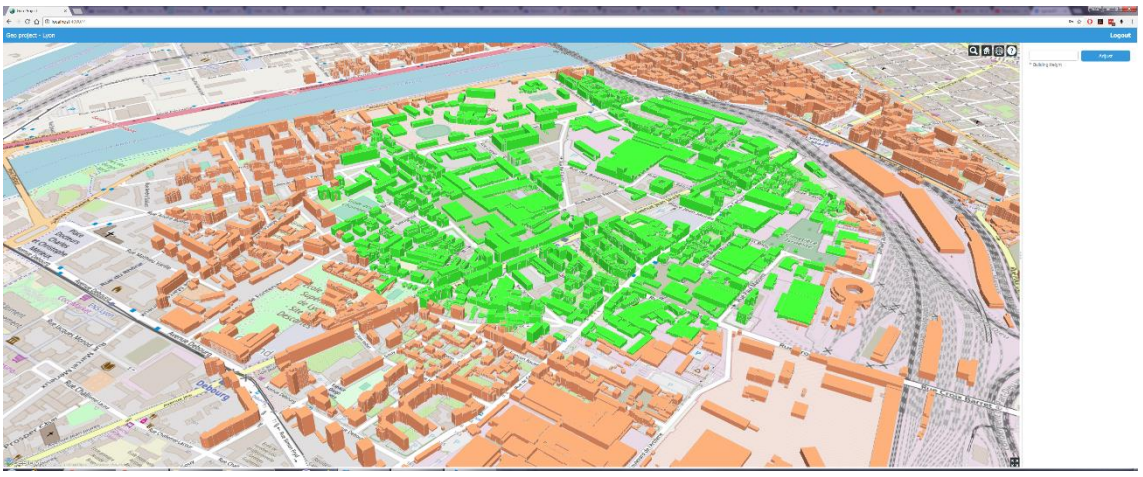


Figure 28 - Altered point of view visualization of a geoquery by our system

process at hand.

10.5. Chapter Conclusions

In the tenth Chapter the urban planning challenges and the possibilities of using artificial intelligence to resolve them were further discussed. The problem of the appropriate choice of

location to host a particular activity or the correct use of an existing building or empty space was posed and treated by scrutinizing the effectiveness of several machine learning classifiers and proposing the dominant.

11. System implementation

11.1. Problem Revision and System Overview

11.1.1. Revision of Problem Formulation

To ensure maximum practical value and exploitability, our system is based on the use of real-world open urban data. After meticulous study we observed that there are numerous open data associated with parking spaces, in the vicinity of the wider geographic area we have selected for our experiment. The suitability of a given space for use as a parking space is a question that meets the requirements of an urban planning problem and additionally has a strong commercial interest. The existence of a tool that can recommend a potential appropriate use of a space or building or make a prediction as to the suitability of an area/building for specific purpose, can be a useful decision support tool for an expert.

Having real-world parking data does not guarantee in itself that we automatically have the knowledge about the salient information therein with regard to the decisive factors that contribute to making a parking lot useful, essential or profitable. In other words, the feature extraction process in this case is far from trivial. In this context, a variety of factors will be explored as potential descriptors, including: distance from landmarks, distance from other parking spots, density of occurrence per specific area, distance from means of public transport along with their plurality in a certain area, distance and density of occurrence with respect to points of touristic interest, economic and monetary points of interest among others.

11.1.2. System Components

The proposed system consists of different components, and operates in distinct stages, as shown in Figure 30. The progression and transition between stages follows sequential procedures for some, while others are being processed in parallel.

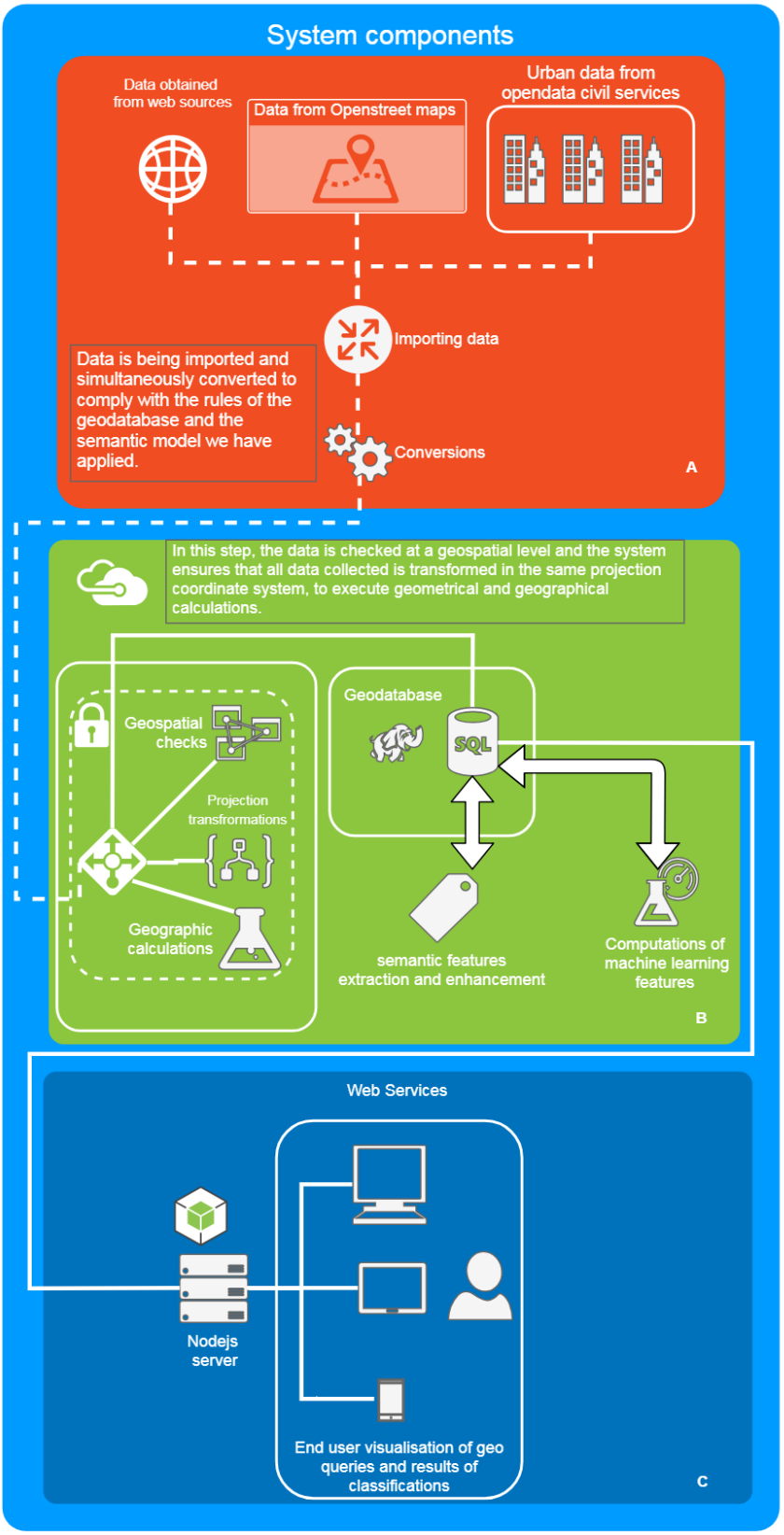


Figure 30 - Functional Block Diagram of the proposed System

After reviewing the existing visualization implementations of corresponding data types, we notice that many attempts have been made with variating purposes, emphasizing at diverse geographic mainly features (terrain, specific water or land masses, etc.) but none has all the desired features. Most software solutions provide data viewing, search capabilities, spatial queries support and visualization of the query results, but experience different disadvantages. Querying geo data requires both advanced specific technical knowledge and also familiarity with the structure and organization of the database used and its implementation platform, as most commands are platform-specific. Furthermore, the visualization is primarily done in a 2D environment, limiting the use of human perception of the three-dimensional space.

The system we propose aims to remedy these gaps, providing solutions in a way easily accessible by anyone as we have chosen to use technologies who can implement these very principles. Whereas geospatial databases require specialized skills to operate and manipulate, we have implemented an interface that renders those skills not mandatory, taking under consideration and exploiting semantic information underlying in the data.

After performing numerous tests, we concluded the best course of action was the use of web-based tools and platforms (javascript , NodeJs, Html5) [92] in order to reduce not only skill dependencies but also the prequisities in software installed in a device and even the requirements for computational power and other technical characteristics of the device itself, as they perform excellently in cross platform and mobile applications.

Part of the aforementioned tests was a first attempt to visualize query results in 3d using html5 and library 3js as most frameworks lack the ability for 3d visualization, as seen in Figure 31.

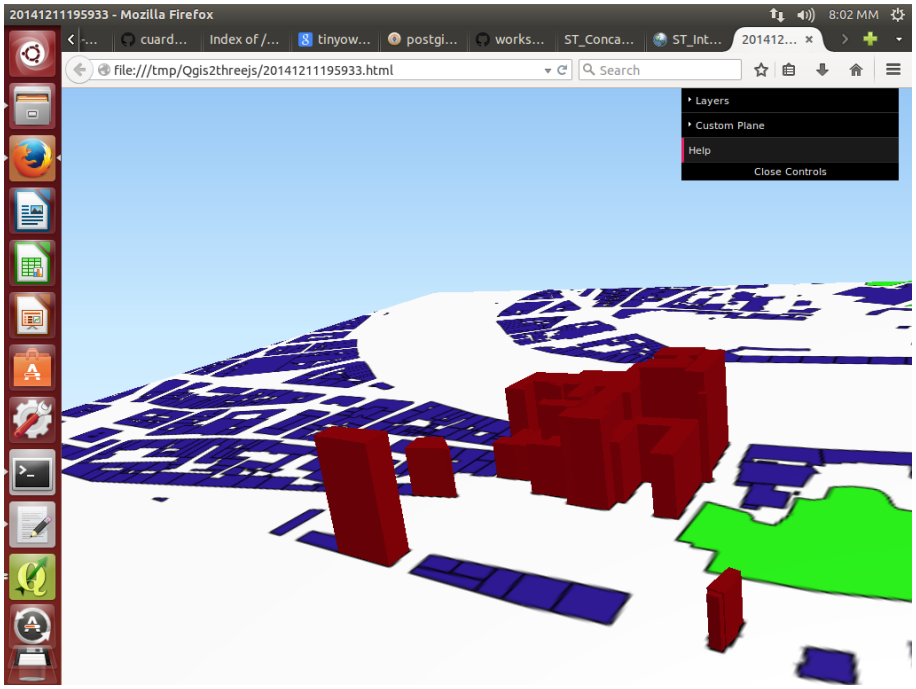


Figure 31 - 3d visualization of query results using 3js library

It is obvious that it had numerous deficiencies, so we moved on to a second attempt using a different viewer (Figure 32)

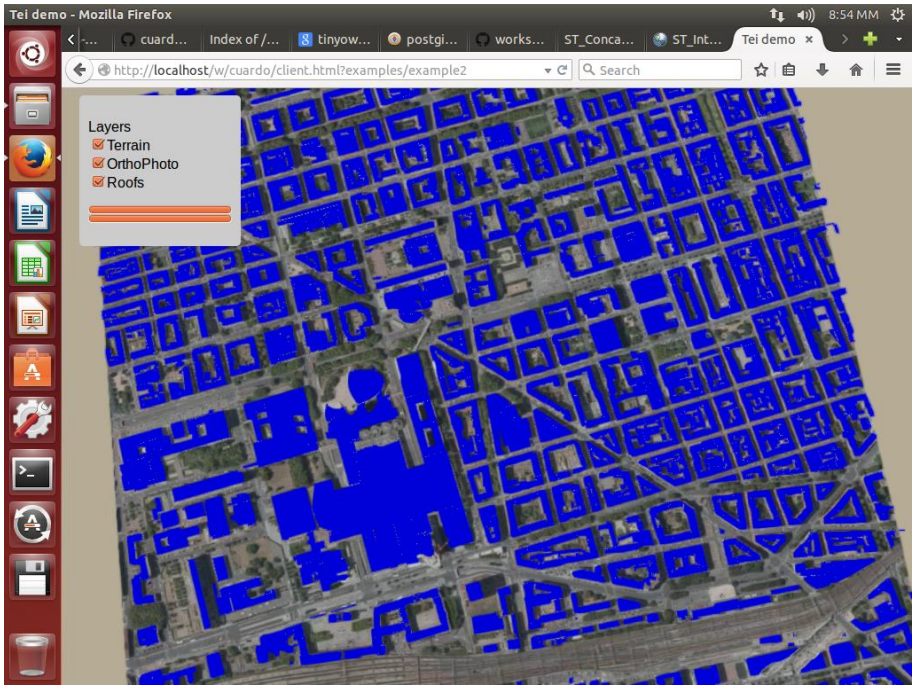


Figure 32 Visualization of geodata using cuardo viewer

Since the inception of the system our goal was to create an interface that has almost zero requirements in knowledge and technical skills, concealing confounding details of the system architecture and mechanisms yet offering substantial functionality. Furthermore, our assessment for the design impose that it should be accessed and operated even on low-powered devices such

as smartphones and tablets. It has been several years now the majority of internet traffic originates from mobile devices, thus leading to rapid developments in hardware specifications for these devices pursued by corresponding enhancements and amplifications in software, resulting to the fact that by now the computational power of the aforementioned devices is far from negligible.

We built a system that combines the benefits of GIS and mapping oriented software with those of 3D object visualization environments. Most GIS /mapping systems focus on its own corresponding features, namely viewing, exploring and analyzing data and composing maps. It is worth mentioning that the aforementioned systems are used principally by specialists and field experts due to the requirements for their management, while systems that can be used by someone without technical knowledge usually provide simple map view with very limited search capabilities. A flowchart of the system can be seen in Figure 30, depicting the several stages of processing required.

The first stage deals with the collection of data. We benefit from open data and evoke the creation or gathering and distribution of them by government institutions. Government and civil organizations and bureaus related with real estate, geographic and town planning services have embraced the principles of open data and provide such.

The initial step of the first stage involves collecting and processing the data in order to prepare it for its import to the database [93]. The system we propose allows and provides for input of different types of data originating from different sources and consequently diverse formats such as raw datasets, provided as plain files or in a more difficult to exploit form through application programming interfaces (APIs) bestowed by web platforms, or even online map services.

For the initial step, the data derived from the city of Lyon. The dataset comprises of around 800000 buildings including water masses, parks and forests obtained from the opendata service of the city. The geodatabase used was postgres with postgis extension. For the initial manipulation of the data Qgis was also used and Ubuntu as the OS in an effort to use only open source software. In Figure 33 we can see the import of buildings after the creation of database

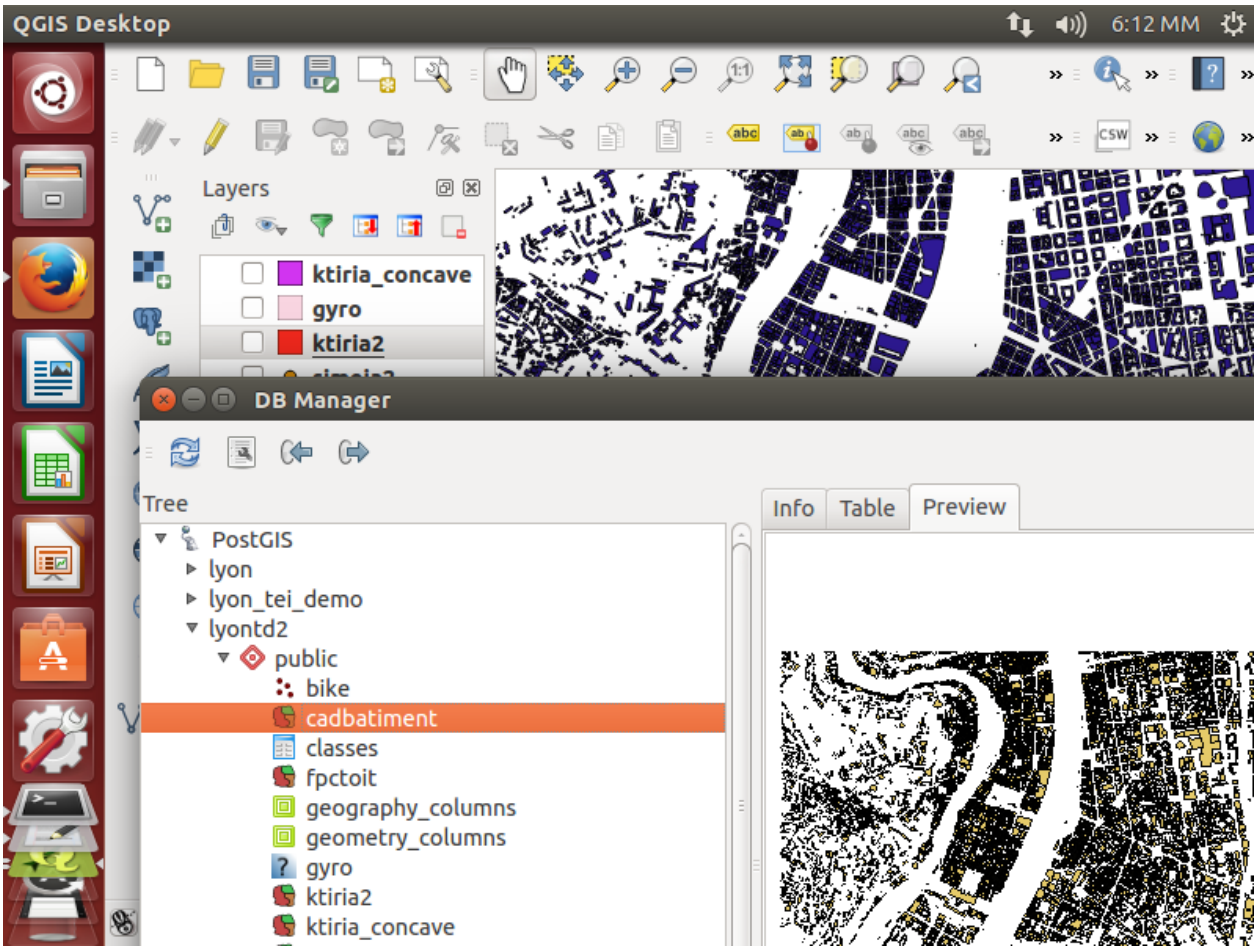


Figure 33 - Import of buildings

Wherever possible, to offset the risks arising from the use of open data we mentioned above, the candidate data sets are put in comparison. Where there are multiple sources for a certain area, a cross checking of information is performed and the components - matching buildings are forwarded to the next control stage and placed in the database. If a difference occurs then visual inspection is performed, and along with other criteria such as the date of acquisition they are aggregated to arrive at the final conclusion.

Data entry does not guarantee validity since the data are georeferenced and may be encoded in different ways and in a different coordinate system depending on the organization providing them, whereby following several checks that control and convert, if necessary, data into a common coordinate system.

In an effort to perform some basic checks on the data we conducted some primitive spatial queries. In Figure 34 we can see the results of such a query- finding buildings with the largest area in the city of Lyon.

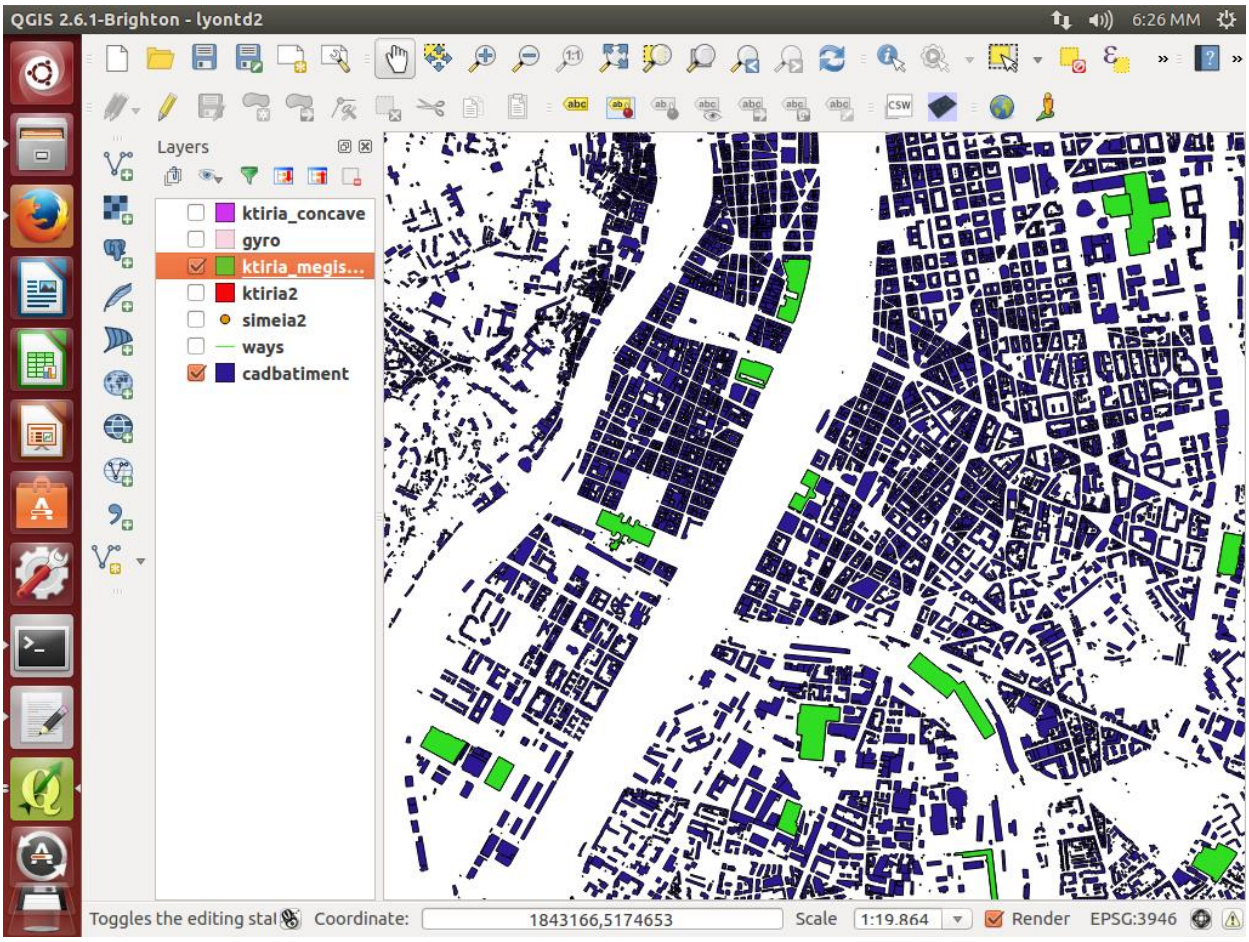


Figure 34 - Spatial query example

There is a very small percentage of data (in our test cases 4-6 buildings out of 800,000) that during the conversion presents errors and incorrect geometrical characteristics, which is inevitably rejected. However, this percentage is negligible (0,00000625 %), as it does not affect neither the size of the set or the credibility of the method.

We can also perform a query to test the exploitation of semantic information imported to the system and the ability for visual verification of the results with Street View as seen in Figure 35.

The next step is the normalization and homogenization of the coordinate system for each of the various data segments, so that geometric operations and geographic correlations can then be applied to correctly update the system and subsequently feed and inform the ontological model with the correct information. As expected, each organization provides its data in different format and diverse systems of geographical coordinates. The data are subjected to specific queries to update the relevant tables accordingly. Finally special queries are used to extract

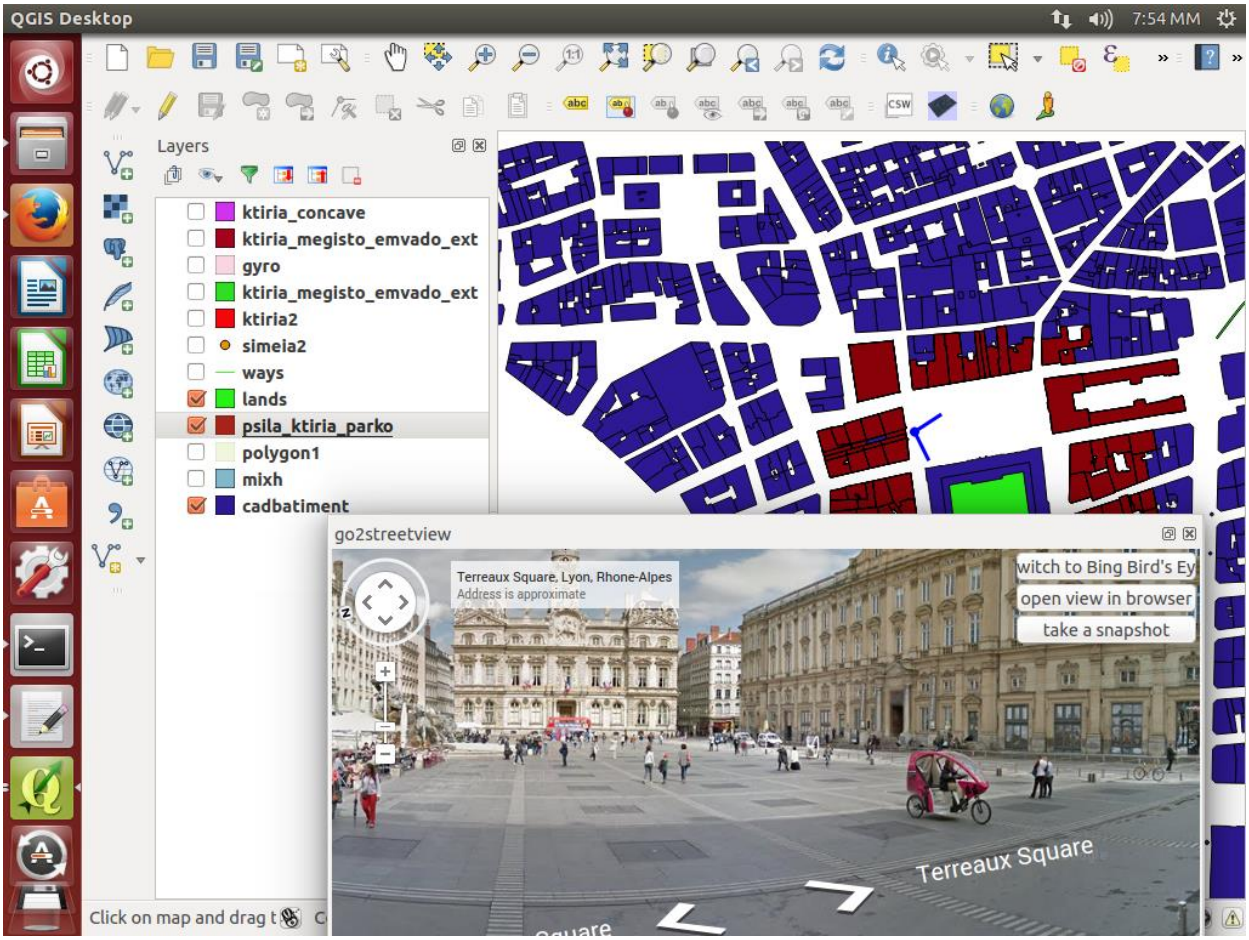


Figure 35 – Visual verification of query results

semantic information from the data provided updating our geodatabase.

11.1.3. System Technical Specifications

What we endorse is not the use of some ready-made software but the construction of one, completely custom extending the capabilities of an existing platform. As a starting point we used the Cesium framework [94] which is an open source library that provides necessary tools for our purpose. It was complemented using JavaScript, html5 and geojson [95] technologies in points that we will discuss later.

Our implementation is divided into 2 basic segments, the database that manages the geospatial data and implements the semantic aspect of the data, and the component of visualization of data and interface with the user.

Regarding the first segment, there does not exist a large variety of implementations that can natively store and manipulate geospatial data inherently. We used PostgreSQL enhanced with plugins such as PostGIS that implements geospatial functions (spatial operations,

calculations of both Euclidean distances and distances considering the curvature of the Earth, areas, paths etc.).

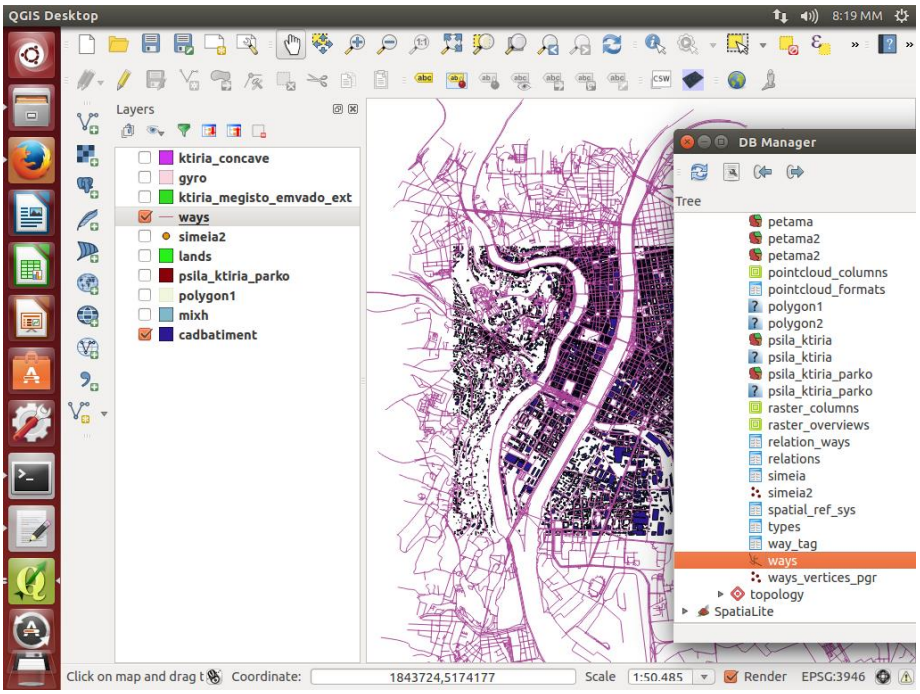


Figure 36 – importing the road network of the city from openstreetmap

In conjunction with several plugins that provide functions for this purpose, it is possible to provide routing and navigation by implementing most common algorithms (A *, Dijktra etc) as seen in Figures 36 and 37.

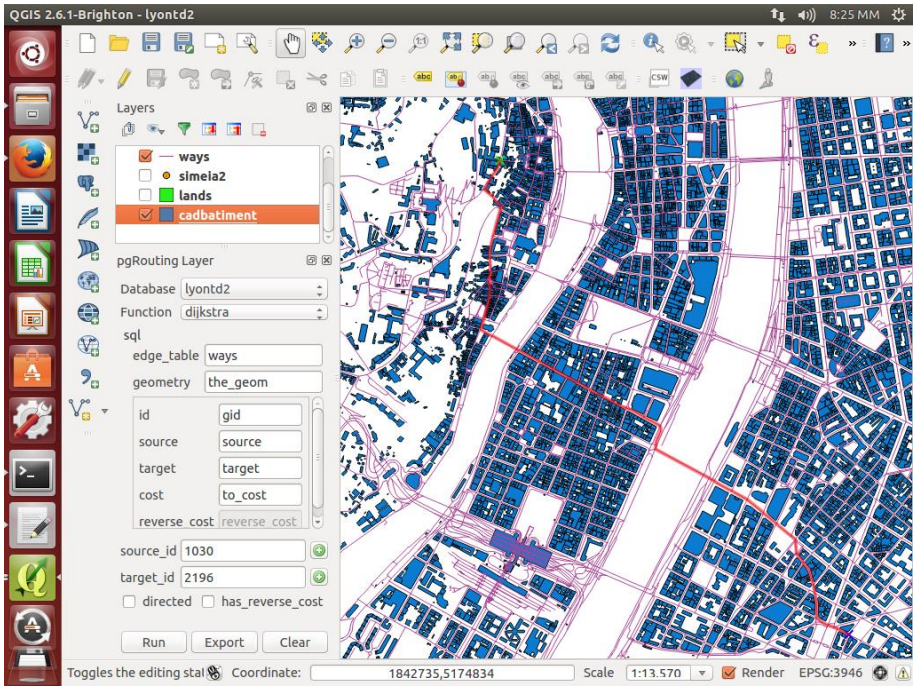


Figure 37 - Using Dijkstra algorithm in road network to find shortest route between 2 user selected points

PostgreSQL together with its PostGIS extension and the spatial functions the latter provides constitute a powerful combination for managing and querying the data related with the current work. They can successfully handle the arduous yet essential tasks of storing and manipulating georeferenced buildings as well as an extensive road network with all the information needed to make it routable and semantically enhanced. They are both open source, and experience widespread dissemination and support from the respective community.

For the interface with the database containing the building information we had to develop a special API, which undertakes the conversion and transfer of data using asynchronous functions, as well as the transfer, conduction and handling of user derived events from the viewer to the database and vice versa.

This API uses json technology and converts the database data volume into a corresponding stream so the web component of the system may visualize it.

For the part of visualization of the results, following a thorough study of numerous other candidate solutions (three.js, sfcgal, qgis html viewer), as a starting point we used the aforementioned CesiumJs libraries, a geospatial 3D mapping platform for creating virtual maps globes. It presents to have some clear advantages regarding the work at hand compared to other platforms, as it provides a complete and documented set of tools that can be used to expand its capabilities. It can exploit the capabilities of the aforementioned systems, namely to be able to execute and visualize spatial queries, but also to allow the management of three-dimensional georeferenced objects, their retrieval, transformation and visualization.

As server runtime environment the NodeJs platform was chosen as it is lightweight, platform independent and doesn't create unnecessary dependencies nor does it burden the machine where it is installed with unnecessary libraries, as it creates a separate folder and is restricted there.

11.1.4. System Implemented Features

The system front-end can be accessed by a plain browser, providing a 3D urban visualization environment that can be freely browsed around in all dimensions, rotation of the

camera and zoom capabilities (Figure 38).

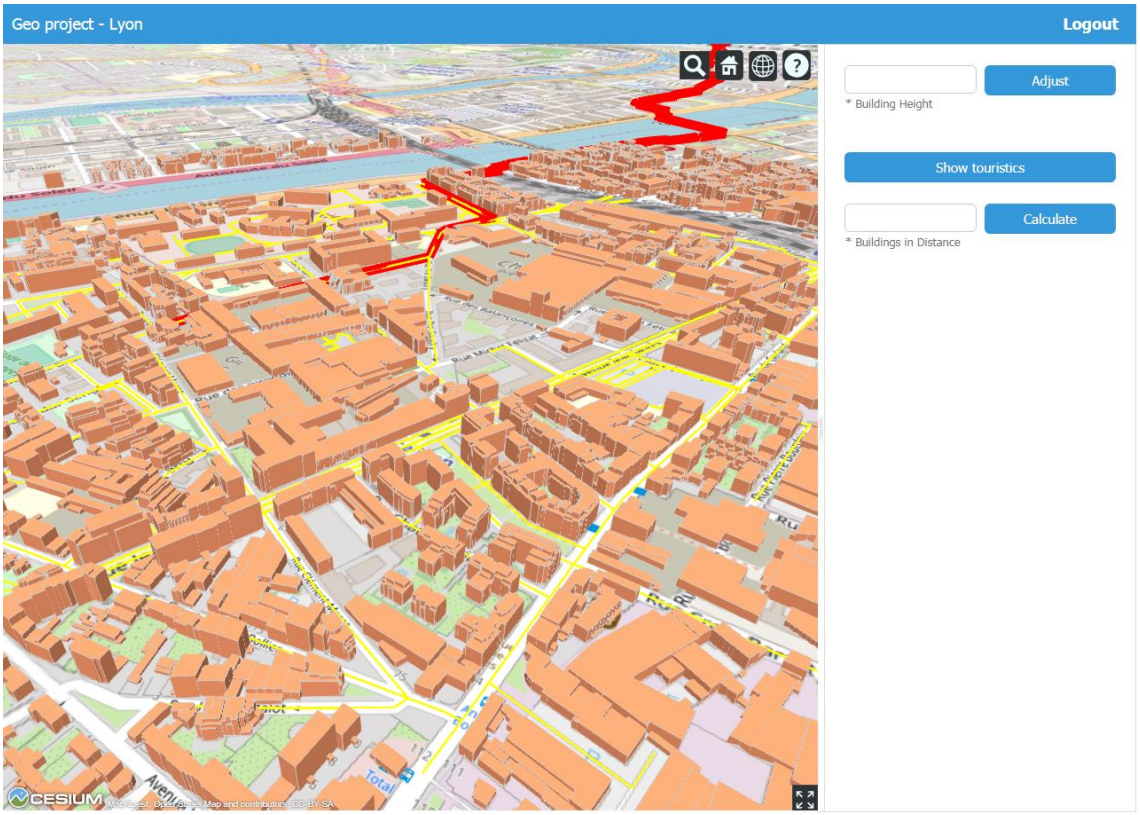


Figure 38 - Front end instance overview. We can observe the underlying map terrain layer, the buildings retracted from the geodatabase and visualized by our system. The thick red line represents the ability to calculate and visualize an optimal route between points.

In terms of urban environment we mean visualization of buildings and structures enriched with information about them (e.g. height or media associated – photos, documents, schematics), road network, routes, points of touristic or financial interest, and the possibility to alter the subsiding terrain map.

The user has the option of selecting a particular building, point or region of interest and viewing its accompanying features (for example its height if it is a building, or it's unique gis code) as they are stored in the geodatabase in real time. Upon selection of a building the associated media with the selected item (e.g. photos, schematics, documents), if any exists, appear on the side on a specific area of the browser application (Figure 39).

The user has an option to view and manage these media as with any modern web application (zoom, download in a local folder etc.). It is worth noting that the media are maintained in the geodatabase as well.

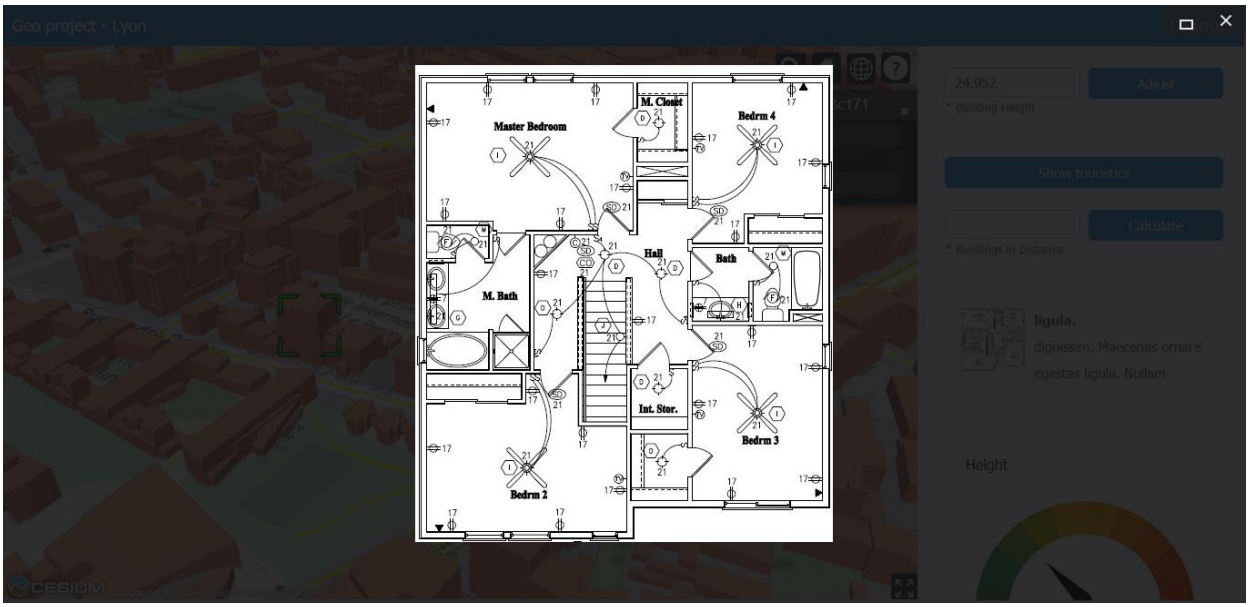


Figure 39 - Visualization of media related to the selected building

Furthermore various layers can be added or removed at the user's request, and certain attributes can be altered and rendered at real time. The user can choose to add or remove if he

Figure 40 - Addition of points of interest

wants some levels of information such as points of touristic interest or the road network in order to adapt to the specific needs of each individual use or to decongest the working interface

or a low processing powered device (Figures 40 and 41).

Figure 41 - Removal of points of interest and resetting the interface to its original settings

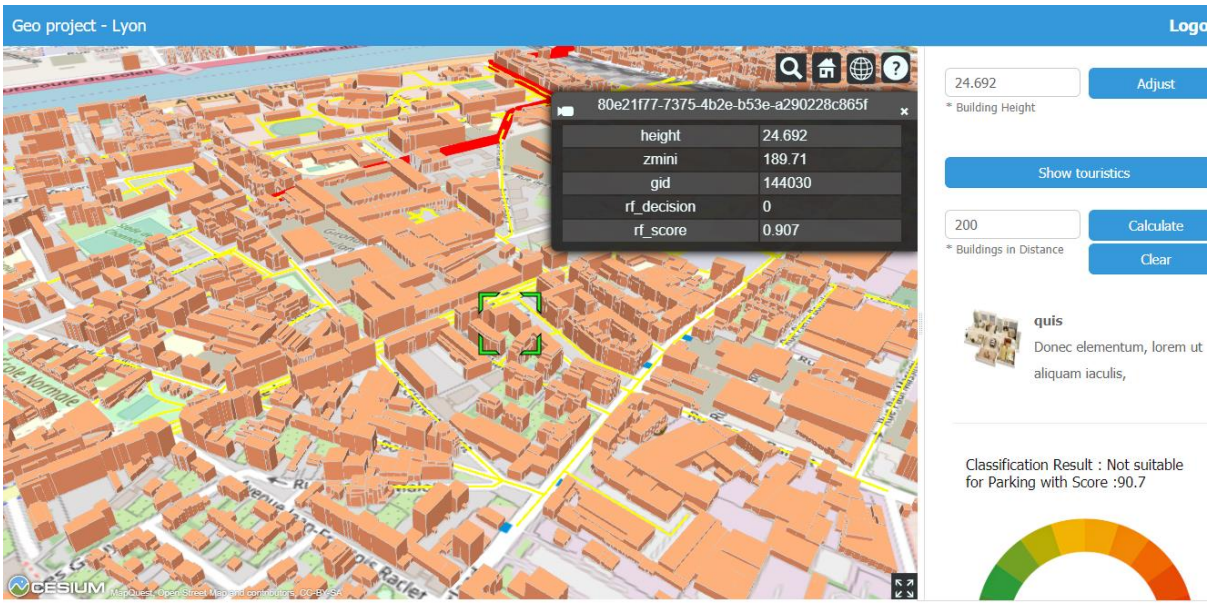
It is worth noting that the data visualized is derived from the geodatabase described earlier and is therefore dynamic as opposed to a static file which is the customary method followed in relevant endeavors. The height of a structure, for example, can be retrieved from the database, altered and visualized again (Figure 42).

(a)

(b)

Figure 42 - (a) Original height of the building, (b) Altering the height of a building

The approach employed herein also provides the possibility to implement spatial queries in an intuitive visual way without requiring as mentioned earlier expert knowledge of the field or the need to write code or commands. For example upon selecting a building and entering a distance in a specific field of the application's front end we can request a query for calculating and visualizing the buildings located within the given radius (Figure 43). Afterwards we may of course clear the results and return to the original view.



(a)

(b)

Figure 43 - Custom spatial query (a) Selection of a building and desired distance (b) visualization of results

Furthermore, as we already mentioned, our system provides Decision Support functions and can visualize the results of machine learning techniques experiments, showing us the result of accuracy for each building selected for the preselected use as a parking (Figure 44).

Figure 44 - Building with classification score provided by machine learning techniques

These attributes extend the basic capabilities of Cesium viewer as they are not supported inherently.

11.2. Chapter Conclusions

The eleventh Chapter is dedicated to implementing the proposed system and presenting its components for data representation, manipulation and visualization. The technical aspects of the system and its functionalities were also displayed.

Part III: Evaluation Discussion and Conclusions

12. Evaluation

12.1. Experimental Evaluation of Machine Learning Techniques

In this Section, we scrutinize the effectiveness of the proposed methods using real-world urban data from the city of Lyon. A series of machine learning techniques have been examined and compared in terms of their efficacy in accurately predicting the suitability of a location/building for a particular use

12.1.1. Urban data description and feature extraction

Each building, after being successfully imported, is represented in the database by heterogeneous data ranging from concrete geometric properties to semantic information. In particular, for each building identified as unique, the following information is available or may be extracted:

- Dimensions
- Location
- Use
- Material
- Address
- Area
- Height
- Semantic information : use, proximity to other landmarks like media transport stations, places of touristic interest, green areas or rivers, ATM, parking areas
- Media of building (e.g. photos, schematics, contracts).
- Distance to any other building or landmark, both Euclidean or based on shorter route algorithms like Dijkstra

The visualization of feature extraction from our system is presented in Figures 45 and 46.

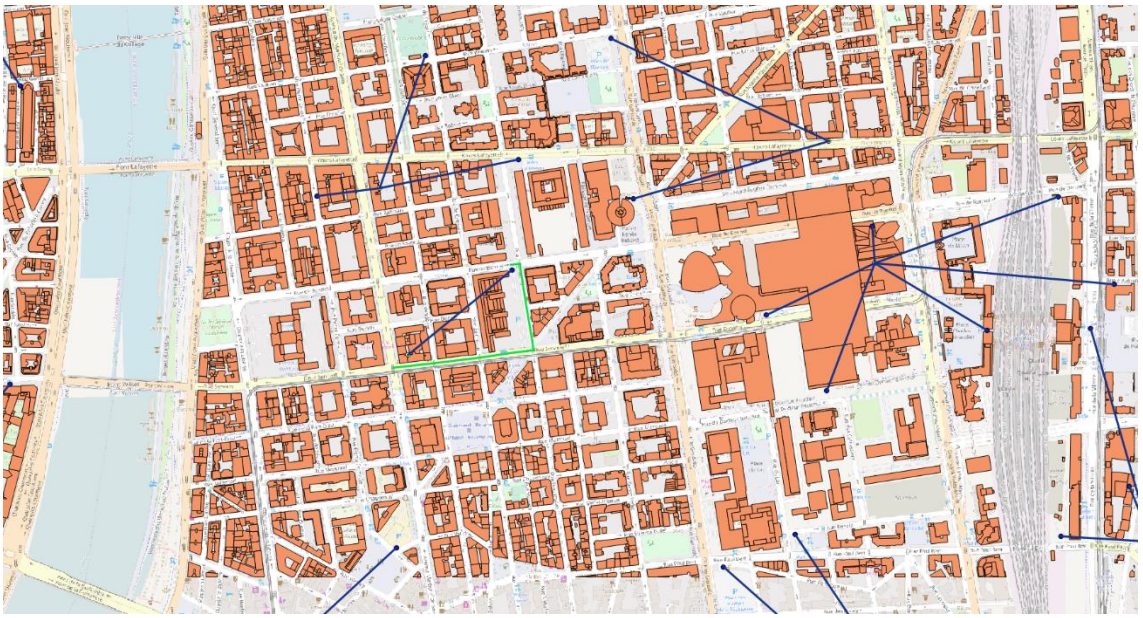


Figure 46 - Feature extraction: distance from nearest atm

Figure 45 - Euclidian distance vs routable road distance

Let the set of buildings:

$$B = \{b_1, b_2, \dots, b_k\}, \quad k = \text{total number of buildings in database}$$

In order to be able to apply and evaluate the selected machine learning techniques to the current context, we first need to isolate representatives of the two classes that will be the subject of the mechanism's functionality. In the current work, we focus on the eligibility of a building to be used as a parking service or enterprise. With respect to the need for positive examples, we have used the real-world information contained in the database concerning the actual use of buildings designated as parking lots. Therefore, for the positive examples we have:

$$P = \{p \in B, u(p) = parking\} \tag{14}$$

where u(p) indicates the use of the building. Similarly:

$$N = \{n \in B, u(n) \neq parking\} \quad (15)$$

Evidently: $B = P \cup N$

Technically, all buildings in N may be used as negative examples in the current context. However, in order to avoid problems stemming from imbalanced datasets we have chosen to use for the experiments the majority of the members of P as the positive class representatives and we have created different sample data sets of randomly selected non-parking buildings as the negative class representatives. The positive examples correspond to the real parking areas scattered in the vicinity of our area of interest. The recorded real parkings in our dataset are 1000. As a contradiction, the other 1000 buildings, which will constitute an example of negative class, have been randomly selected ensuring obviously they do not belong to the first class. The use of real data gives us the possibility to clearly and directly verify the outcome. We have created 8 different negative datasets, the choice of the negative examples being random. The data that will serve as negative examples consists of every other buildings in the dataset since we cannot exclude any of them, due to the lack of a computational model to decide on its suitability.

In particular, for the positive examples and according to the notation above, we have:

$$P_e \subset P, P_e = \{p_1, p_2, \dots, p_{1000}\} \quad (16)$$

whereas, for the negative examples:

$$N_e^i \subset N, N_e^i = \{n_1^i, n_2^i, \dots, n_{1000}^i\} \quad (17)$$

$$\bigcap_{i=1}^{8} N_e^i = \emptyset$$

In order for these datasets to be used in the training and evaluation, each participating building has to be represented by a feature vector. Each element of these vectors represents a metric contributing to the aforementioned processes, whereas the value contained in the feature vector of a building represents the assessment of the real-world data of the building against this metric. In the general case, each feature value is a real number, hence, we may consider a function mapping the building properties, as expressed in the database, to an n-dimensional feature vector:

$$F: B \mapsto \mathbb{R}^n$$

In practice, each feature vector consists of the following features:

- Number of other parking places in the area (1000m)
- Distance to the nearest next parking (Euclidean)
- Number of ATMs at a distance of 1000m
- Distance from the nearest ATM (Euclidean)
- Distance from the nearest ATM (Dijkstra using the actual routable road network)
- Number of spots of tourist interest (1000m)
- Distance to the nearest spot of tourist interest
- Building area (in m2)

As an output of the experiments, a binary classification is desired between 'parking' and 'no parking' classes. Where the implementations allow it, the same word classes have been used, while in the rest of occasions, where the output has to be scalar, to maintain uniformity 1 corresponds to the 'parking' class and 0 to 'no parking' class. In the following, and in compliance with the above notation, we will represent by P_c the buildings corresponding to the set of samples predicted as positive by the classifier, and by N_c the buildings corresponding to the set of samples predicted as negative by the classifier.

12.1.2. Experimental setup

The lack of a mathematical model to coherently describe the discussed urban data and its characteristics as well as the complexity of the problem make the selection of an appropriate classifier to automatically and successfully predict the suitability of urban locations for particular uses a challenging task.

To ensure a sound and solid experimental evaluation, the tests performed should be expanded and replicated in multiple sample data sets as described previously. To that end we created 8 different sample data sets, each containing features of the 1000 positive examples, i.e. the actual parking areas, which are the same in all sample data sets, and features of 1000 negative examples, i.e. randomly selected buildings, which are different in each sample data set. For each sample data set, two subclasses of experiments have been created: the former uses

the entire data and randomly chooses the sections used for training, validation and testing using a ratio of 85%, 5% and 10% respectively, while in the other, we masked a segment of data completely from the classifier, only to present it as input after the phase of training, for testing.

In the following presentation of the assessment of the results, we have adopted the following terms:

True Positives (TP): The cases in which the classifier predicted yes and the actual sample's class was also yes, formally $TP = P_c \cap P_e$

True Negatives (TN): The cases in which the classifier predicted no and the actual sample's class was no, formally $TN = N_c \cap N_e^i$

False Positives (FP): The cases in which the classifier predicted yes and the actual sample's class was no, formally $FP = P_c \cap N_e^i$

False Negatives (FN) : The cases in which the classifier predicted no and the actual sample's class was yes, formally $FN = N_c \cap P_e$

The above are summarized in the following table (Table 2)

		Actual Class	
		YES (P_e)	NO (N_e^i)
Classifier's Prediction	YES (P_c)	TP	FP
	NO (N_c)	FN	TN

Table 2 –Summary of prediction results

For the evaluation of the results, the following metrics are used:

- Accuracy: the ratio of the number of correct predictions to the total number of input samples number of input samples

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions\ made} = \frac{|TP| + |TN|}{|P_c| + |N_c|}$$

- Specificity: It corresponds to the proportion of negative samples that are

mistakenly considered as positive, with respect to all negative samples.

$$Specificity = \frac{False\ Positive}{False\ Positive + True\ Negative} = \frac{|FP|}{|N_e^i|}$$

- Precision: It is the number of correctly predicted positive results divided by the number of all samples predicted as positive by the classifier.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{|TP|}{|P_c|}$$

- Recall (or Sensitivity): It is the number of correctly predicted positive results divided by the number of all positive samples regardless of prediction (all samples that should have been identified as positive)

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{|TP|}{|P_e|}$$

- F1 measure: is the Harmonic Mean between Precision and Recall. Its range is [0, 1]. It provides information on how precise the classifier is (how many instances it classifies correctly), as well as how robust it is (if it misses a significant number of instances).

$$F1 = 2\frac{1}{\frac{1}{Precision} + \frac{1}{Recall}}$$

- Gmean: The geometric mean (G-mean) is the root of the product of class-wise sensitivity. This measure tries to maximize the accuracy on each of the classes while keeping these accuracies balanced. For binary classification G-mean is the squared root of the product of the sensitivity and specificity.

$$G\ Mean = \sqrt{Sensitivity \cdot Specificity}$$

12.1.3. Experimental results

Given the fact that the problem of predicting the appropriateness of city locations for specific uses using real-world urban data and machine learning has not, to our knowledge, been studied before in the literature, it was deemed useful and necessary to conduct a detailed experimentation process considering a variety of machine learning classification methods, the

results of which are compared and discussed. The examined classifiers include: feedforward neural networks (multilayer perceptrons), Support Vector Machines, bag of decision trees and random forests, k-Nearest Neighbors and Naïve Bayes. In the subsections that follow, a detailed presentation of the experimental results for each method is provided, followed by a comparative analysis.

For the classifiers, we also need to assess the margin for augmenting their performance through optimization of their parameters. The results we quote are the ones subsequent to the several stages of optimization. For most we followed a technique often used in machine learning lately, called Bayesian Optimization. In machine learning problems dealing with several hyperparameters the process to tune the classifier usually frequently involves costly plentiful costly evaluations both in computational resources and time. To avoid that we could use Bayesian Optimization to optimize the parameters. We can build a probabilistic model for the objective and compute the posterior predictive distribution integrating all the possible true functions, thus leading to optimizing a cheap proxy function instead whose model is much cheaper than the true objective. The main insight of the idea is to make the proxy function exploit uncertainty to balance exploration against exploitation. However this solution is not universal, nor yielding best results in all occasions, especially in neural networks [96], where manual optimization was applied.

12.1.4. Multilayer perceptron results

The first set of experiments involves the optimization of neural networks, i.e. multilayer perceptrons (MLP), specifically the network architecture, and the multitude of hidden layers as well as neurons per layer. We will initially test two-layer architectures by keeping the number of neurons in the 2nd layer stable and perform testing for the number of neurons in the first layer (Figure 47), since it is often argued that problems rarely need to use over 2 hidden layers of neurons. We conclude that minimal error occurs for 70 neurons. Performing tests respectively for the second layer differentiate results to a negligible degree, so the optimal solution is to keep 15 neurons in the 2nd layer, which is a good trade-off of complexity over results and time. Tests were also performed with single layer perceptron but did not produce near as promising results. As a metric of performance the Mean Square Error (MSE) of misclassification was used.

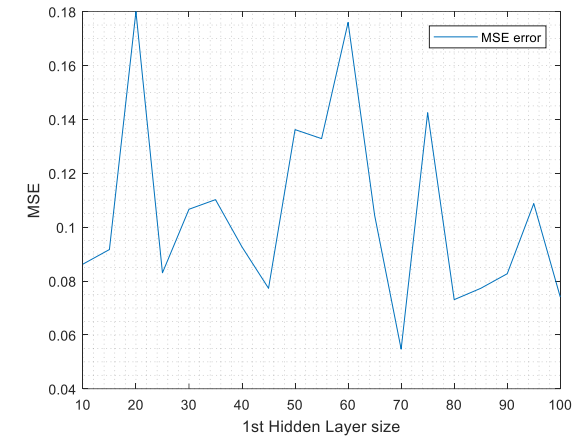


Figure 47 - Architecture and graph of Mean Square Error (MSE) plot as varied for 1st layer number of neurons

Subsequently the configuration providing the best result was chosen for the continuation of experiments and analysis of behavior in unknown inputs where the first results were not encouraging, since there were data pockets for which the Mean Square Error of misclassification ranged from 32-36% which differs greatly from network performance in the previous experiment as shown in Table 3. There are numerous other parameters in the architecture of the network (hidden neuron activation functions, seasons, learning rate etc) we customized and performed further testing, but we still encountered networks' usual problems of local minima and overfitting that prevented achieving better results.

Dataset	Accuracy	Specificity	Precision	Recall	F_measure	Gmean
1	0,790	0,84	0,82	0,74	0,78	0,79
2	0,740	0,92	0,88	0,56	0,68	0,72
3	0,660	0,72	0,68	0,60	0,64	0,66
4	0,640	0,67	0,65	0,61	0,63	0,64
5	0,640	0,68	0,65	0,60	0,63	0,64
6	0,630	0,78	0,69	0,48	0,56	0,61
7	0,675	0,65	0,67	0,70	0,68	0,67

8	0,675	0,78	0,72	0,57	0,64	0,67

Table 3 - Results for Artificial Neural Networks

12.1.5. SVM results

The initial tests were carried out using the default radial basis function (5). For the evaluation of the results, in addition to the MSE used previously, SVM supports the kfoldloss metric that gives more valid results as it does not randomly select a percentage of the inputs to hide it and use it for control but separates inputs into random parts of a particular size and tests them all at the testing stage. So all the data, at some time, will be used as both training input and testing input.

Using MSE to evaluate the results, the error ranges at very low levels (around 7-8%), but with the stricter and more accurate kfoldloss the misclassification varies in the range of 18-21%. In combination with our observation from last experiment, we tend to assume the SVM is a better suited classifier for our case than MLP.

Proceeding with the next stage of experiment we intentionally hide some input data and ask the SVM to classify the hidden inputs. The results present approximately the same error percentages as kfoldloss metric.

Interestingly, in separate experiments on positive and negative inputs (that is, if we only give vectors of parking areas as input and then only non parking we observe that it has a very high efficiency (> 92%) predicting correctly the negative class, that is, we can present it to building and can say with great efficiency that it is not a parking.

Deriving from the latter deduction it became prominently perceptible that it could prove propitious examining alternate configurations in the SVM architecture (kernels and their hyperparameters), using the aforementioned technique Bayesian Optimization.

The hyperparameters we try to optimize in this case, are box and sigma (Figure 48). Box refers to the slack variable which controls the error margin we allow the classifier to undergo. Even though we want our classifier to make no mistakes at all and find a hyperplane that separates all positive and negative examples without exemption, sometimes it is preferable to allow some mistakes, because absolutely strict separation can lead to poorly fit models. In the aforementioned cases of complex non-linearly separable real word problems some examples can also be mislabeled or extremely unusual (noise in the data). In order to achieve a better overall

solution or a solution at all in other cases, we must allow some misclassified points, and our goal is altered to discovering the solution containing the least mistakes.

By using the optimized parameters in the unknown data, we perceive a much better

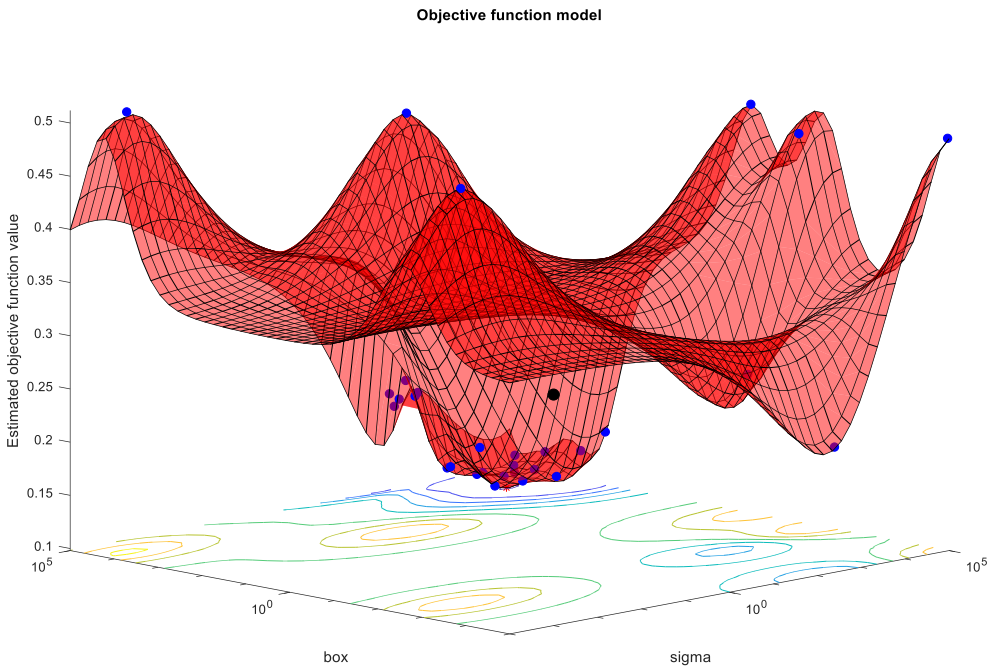


Figure 48 - Optimization of hyperparameters box and sigma

performance, in random samples the MSE is limited to 8-9% while the kfoldloss (for every possible combination of the new hidden inputs) varies about 16-18% (Table 4).

Dataset	Accuracy	Specificity	Precision	Recall	F_measure	Gmean
1	0,85	0,91	0,90	0,78	0,83	0,84
2	0,85	0,90	0,89	0,80	0,84	0,85
3	0,77	0,81	0,79	0,73	0,76	0,77
4	0,77	0,80	0,78	0,73	0,76	0,76
5	0,78	0,87	0,84	0,69	0,76	0,77
6	0,80	0,90	0,88	0,70	0,78	0,79
7	0,78	0,83	0,81	0,73	0,77	0,78
8	0,80	0,90	0,88	0,70	0,78	0,79

Table 4 - Results for SVMs

12.1.6. Bag of Decision Trees & Random Forests

Tree-based methods partition the feature space into a set of rectangles, and then fit a simple model in each one. Plain decision trees (either classification or regression) present drawbacks and limitations: They have a very high tendency to over-fit the data, small variations in the data might result in a completely different tree providing us with unstable results, their implementations are based on heuristic algorithms such as the greedy algorithm where locally

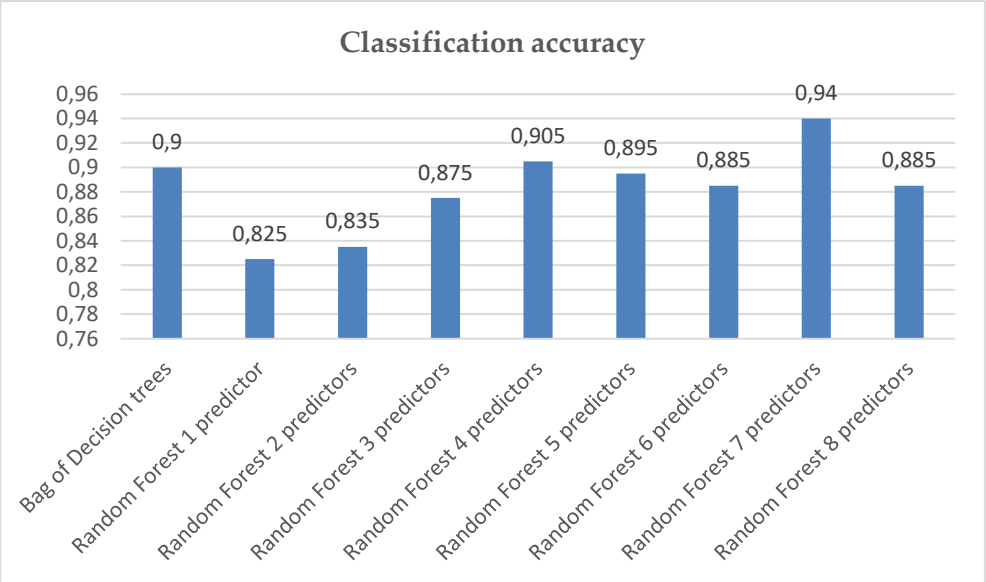


Table 5 - Behavior of Random Forest classifier with different number of features

optimal decisions are made at each node. Such algorithms cannot guarantee to return the globally optimal decision tree.

Therefore several other strategies have been adopted to confront the above such as pruning and bootstrap aggregation (bagging). In test conducted, it proved to be a very promising classifier with a misclassification error that does not exceed 8-10% (Table 5).

Those results motivated the employment of the random forests classifier to our data which provides an improvement over bagged trees by way of a random small tweak that decorrelates the trees. Unlike bagging, in the case of random forests, as each tree is constructed, only a random sample of predictors is taken before each node is split. Since at the core, random forests too are bagged trees, they lead to reduction in variance. Experimenting with the number of predictors, we advance to even better results and classification success at times reaches 96% (Table 6).

Dataset	Accuracy	Specificity	Precision	Recall	F_measure	Gmean
1	0,96	0,98	0,98	0,93	0,95	0,95
2	0,94	0,98	0,98	0,89	0,93	0,93
3	0,86	0,93	0,92	0,79	0,85	0,86
4	0,92	0,93	0,93	0,91	0,92	0,92
5	0,93	0,92	0,92	0,94	0,93	0,93
6	0,93	0,92	0,92	0,93	0,93	0,92
7	0,92	0,92	0,92	0,92	0,92	0,92
8	0,86	0,92	0,91	0,80	0,85	0,86

Table 6 - Results for optimized Random Forests

12.1.7. KNN (K nearest neighbors)

K nearest neighbors algorithm may be one of the simplest classification algorithms but produces generally competitive results. It is non-parametric which means it does not presume in advance any models for the data distribution and its explicit training phase is minimal, making it very fast. However this also means that most data is needed during the test phase so altering the training data may conclude to substantial variations of the results or poor classification models. The mean error from all the data sets used ranges mostly around 30% (Table 6).

Dataset	Accuracy	Specificity	Precision	Recall	F_measure	Gmean
1	0,73	0,92	0,87	0,53	0,66	0,70
2	0,71	0,94	0,89	0,48	0,62	0,67
3	0,72	0,90	0,84	0,53	0,65	0,69
4	0,69	0,92	0,85	0,46	0,60	0,65
5	0,71	0,92	0,86	0,50	0,63	0,68
6	0,68	0,91	0,83	0,44	0,58	0,63
7	0,68	0,93	0,86	0,43	0,57	0,63
8	0,69	0,87	0,80	0,51	0,62	0,67

Table 7 - Results of KNN

12.1.8. Naive Bayes

Naïve Bayes method is based on the Bayes' theorem that provides a way to calculate posterior probability. It assigns the most likely class to a given example described by its feature vector, based on the naïve assumption that features are independent given class, that is, $P(X|C) = \prod_{i=1}^{n} P(X_i|C)$ where $X = (X_1, \dots, X_{n)}$ is a feature vector and C is a class. The predictors are assumed conditionally independent. Even though this assumption is unrealistic and often violated in practice [97], the classifier is remarkably successful and tends to yield results competitive to far more sophisticated techniques.

It is a fast algorithm performing well in multiclass scenarios, but in our dataset test case its drawbacks prevailed and it presented poor performance compared to other classifiers and the error ranges from 25-27% in the hidden data test (Table 8).

Dataset	Accuracy	Specificity	Precision	Recall	F_measure	Gmean
1	0,77	0,85	0,82	0,68	0,74	0,76
2	0,74	0,85	0,81	0,63	0,71	0,73
3	0,73	0,78	0,76	0,68	0,72	0,73
4	0,73	0,78	0,76	0,68	0,72	0,73
5	0,73	0,78	0,76	0,68	0,72	0,73
6	0,73	0,78	0,76	0,68	0,72	0,73
7	0,73	0,78	0,76	0,68	0,72	0,73
8	0,73	0,78	0,76	0,68	0,72	0,73

Table 8 - Results of Naïve Bayes

12.2. Comparisons - First Conclusions

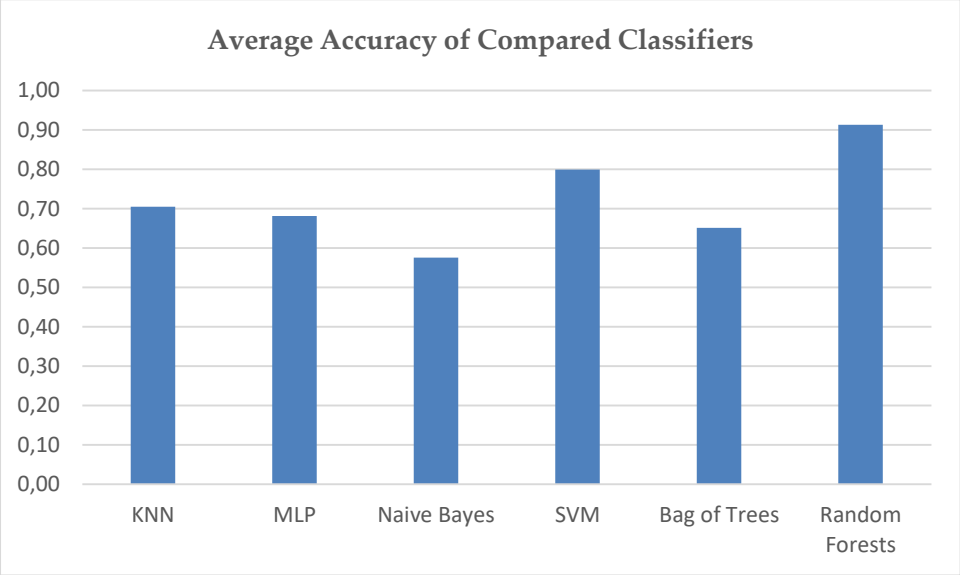
The results of the comprehensive comparisons can be aggregated in the following table and graphs. In Table 9 the average of the metrics applied for all datasets of each classifier after its optimization is presented.

Classifier	Accuracy	Specificity	Precision	Recall	F_measure	Gmean
MLP	0,681	0,755	0,719	0,608	0,655	0,674

SVM	0,799	0,865	0,846	0,733	0,784	0,796
KNN	0,699	0,914	0,850	0,485	0,617	0,665
Naive Bayes	0,736	0,798	0,770	0,674	0,718	0,733
Bag of Decision trees	0,651	0,601	0,636	0,700	0,666	0,649
Random Forest	0,913	0,938	0,934	0,889	0,910	0,912

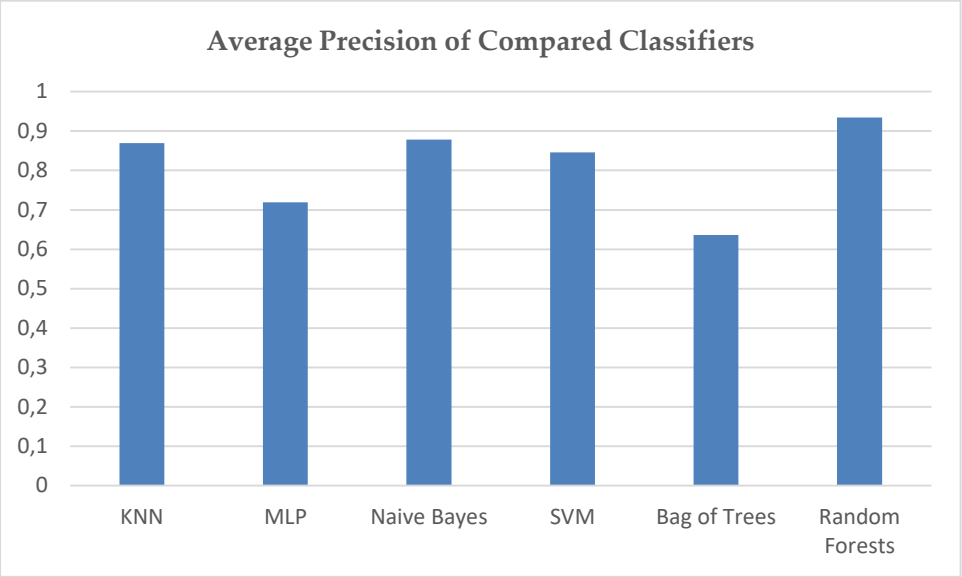
Table 9 - Average metrics for all datasets of all classifiers

In Graph 1 the results of accuracy of compared classifiers is presented. We observe that



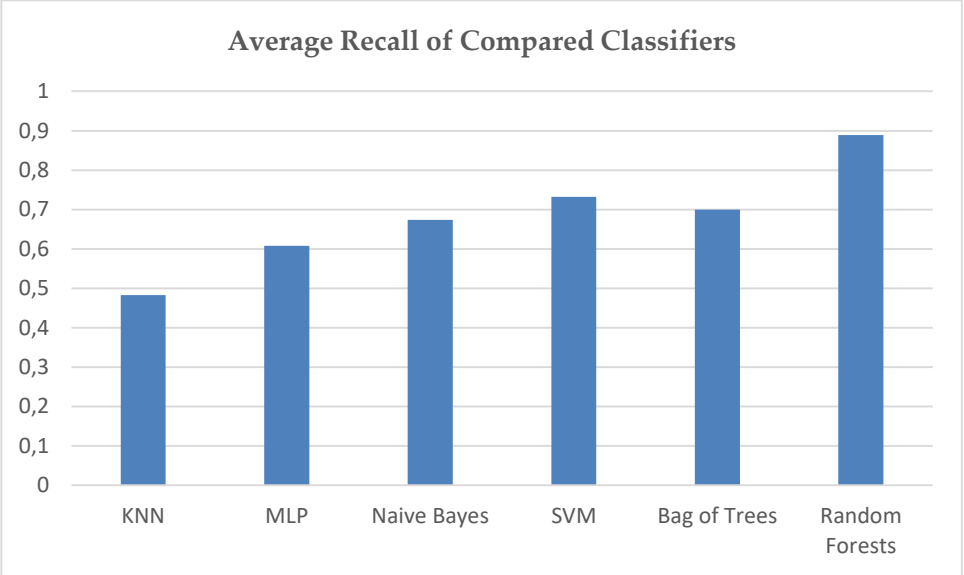
Graph 1 - Comparison of Accuracy of all Classifiers

the use of machine learning techniques we propose, with proper parameterization and training can help finding or locating solutions to urban planning problems. Our reasoning that the large volume and diversity of data combined with the absence of a computational model is an ideal field of action for the aforementioned techniques was confirmed. In Graph 2 we present the average precision of all classifiers,



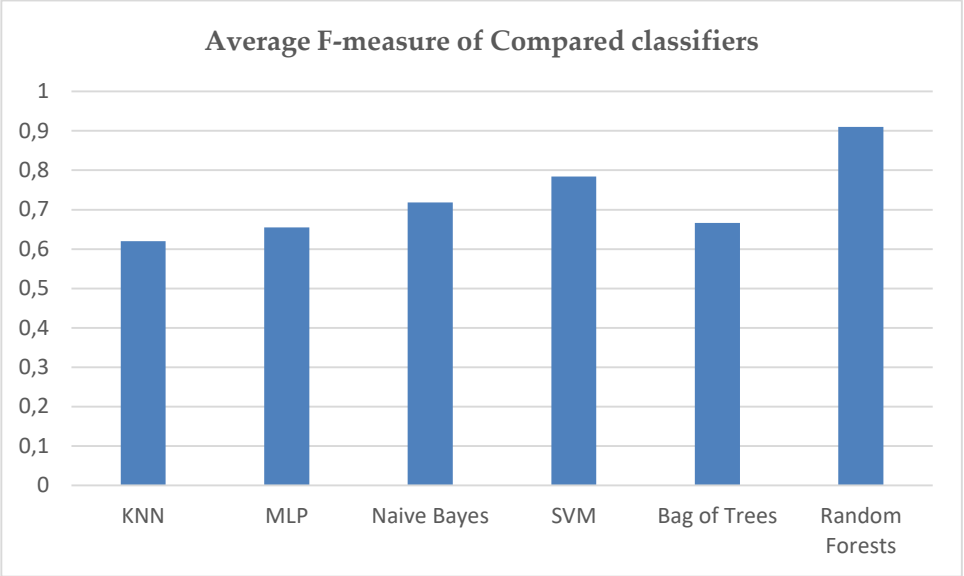
Graph 2 - Comparison of Precision of all Classifiers

while in Graph 3 the recall is presented,



Graph 3 - Comparison of Recall of all Classifiers

and finally in Graph 4 the F-measure



Graph 4 – Comparison of F-measure of all Classifiers

13. Discussion and Conclusions

13.1. Discussion – Conclusions

In this work, we presented a visual semantic decision support system that can be used in the context of urban planning applications. The system fuses and merges various types of data modalities from different sources of urban data, encodes them using a semantic model that can capture and utilize both low-level geometric information and higher level semantic information and subsequently uses a machine learning model based on random forests to estimate the suitability of different city spaces for specific urban uses. The proposed methods have been validated on real-world data and compared with a wide range of machine learning techniques, and the evaluation indicates promising results.

The results of the enrichment of the system with machine learning techniques show that there is room for development of relevant research. The enormous amount of data, the speed at which they are acquired and gathered, and the ever-increasing spread of all sensors installed in the city combined with the advent of IoT, enact it as an important part of all the systems that manage or relate to urban data.

Several classifiers (Naïve Bayes, K Nearest Neigbhors, Artificial Neural Networks, SVMs, Bag of Trees, Random Forests, as well as optimized models of these specifically for the problem data sets were tested), with the dominant one demonstrating accuracy scores over 91% in all data sets, whereas others displaying also very satisfactory results scores (about 86%), which can be characterized as promising.

13.2. Future work

As future work directions, we intend to acquire and add more urban data, such as traffic data for more realistic distance calculations, and parking revenue data and parking traffic to examine that aspect as well. In addition, we are exploring potential ways of transforming and preparing the data to feed it into convolutional neural network and deep learning and evaluate and compare results. Finally, we aim at replicating the experiment for other cities, provided that we can acquire similar real-world data

13.3. Summary

The first step is that of data import, following their collection and availability to the system. During their import, the data are appropriately converted to comply with the rules of the

geodatabase and the semantic model applied. Relevant checks are also performed at this step, for conversion errors or incomplete data.

In the next step, the data are checked at geospatial level, ensuring that, despite being collected from diverse sources, they are eventually located in the same projection coordinate system, in order to use the same metric system and execute uniform geometrical and geographical calculations.

The third step involves the extraction of semantic features from the data. For example, metadata, use of spaces and buildings, parks and other green areas, public transport stations, where available, are detected and interpreted by our system to populate the ontological model in the geodatabase.

The next stage is data pre-processing and execution of computations for each research question we endeavor to implement, while visualization is performed. For the parking query examined herein, parking spaces and corresponding random buildings are selected, geometric and realistic road distances are calculated, and data is enriched with ontological features extracted from the calculations (e.g. adjacency to specific spaces/buildings based on criteria).

To conclude, the aforementioned data have been imported to our system, combined with corresponding data originating from other open data providers (e.g. the road routable network), undergone the appropriate manipulation to extract the semantic information the framework embassies, while simultaneously calculating the features to be used in the fore coming experiments.

14. References

[1] E. Verbree and P. J. M. Van Oosterom, "The STIN method: 3D surface reconstruction by observation lines and Delaunay TENs," in *Proceedings of ISPRS Workshop on 3D-reconstruction from airborne laserscanner and InSAR data, Dresden, Germany*, 2003.

[2] A. A. Alesheikh, H. Helali, and H. A. Behroz, "Web GIS: technologies and its applications," in *Symposium on geospatial theory, processing and applications*, 2002, vol. 15.

[3] Q. Zhu *et al.*, "Towards semantic 3D city modeling and visual explorations," in *Advances in 3D Geo-Information Sciences*, Springer, 2011, pp. 275–294.

[4] G. Gröger and L. Plümer, "CityGML–Interoperable semantic 3D city models," *ISPRS J. Photogramm. Remote Sens.*, vol. 71, pp. 12–33, 2012.

[5] J. Döllner, K. Baumann, and H. Buchholz, *Virtual 3D city models as foundation of complex urban information spaces*. na, 2006.

[6] T. H. Kolbe, "Representing and exchanging 3D city models with CityGML," in *3D geo-information sciences*, Springer, 2009, pp. 15–31.

[7] E. Dimopoulou, D. Kitsakis, and E. Tsiliakou, "Investigating correlation between legal and physical property: possibilities and constraints," in *Third International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2015)*, 2015, vol. 9535, p. 95350A.

[8] S. P. Singh, K. Jain, and V. R. Mandla, "Virtual 3D city modeling: techniques and applications," *ISPRS-Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, no. 2, pp. 73–91, 2013.

[9] T. de Vries and S. Zlatanova, "3D intelligent cities," *GeoInformatics*, vol. 14, no. 3, p. 6, 2011.

[10] T. H. Kolbe, G. Gröger, and L. Plümer, "CityGML: Interoperable access to 3D city models," in *Geo-information for disaster management*, Springer, 2005, pp. 883–899.

[11] G. Gröger, T. Kolbe, and A. Czerwinski, "Candidate OpenGIS CityGML Implementation Specification (City Geography Markup Language)," *Open Geospatial Consort. Inc OGC*, 2007.

[12] B. Mao, Y. Ban, and L. Harrie, "A multiple representation data structure for dynamic visualisation of generalised 3D city models," *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 2, pp. 198–208, 2011.

[13] A. Gospodini, "Portraying, classifying and understanding the emerging landscapes in the post-industrial city," *Cities*, vol. 23, no. 5, pp. 311–330, 2006.

[14] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, "Smart cities and the future internet: Towards cooperation frameworks for open innovation," in *The future internet assembly*, 2011, pp. 431–446.

[15] H. Chourabi *et al.*, "Understanding smart cities: An integrative framework," in *2012 45th Hawaii international conference on system sciences*, 2012, pp. 2289–2297.

[16] N. Karacapilidis and D. Papadias, "Computer supported argumentation and collaborative decision making: the HERMES system," *Inf. Syst.*, vol. 26, no. 4, pp. 259–277, 2001.

[17] J. Van der Bent, J. Paauwe, and R. Williams, "Organizational learning: an exploration of organizational memory and its role in organizational change processes," *J. Organ. Change Manag.*, vol. 12, no. 5, pp. 377–404, 1999.

[18] R. H. Bonczek, C. W. Holsapple, and A. B. Whinston, *Foundations of decision support systems*. Academic Press, 2014.

[19] J. P. Shim, M. Warkentin, J. F. Courtney, D. J. Power, R. Sharda, and C. Carlsson, "Past, present, and future of decision support technology," *Decis. Support Syst.*, vol. 33, no. 2, pp. 111–126, 2002.

[20] M. S. Scott-Morton and P. G. Keen, *Decision support systems: an organizational perspective*. Addison-Wesley, Reading, MA, 1978.

[21] J. E. Aronson, T.-P. Liang, and E. Turban, *Decision support systems and intelligent systems*, vol. 4. Pearson Prentice-Hall New York, 2005.

[22] A. Lang Golub, *Decision analysis: an integrated approach*. Wiley, 1997.

[23] H. A. Simon, "The new science of management decision.," 1960.

[24] R. H. Sprague Jr and E. D. Carlson, *Building effective decision support systems*. Prentice Hall Professional Technical Reference, 1982.

[25] C. Levinson Stephen, "Presumptive meanings. The theory of generalized conversational

implicature," *Camb. Mass. Inst. Technol.*, 2000.
[26] N. Chomsky and D. W. Lightfoot, *Syntactic structures*. Walter de Gruyter, 2002.
[27] T. Gruber, "Ontology," in *Encyclopedia of Database Systems*, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, pp. 1963–1965.
[28] W. Kuhn, "Geospatial Semantics: Why, of What, and How?," in *Journal on Data Semantics III*, 2005, pp. 1–24.
[29] K. Janowicz, S. Scheider, T. Pehle, and G. Hart, "Geospatial semantics and linked spatiotemporal data–Past, present, and future," *Semantic Web*, vol. 3, no. 4, pp. 321–332, 2012.
[30] K. Janowicz, M. Raubal, and W. Kuhn, "The semantics of similarity in geographic information retrieval," *J. Spat. Inf. Sci.*, vol. 2011, no. 2, pp. 29–57, May 2011.
[31] O. Ahlqvist and A. Shortridge, "Characterizing Land Cover Structure with Semantic Variograms," in *Progress in Spatial Data Handling: 12th International Symposium on Spatial Data Handling*, A. Riedl, W. Kainz, and G. A. Elmes, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 401–415.
[32] F. Probst and M. Lutz, "Giving meaning to GI web service descriptions," *Ontol.-Based Discov. Compos. Geogr. Inf. Serv.*, p. 206, 2004.
[33] NASA, "A.40 computational modeling algorithms and cyberinfrastructure," Jan. 2012.
[34] C. Schlieder, "Digital heritage: Semantic challenges of long-term preservation," *Semantic Web*, vol. 1, no. 1,2, pp. 143–147, Jan. 2010.
[35] W. Kuhn, "Modeling vs encoding for the Semantic Web," *Semantic Web*, vol. 1, no. 1,2, pp. 11–15, Jan. 2010.
[36] A. U. Frank, "Chapter 2: Ontology for Spatio-temporal Databases," in *Spatio-Temporal Databases: The CHOROCHRONOS Approach*, T. K. Sellis, M. Koubarakis, A. Frank, S. Grumbach, R. H. Güting, C. Jensen, N. A. Lorentzos, Y. Manolopoulos, E. Nardelli, B. Pernici, B. Theodoulidis, N. Tryfona, H.-J. Schek, and M. O. Scholl, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 9–77.
[37] T. Bittner, M. Donnelly, and B. Smith, "A spatio-temporal ontology for geographic information integration," *Int. J. Geogr. Inf. Sci.*, vol. 23, no. 6, pp. 765–798, 2009.
[38] A. G. Cohn and S. M. Hazarika, "Qualitative Spatial Representation and Reasoning: An Overview," *Fundam. Informaticae*, vol. 46, no. 1–2, pp. 1–29, Jan. 2001.
[39] C. B. Jones, A. I. Abdelmoty, D. Finch, G. Fu, and S. Vaid, "The SPIRIT Spatial Search Engine: Architecture, Ontologies and Spatial Indexing," in *Geographic Information Science*, 2004, pp. 125–139.
[40] N. Chrisman, "Exploring geographic information systems," Wiley, 1997.
[41] F. Harvey, W. Kuhn, H. Pundt, Y. Bishr, and C. Riedemann, "Semantic interoperability: A central issue for sharing geographic information," *Ann. Reg. Sci.*, vol. 33, no. 2, pp. 213–232, May 1999.
[42] A. Gangemi and P. Mika, "Understanding the Semantic Web through Descriptions and Situations," in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, 2003, pp. 689–706.
[43] K. A. Nedas and M. J. Egenhofer, "Spatial-Scene Similarity Queries," *Trans. GIS*, vol. 12, no. 6, pp. 661–681, 2008.
[44] S. Borgo, N. Guarino, and L. Vieu, "Formal ontology for semanticists," *Res. Inst. Comput. Sci. Toulouse–CNRS Lab. Appl. Ontol. Www Loa-Cnr It*, 2005.
[45] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowl. Acquis.*, vol. 5, no. 2, pp. 199–220, 1993.
[46] N. Drummond and M. Horridge, "A practical introduction to ontologies & OWL," *Retreive Httpwww Co-Ode Orgresourcestutorialsintro Univ. Manch.*, 2005.
[47] N. Guarino, "Semantic matching: Formal ontological distinctions for information organization, extraction, and integration," in *International Summer School on Information Extraction*, 1997, pp. 139–170.
[48] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari, "Wonderweb deliverable d17," *Comput. Sci. Prepr. Arch.*, vol. 2002, no. 11, pp. 74–110, 2002.
[49] N. R. Chrisman, "Beyond Stevens: A revised approach to measurement for geographic information," in *AUTOCARTO-CONFERENCE-*, 1995, pp. 271–280.

[50] D. Martin *et al.*, “OWL-S: Semantic markup for web services,” *W3C Memb. Submiss.*, vol. 22, no. 4, 2004.
[51] P. Mika, “Social Networks and the Semantic Web,” in *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, Washington, DC, USA, 2004, pp. 285–291.
[52] E. Bottou and V. Vapnik, “Local Learning Algorithms,” *Neural Comput.*, vol. 4, pp. 888–900, 1992.
[53] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer Science & Business Media, 2013.
[54] L. Breiman, “Random Forests,” *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001.
[55] A. Liaw and M. Wiener, “Classification and regression by randomForest,” *R News*, vol. 2, no. 3, pp. 18–22, 2002.
[56] P. Vincke, *Multicriteria decision-aid*. John Wiley & Sons, 1992.
[57] M. R. Patel, M. P. Vashi, and B. V. Bhatt, “SMART-Multi-criteria decision-making technique for use in planning activities.”
[58] L. López and J. Carlos, “Multicriteria decision aid application to a student selection problem,” *Pesqui. Oper.*, vol. 25, no. 1, pp. 45–68, Apr. 2005.
[59] P. Vincke, *Multicriteria decision-aid*. John Wiley & Sons, 1992.
[60] J.-P. Brans and B. Mareschal, “Promethee Methods,” in *Multiple Criteria Decision Analysis: State of the Art Surveys*, J. Figueira, S. Greco, and M. Ehrogott, Eds. New York, NY: Springer New York, 2005, pp. 163–186.
[61] L. Liu, E. A. Silva, C. Wu, and H. Wang, “A machine learning-based method for the large-scale evaluation of the qualities of the urban environment,” *Comput. Environ. Urban Syst.*, vol. 65, pp. 113–125, Sep. 2017.
[62] S. Jiang, A. Alves, F. Rodrigues, J. Ferreira, and F. C. Pereira, “Mining point-of-interest data from social networks for urban land use classification and disaggregation,” *Comput. Environ. Urban Syst.*, vol. 53, pp. 36–46, Sep. 2015.
[63] M. M. Rathore, A. Ahmad, A. Paul, and S. Rho, “Urban planning and building smart cities based on the Internet of Things using Big Data analytics,” *Comput. Netw.*, vol. 101, pp. 63–80, Jun. 2016.
[64] S. Sarkar *et al.*, “Effective Urban Structure Inference from Traffic Flow Dynamics,” *IEEE Trans. Big Data*, vol. 3, no. 2, pp. 181–193, Jun. 2017.
[65] P. Frankhauser, C. Tannier, G. Vuidel, and H. Houot, “An integrated multifractal modelling to urban and regional planning,” *Comput. Environ. Urban Syst.*, vol. 67, pp. 132–146, Jan. 2018.
[66] “Fractalopolis,” 08-Apr-2019. [Online]. Available: https://sourcesup.renater.fr/fractalopolis/. [Accessed: 08-Apr-2019].
[67] B. Streich, “Dynamic Visualization of Urban Sprawl Scenarios,” p. 26.
[68] T. Holzmann, M. Maurer, F. Fraundorfer, and H. Bischof, “Semantically Aware Urban 3D Reconstruction with Plane-Based Regularization,” in *Computer Vision – ECCV 2018*, vol. 11218, V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, Eds. Cham: Springer International Publishing, 2018, pp. 487–503.
[69] A. Zagarskikh, A. Karsakov, and A. Bezgodov, “Efficient Visualization of Urban Simulation Data Using Modern GPUs,” *Procedia Comput. Sci.*, vol. 51, pp. 2928–2932, Jan. 2015.
[70] R. Cabezas, J. Straub, and J. W. Fisher, “Semantically-Aware Aerial Reconstruction From Multi-Modal Data,” presented at the Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 2156–2164.
[71] S. Du, F. Zhang, and X. Zhang, “Semantic classification of urban buildings combining VHR image and GIS data: An improved random forest approach,” *ISPRS J. Photogramm. Remote Sens.*, vol. 105, pp. 107–119, Jul. 2015.
[72] M. Rouhani, F. Lafarge, and P. Alliez, “Semantic segmentation of 3D textured meshes for urban scene analysis,” *ISPRS J. Photogramm. Remote Sens.*, vol. 123, pp. 124–139, Jan. 2017.
[73] T. H. Kolbe, G. Gröger, and L. Plümer, “CityGML: Interoperable Access to 3D City Models,” in *Geo-information for Disaster Management*, P. van Oosterom, S. Zlatanova, and E. M. Fendel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 883–899.
[74] P. D. Smart, J. A. Quinn, and C. B. Jones, “City model enrichment,” *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 2, pp. 223–234, Mar. 2011.

[75] “Specifications | DATEX II.” [Online]. Available: https://datex2.eu/datex2/specifications. [Accessed: 08-Apr-2019].
[76] “European Committee for Standardization.” [Online]. Available: https://www.cen.eu/Pages/default.aspx. [Accessed: 08-Apr-2019].
[77] M. Janssen, Y. Charalabidis, and A. Zuiderwijk, “Benefits, adoption barriers and myths of open data and open government,” *Inf. Syst. Manag.*, vol. 29, no. 4, pp. 258–268, 2012.
[78] M. Janssen, Y. Charalabidis, and A. Zuiderwijk, “Benefits, adoption barriers and myths of open data and open government,” *Inf. Syst. Manag.*, vol. 29, no. 4, pp. 258–268, 2012.
[79] H. Guo, J. Wang, Y. Gao, J. Li, and H. Lu, “Multi-view 3D object retrieval with deep embedding network,” *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5526–5537, 2016.
[80] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[81] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates,” in *2017 IEEE international conference on robotics and automation (ICRA)*, 2017, pp. 3389–3396.
[82] O. Araque, I. Corcuera-Platas, J. F. Sánchez-Rada, and C. Iglesias, “Enhancing Deep Learning Sentiment Analysis with Ensemble Techniques in Social Applications,” *Expert Syst. Appl.*, vol. 77, Feb. 2017.
[83] A. Severyn and A. Moschitti, “Learning to rank short text pairs with convolutional deep neural networks,” in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, 2015, pp. 373–382.
[84] J. Korczak and M. Hemes, “Deep learning for financial time series forecasting in A-Trader system,” in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2017, pp. 905–912.
[85] T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654–669, Oct. 2018.
[86] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle, “Deep learning for computational biology,” *Mol. Syst. Biol.*, vol. 12, no. 7, p. 878, 2016.
[87] R. Schirrmeister, L. Gemein, K. Eggensperger, F. Hutter, and T. Ball, “Deep Learning with Convolutional Neural Networks for Decoding Visualization of EEG Pathology,” *ArXiv Prepr. ArXiv170808012*, 2018.
[88] P. T. Komiske, E. M. Metodiev, and M. D. Schwartz, “Deep learning in color: towards automated quark/gluon jet discrimination,” *J. High Energy Phys.*, vol. 2017, no. 1, p. 110, Jan. 2017.
[89] K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko, and K.-R. Müller, “SchNet – A deep learning architecture for molecules and materials,” *J. Chem. Phys.*, vol. 148, no. 24, p. 241722, Mar. 2018.
[90] S. Zhang, L. Yao, A. Sun, and Y. Tay, “Deep Learning Based Recommender System: A Survey and New Perspectives,” *ACM Comput Surv*, vol. 52, no. 1, pp. 5:1–5:38, Feb. 2019.
[91] S. S. Fainstein and J. DeFilippis, *Readings in Planning Theory*. John Wiley & Sons, 2015.
[92] V. Subramanian, “Looking Ahead,” in *Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node*, V. Subramanian, Ed. Berkeley, CA: Apress, 2019, pp. 529–534.
[93] “Données métropolitaines de Grand Lyon.” [Online]. Available: https://data.beta.grandlyon.com/en/. [Accessed: 12-Jul-2019].
[94] “CesiumJS - Geospatial 3D Mapping and Virtual Globe Platform.” [Online]. Available: https://cesiumjs.org/. [Accessed: 12-Jul-2019].
[95] H. Butler, M. Daly, A. Doyle, S. Gillies, T. Schaub, and C. Schmidt, “The GeoJSON format specification,” *Rapp. Tech.*, vol. 67, 2008.
[96] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian Optimization of Machine Learning Algorithms,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 2951–2959.
[97] I. Rish, “An empirical study of the naive Bayes classifier,” in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, 2001, vol. 3, pp. 41–46.