UNIVERSITÀ
DEGLI STUDI
FIRENZE

Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione (DINFO)
Corso di Dottorato in Ingegneria dell'Informazione

Curriculum: Ingegneria Informatica

Big Data Solutions: Models and Algorithms in Smart City Domains

Candidate
Irene Paoli

Supervisor
Prof. Paolo Nesi

PhD Coordinator
Prof. Fabio Schoen

ciclo XXXII, 2016-2019

Università degli Studi di Firenze, Dipartimento di Ingegneria dell'Informazione (DINFO).

To S.

Acknowledgments

I would like to acknowledge the support, effort and input of my supervisor, Prof. Paolo Nesi. In particular my thanks go to all my colleagues of the DISIT Lab who were of great help during my research. I would like to thank my friends for providing me with unfailing support. Last but not the least, I would like to thank my family for supporting me spiritually through these PhD years. This accomplishment would not have been possible without them. Thank you.

Abstract

The connection, integration and analysis of the information produced by the various forms of data from smart cities, provides a more cohesive and smart understanding of the city that enhances efficiency and sustainability. This rich interconnection of data can be used to better depict, model and predict urban processes and simulate the likely outcomes of future urban development. All these needs express the creation of solutions capable of analyzing heterogeneous kinds of data, providing a multitude of final applications based on the kind of users who requires a certain service, in particular through the use of real-time analytics to manage aspects of how a city functions is regulated.

This thesis's purpose is to study and develop algorithms to improve services to citizens, making the city more knowable and controllable in new, dynamic and interconnected ways. To this aim, Smart City solutions related to urban mobility and environmental monitoring have been studied to improve the quality of citizens' lifestyle, Social Media data have been analyzed to understand the tendency of an information on Twitter to be widespread and to predict the audience of scheduled television programmes.

Contents

.

Chapter 1

Introduction

The *Smart City* concept has received a great attention in the urban development policies areas. Cities are playing a central role as drivers of innovation especially in health, inclusion, business and environmental areas, while Internet technologies are increasingly important for urban development.
This raises the question of how cities, surrounding regions and rural areas can evolve towards innovative, sustainable, open and user-oriented ecosystems, to promote experimentation and accelerate the research cycle, innovation and adoption in real-life environments [161].

The starting point is the definition which states that a city may be called *smart* “when investments in human and social capital and traditional (transport) and modern (ICT) communication infrastructure fuel sustainable economic growth and a high quality of life, with a wise management of natural resources, through participatory government” [43]. This is a holistic definition that takes into account the social and economic needs related to the welfare of citizens, as well as the needs related to urban development, also encompassing peripheral and less developed cities. The challenge in smart cities concerns the urban welfare creation and the quality of life. To this aim, the IoT connected devices (sensors, actuators and other agents) inside Smart Cities are rapidly growing. Thanks to the connection of mobile devices, sensors and actuators, in a Smart City environment a large amount of heterogeneous data are produced, and Big Data approaches are necessary in order to efficiently integrate and manage data, and produce additional knowledge. Thus, the aggregated data can be used and analyzed to produce smart services by generating predictions and suggestions for final users [210].

The connection, integration and analysis of the information produced by the various forms of data from smart cities, provides a more cohesive and smart understanding of the city that enhances efficiency and sustainability [88], [182]. This rich interconnection of data can be used to better depict, model and predict urban processes and simulate the likely outcomes of future urban development [161], [19]. All these needs express the creation of solutions capable of analyzing heterogeneous kinds of data, providing a multitude of final applications based on the kind of user who requires a certain service, in particular through the use of real-time analytics to manage aspects of how a city functions and is regulated.

At the same time, the massive use of social media platforms as new communicative infrastructures and new forms of social connectivity, has made Twitter one of the most important source of information. Since its launch in 2006, Twitter has turned from a niche service to a mass phenomenon by the beginning of 2013. The platform claims to have more than 200 million active users, who "post over 400 million tweets per day". In addition to interpersonal communication, Twitter is increasingly used as a source of real-time information and a place for debate in news, politics, business, and entertainment. The expressiveness and interaction of millions of private users lead to a significant need for the development of innovative methods and prediction models able to deal with such sources of data.

This thesis' purpose is twofold. On the one hand, the aim is to study and develop algorithms to make a city knowable and controllable in new, more fine-grained, dynamic and interconnected ways that "improve[s] the performance and delivery of public services while supporting access and participation" [6]. On the other hand, the aim is to analyze features extracted from Twitter in order to create predictive models.

This thesis describes the PhD research activity carried out at DISIT laboratory (Distributed Data Intelligence and Technology Lab.) of the Department of Information Engineering (DINFO) at the University of Florence. The work of the thesis is divided into two main parts.

Following the introduction presented in Chapter 1 that provides a summary of the background, is **Part I** which presents Smart City solutions related to urban mobility with the aim to improve the quality of citizens' lifestyle. Part I contains four chapters.

Chapter 2 describes a methodology to instrument the city via the placement of Wi-Fi Access Points, and to use them as sensors to capture and understand and predict city users behavior with a significant precision rate. The research work presented in Chapter 2 has led to the publication of the paper *Wi-Fi based city users' behaviour analysis for smart city* in *Journal of Visual Languages and Computing, 2017.* [23] In this research context my contribution has been related to the city user's behavior analysis, in particular the APs clustering and the creation of predictive models for access point connections.

Chapter 3 is focused on presenting the research results regarding a solution to predict the number of available parking slots in city garages with gates comparing three different predictive techniques. The research work presented in Chapter 3 has led to the publication of the paper *Predicting Available Parking Slots on Critical and Regular Services by Exploiting a Range of Open Data* in the *IEEE Access Open Journal, 2018.* [13] In this research context my contribution has been related to the creation and comparison of predictive models, starting from the feature definition, the imputation of missing data and the descriptive statistics, till the model creation and the error measurement definition.

Chapter 4 presents different solutions to understand whether an individual on the move is stationary, walking, on a motorized private or public transport, with the aim of delivering to city users personalized assistance messages for sustainable mobility. The research work presented in Chapter 4 has led to the publication of the paper *Automated Classification of Users' Transportation Modality in Real Conditions* actually under review. In this research context my contribution has been related to the metrics definition, descriptive analysis and the classification models creation and comparison.

Chapter 5 describes a system to carry out automatic real-time statistical data analysis from environmental sensors positioned in Smart Cities and provide services, independently on the number and position of sensors, on a large number of devices in a modality that can be understood by everybody. The research work presented in Chapter 5 has led to the publication of the short paper *Environmental Data Network and Automated Analysis and Representation* in the *Proceedings of the i-Cities National Conference, Pisa, 2019.* In this research context my contribution has been related to the real time air quality data interpolation and the validation procedure, and to the creation of an anomaly detection systems of sensor dysfunctions.

Part II contains two chapters related to Social Media Data analysis:

Chapter 6 is focused on presenting the research results regarding a solution to predict and understand retweet proneness of a post on Twitter (tendency or inclination of a tweet to be retweeted). The research work presented in Chapter 6 has led to the publication of the paper *Assessing the reTweet proneness of tweets: predictive models for retweeting* in the *Multimedia Tools and Applications Journal, 2018.* [135] In this research context my contribution has been related to the twitter metrics definition, descriptive analysis and the predictive models creation and comparison.

Chapter 7 is focused on presenting the research results regarding a solution to predict the audience of scheduled television programmes, where the audience is highly involved such as it occurs with reality shows (i.e., X Factor and Pechino Express, in Italy), exploiting a set of metrics based on Twitter data. The research work presented in Chapter 7 has led to the publication of the paper *Predicting TV programme audience by using twitter based metrics* in the *Multimedia Tools and Applications Journal, 2018.* [55] In this research context my contribution has been related to the twitter metrics definition, descriptive analysis and the predictive models creation and comparison.

After the two main parts presented above, Chapter 8 is a summary of this thesis, including the conclusions.
Concluding this thesis there is the bibliography. Related publications by the author are reported before the introductory Chapter.

The following section presents the main research projects carried out in the DISIT laboratory and the main architectures developed.

1.1 Projects and Frameworks

The research activity is carried out within the projects:

- Sii-Mobility `http://www.sii-mobility.org` for the study of mobility and transport aspects, for the evaluation of service quality and for the study of events.
- RESOLUTE H2020 `http://www.resolute-eu.org` for the resilience aspects, the collection from the related to mobility, the transport system, the flows of people in the city, the risk assessment.
- Snap4City: `http://www.snap4city.org/` for the implementation of an IOT and Big Data management platform.
- TRAFAIR: `http://trafair.eu/` for understanding traffic flows to improve the air quality. One of the main objectives is to establish and predict the air quality levels, such indexes strongly depend on the production of pollution connected to the traffic congestion in the urban areas.

These projects use the model and tools developed by Km4City, a framework created within the DISIT laboratory that provides a single access point for interoperable city data through the web or mobile platforms. Km4City covers aspects of mobility and transport, energy, banking, parking, commerce, culture, cycle paths, green areas, health, tourism and much more. Km4City now processes more than 1 million new data in real time per day, making them accessible in an aggregated way and producing suggestions, trajectories, destination source maps, search responses, predictions, decision support, etc. Within the framework Km4City is placed Twitter Vigilance, the back-bone of the projects Sii-Mobility and RESOLUTE H2020, which allows the interconnection, storage and subsequent querying of heterogeneous data provided by different institutional realities (for example: the portals of the Tuscan region as MIIC, Moving in Tuscany, Observatory of Transports), Open-Data provided by individual municipalities. Twitter Vigilance is a multi-user on-the-edge platform that allows you to do Data-Analytics and Sentiment-Analysis on Social Media Twitter. This means that it is possible to control and analyse the level of appreciation and/or dissent (e.g. of an event, product or people) or to carry out short-term and long-term comparative trend assessments, almost real-time identification of the occurrence of explosive events, critical situations, etc. Twitter Vigilance currently processes data in the order of 140 million lines collected since April 2015. Within the Snap4City project, the Snap4City platform was developed. Snap4City solution provides an exile method to quickly create a large range of smart city applications exploiting heterogeneous data and enabling services for stakeholders by IOT/IOE, data analytics and big data technologies.

1.1.1 Sii-Mobility architecture overview

The reference architecture of Sii-Mobility is depicted in Figure 1.1. The solution allows to collect data coming from different kind of sources (open data, private data, real time data), domains (mobility, environment, energy, culture, e-health, weather, etc.), and protocols. The architecture is based on a semantic aggregation of data and services according to the Km4City ontological model. Data providers as City Operators and Data Brokers offer data which are collected by the Smart city in pull by using Extract Transform and Load (ETL) processes scheduled on the Big Data processing back office based on a Distributed Smart City Engine Scheduler (DISCES) tool developed for Sii-Mobility and made open source. Among the data collected those provided in Open Data from the municipalities, Tuscany region (Observatory of mobility), LAMMA weather agency, ARPAT environmental agency, etc., and several private data coming from City Operators: mobility, energy, health, cultural heritage, services, tourism, wine and food services, education, wellness, etc. Data Brokers collect and manage real time data coming from sensors (IoT), and from vehicular kits (On board Device) which are developed for monitoring and informing car, bus and bike drivers, etc.

Once the data are collected the back office performs several processes for improving data quality, re-conciliating data and converting data into triples for the RDF store of the KB, implemented by using a Virtuoso triple store. DISCES is allocating processes on several virtual machines allocated on the cloud according to their schedule and requests arriving from the Decision Makers, Developers and Data Analytics (typically 3.5-5 thousand of jobs per day, collecting multiple data per job, for example all the busses on a line according to DATEX II protocol. The processes for data collection can be scheduled according to several different policies to cope with Open Data (to verify if they change sporadically), quasi real time data (changing a few times per day) to real time data (changing every few seconds, such as the position of the Bus, or the position of the City Users) and taking into account all the

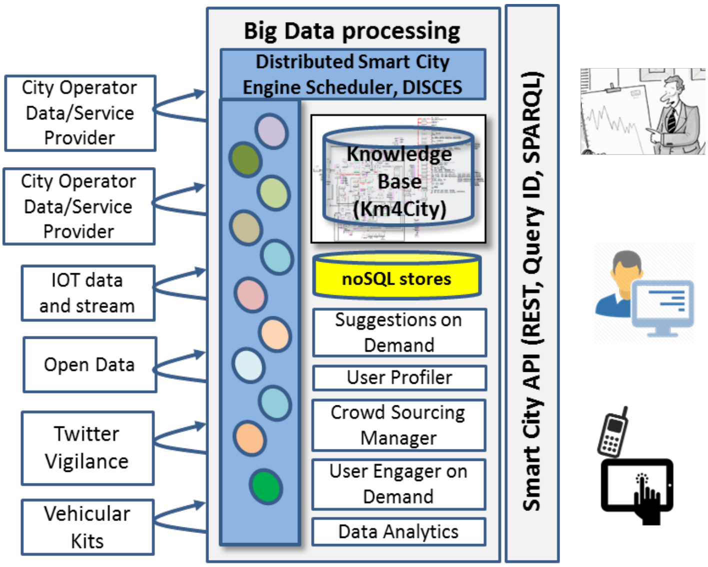


Figure 1.1: Sii-Mobility Architecture.

permissions access connected to each different piece of information managed in the Km4City Knowledge base.

For semantic aggregation of data and service it has been decided to exploit and improve the Km4City Ontology (https://www.km4city.org) [20], [21], adding a number of details regarding mobility and transport, sensors, environment, with respect to former model. Now Km4City is modeling multiple domain aspects related to mobility, services, Wi-Fi, cultural services, energy, structure (streets, civic numbers, green areas, sensors, busses, etc.).

The above collected data is exploited by a number of scheduled data analytics processes to compute: user behavior and mobility, recommendations, suggestions and personal assistant messages according to the city and city operator strategies.

In order to be capable of providing contextual information web and mobile Apps provide data to the Sensor Server and Manager. The data collected from Apps (mainly mobiles) are related to many different aspects: the position of the city users, preferences (user profiles), requests to the Smart City API, searching queries, action performed on mobile, velocity, accelerations, etc. All these kinds of data are useful to understand the user behavior, and thus, to engage the users generating ad-hoc suggestions and recommendations.

In the architecture proposed, in addition to the RDF store for the knowledge base, presents several noSQL stores (namely: HBase and Mongo) for storing tabular data as those arriving from sensors and user profiles, and to make versioning of collected data that have to be passed into the RDF store for reasoning. This approach allows to have the needed tabular data accessible for Data Analytics processes such as those performed for the: estimations of recommendations, engagements, traffic flow predictions, parking forecast, clustering of sensor data behavior, and anomaly detection. When needed, federated queries can be performed among RDF and tabular stores. The resulted architecture provided several services via Smart City API to Development Tools or to the City Users Tools (Applications).

1.1.2 Twitter Vigilance architecture overview

The Twitter Vigilance platform (`http://www.disit.org/tv/`) has been designed and realized by the DISIT Lab of University of Florence as a multipurpose comprehensive tool providing different tasks and metrics suitable for Twitter search API and streams, their monitoring and analysis, for research purpose [12]. The architecture is depicted in 1.2. In Twitter Vigilance, a distributed crawler performs data gathering and extraction by using Twitter Search API. The data acquisition approach is based on the concept of *"Twitter Vigilance Channel"*, consisting in a set of simple and complex search queries which can be defined by a registered user by combining keywords, hashtags, user's IDs, citations, etc., in a structured logical syntax, according to the search syntax of Twitter. The search queries associated with each *Twitter Vigilance Channel* are posed to the Twitter platform via a crawler. Both configuration parameters and statistical results are accessible from the front-end interface for the user. Collected tweets are made accessible to the back-office processes, which implement statistical analysis, natural language processing (NLP) and sentiment analysis (based on distributed NLP on Hadoop [42]), as well as general data indexing. The metrics resulted by the back-office processes are stored on a dedicated database and made accessible to the front-end graphical user interface (see 1.3 as an example), which allows visual analytics, temporal trends and time series visualizations, data results navigation, Twitter users statistics and analysis. All these kinds of analysis are performed at both Twitter Vigilance Channel level and at single search level. In the specific, the following information and metrics can be retrieved: number of tweets and retweets; user citations (to detect potential influencers, pushers, emerging citations, etc.); hashtags (to understand which are the most used, emerging, evolving, etc.); keywords tagged with their part-of-speech (that is, their grammatical function), in terms of nouns, verbs, and adjectives; sentiment analysis; relationships among users; etc.

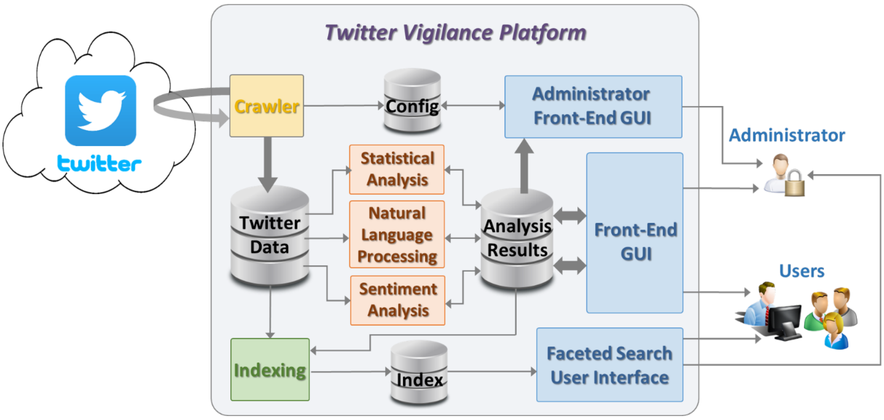


Figure 1.2: Twitter vigilance architecture

The derived metrics and information can be useful to understand which are the most widely used or emerging hashtags, as well to detect which are the most influential in determining the positive/negative signature and polarity detection in the sentiment analysis, and thus for better tuning the tweet collected and for pre-computing basic metrics that can be useful for the researcher to make further analysis in different domains and generically for communication and media, predictive models [24, 25]. It can be a useful tool for identifying reasons for positive/negative tweets, as well as the reaction of the community.

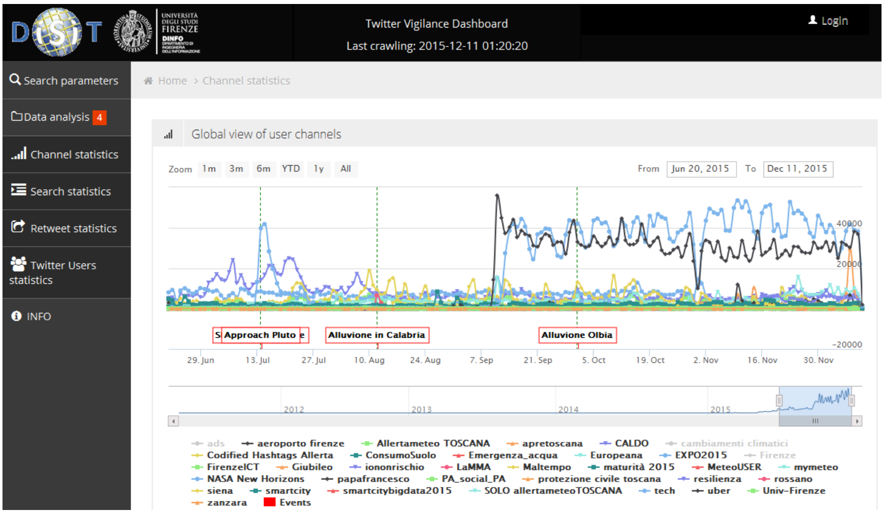


Figure 1.3: Twitter vigilance front-end graphic user interface, showing temporal trends volume based metrics calculated for different user defined channels.

1.1.3 Snap4City architecture overview

The Snap4City solution allows to ingest and manage Big Data coming from IoT devices, applications and services, compute actions for users, for instance providing notifications and a set of visual tools enabling the production of interactive dashboards for data analytics and supporting decision-making processes (useful for many different kinds of users: Public Administrations, final users, developers etc.). Ingested data are collected and aggregated in the Snap4City Knowledge Base, connected to the Km4City multi ontology, and indexed in order to speed up and facilitate data retrieval actions. Snap4City allows also the creation of data-driven applications, based on Micro-services, exploiting mobile and web apps, flows of processing data and IoT data running on the platform [24].

The main tools composing the Snap4City Smart City solution are: data ingestion and aggregation tools, data management tools, data processing tools, APIs system, data indexing system, development tools, tools for final users.

The Data quality has also to be monitored, for example by estimating

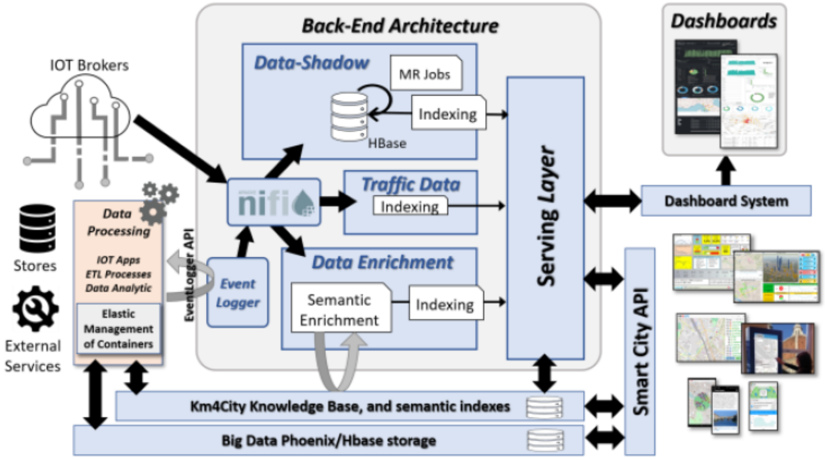


Figure 1.4: Snap4City Indexing Back-End Architecture.

predictions and/or anomaly detection on the basis of historical values, or by defining healthiness criteria, i.e. rules based on data retrieval frequency, non-stationarity, conformity into bounds, etc. In order to treat them uniformly (e.g., in order to perform searches, exploit them in data analytics, visualize and render them with customized dashboards or other visual tools etc.), a semantic regularization process is also needed. To this purpose, a number of ontologies have been proposed [87], [70].

Applications for the final Users (public administrations, citizens but also system managers) are finally provided, in form of visual and interactive tools allowing the creation of Mobile and web Apps, as well as graphic dashboards, thanks to the presence of the Dashboard System [24].

1.2 Related Publications by the author

This research activity has led to several publications in international journals. Many of the results described in this thesis also appear in the articles by the author listed below.[1]

International Journals

1. Bellini, P., Cenni, D., Nesi, P., & **Paoli, I.** (2017). "Wi-Fi based city users' behaviour analysis for smart city". *Journal of Visual Languages and Computing 42*, 31-45.
2. Crisci, A., Grasso, V., Nesi, P., Pantaleo, G., **Paoli, I.**, & Zaza, I. (2018). "Predicting TV programme audience by using twitter based metrics". *Multimedia Tools and Applications*, 77(10), 12203-12232.
3. Nesi, P., Pantaleo, G., **Paoli, I.**, & Zaza, I. (2018). "Assessing the reTweet proneness of tweets: predictive models for retweeting. *Multimedia Tools and Applications*, 77(20), 26371-26396.
4. Badii, C., Nesi, P., & **Paoli, I.** (2018). "Predicting available parking slots on critical and regular services by exploiting a range of open data". *IEEE Access, 6*, 44059-44071.

Submitted

1. Badii, C., Difino, A., Nesi, P., **Paoli, I**., & Paolucci, M. "Automated Classification of Users' Transportation Modality in Real Conditions". Submitted to *Mobile Networks and Applications.*
2. Bellini, P., Nesi, P., **Paoli, I.**, & Soderi, M. "Web of Data Assessment and Characterization through Multilayer Metrics, PCA and Store Clusters". Submitted to *IEEE Access.*
3. Bellini, E., Bellini, P., Nesi, P., Pantaleo, G., **Paoli, I.**, & Paolucci, M. "Big Multimedia Data approach for Smart City Disaster Resilience management". Submitted to *IEEE Access.*

National Conferences

1. Badii, C., Cenni, D., Pantaleo, G., Nesi, P., **Paoli, I.**, & Paolucci, M. (2019). "Environmental Data Network and Automated Analysis and Represenation". Presented at the *5th Italian Conference on ICT for Smart Cities and Communities i-Cities Conference, i-Cities*, Pisa.

[1] The author's bibliometric indices are the following: H-index = 4, total number of citations = 32 (source: Google Scholar on October, 2019).

Part I

Smart City Solutions

Chapter 2

Wi-Fi based city users' behavior and predicting connections

In this chapter is described a methodology to instrument the city via the placement of Wi-Fi Access Points, AP, and to use them as sensors to capture and understand city user behavior with a significant precision rate (the understanding of city user behavior is concertized with the computing of heat-maps, origin destination matrices and predicting user density).[1][2]

2.1 Introduction

The understanding of city users' behavior is one of the most challenging activities in a Smart City context: how the tourists (short, medium and long term) are moving and using the city, how the commuters are arriving and leaving the city, etc. City services are mainly related to mobility, government, energy, culture, events, commercial activities, environment, etc. Among these services, mobility is considered as a commodity; thus, transportation and mobility analyses are valuable aspects always considered for an effective

[1] This chapter has been published as "Wi-Fi based city users' behaviour analysis for smart city" in *Journal of Visual Languages and Computing, 2017.* [23].

[2] *Acknowledgments:* The research work described has been conducted for RESOLUTE project. The authors would like to thank the Municipality of Florence for the collaboration in collecting data, and to the European Commission for funding the RESOLUTE H2020 project Grant Agreement n. 653460 (http://www.resolute-eu.org).

definition of Smart City. According to [79] , Smart Mobility is among the key factors of a modern Smart City, including local and international accessibility, availability of ICT infrastructures, sustainable, innovative and safe transport systems. [42] include traditional transport communication infrastructures among the essential requirements for Smart Cities.

In the context of mobility and transport, traffic/flow analysis is a major prerequisite for planning traffic routing. Hence, it is a central part of the so called Intelligent Transportation Systems (ITS) for public transportation. Traffic flow analysis is commonly used to ease the transportation management, for regulating the access control to the cities, for Smart Parking, for traffic surveillance providing information about road conditions and travel, or for monitoring and controlling the environmental conditions, such as harmful emissions (e.g., CO 2 , PM10, ozone). The European Commission indicates, among the main topics that should be considered with special attention in the framework of the CARS 2020 process, the implementation and promotion of ITS, including Smart Mobility [CARS 2020].

Some of the techniques adopted for traffic monitoring and management can be declined for people flow analysis and thus to support understanding the city user behavior. For the city municipality it is very important to know the movements of city users within a certain precision, and detecting where and how they are crossing the city and exploiting services by using different kinds of transportation solutions: car, bike, walking, taxi, car sharing, buses, tram, etc., targeting services into the city [134], [11], [20].

Usually *Telecom* operators do not provide detailed information about city user behavior: they may provide the number of people connected to each cluster of cellular antennas at a given time slot during the day, but not how the people move actually in the city, passing from one cluster/cell to another. Moreover, the *Telecom* operator collects the cellular traffic from all the city users including residences that are stably at home and thus are not walking and using the city services on the road. *Telecom* operators are also constrained by the national contract from operator to the citizen in term of privacy and data use. At this regard, specific tracking services for mobile devices are needed and, when applied, the citizens have to be informed via an informed consent (e.g., terms of use, privacy policy).

The typical descriptor of people flow analysis in the city is the so called OD matrix (Origin Destination Matrix). The OD matrix presents on both axes the city zones, while the single element (at the intersection) contains the number of people (or the probability) of passing from the zone of origin to the zone of destination, in a given time window, for a given kind of users, for a given day of the week. Therefore, the OD matrix estimation is the main target results to understand the city usage, and thus it is a very relevant data source for traffic/people flow prediction and management. In particular, OD matrices are can be used as default descriptors of the traffic conditions and are used for (i) planning optimized routes predicting shortest and viable paths exploited by routing and path algorithms; (ii) providing info-traffic services on desktop or mobile devices, via the so called Advanced Traffic Management Systems (ATMS), ITS managing busses and vehicles (intelligent Transportation Systems), and UTS (urban traffic systems) managing semaphore networks; (iii) planning evacuations.

OD matrices are typically time dependent, and thus their dynamic real-time estimation may be needed, or at least the estimation of their values every 15 min, and distinguishing from the different days of the week (working days, festive and prefestive days). Their values are of primary interest if they represent the maximum or at least sustainable traffic values, disregarding when the traffic infrastructure cannot sustain the traffic flow. In the context of traffic flow, some methods for computing OD matrices use parametric estimation techniques (e.g., Maximum Likelihood, Generalized Least Squares, Bayesian inference). Maximum Likelihood methods minimize the likelihood of computing the OD matrix and the guessing traffic. Other methods based on traffic counts include Combined Distribution and Assignment (CDA) [44] , Bi-level Programming [59], [106], Heuristic Bi-level Programming [118], Path Flow Estimation (PFE) [137], or Neural Networks [83]. For example, Ashok and BenAkiva [8] used a Kalman filtering technique to update the OD matrix. Time dependent offline estimation deals with time-series of traffic counts.

Typically, building an OD matrix for mobility requires installing devices to count every single vehicle, and eventually recording the speed of each vehicle on the road. A traffic counter is a device that records vehicular data (i.e., speed, type, weight). At this regard, the US Federal Highway Administration defines three main traffic counting methods: human observation (manual), portable traffic recording devices and permanent automatic traffic recorders (ATR). Thus, at the level of traffic flow observation several different techniques are used: video cameras, pneumatic road tubes, piezoelectric sensors embedded in the roadway as inductive loop detectors, magnetic sensors and detectors, microwave radar sensors, Doppler sensors, passive infrared sensors, passive acoustic array sensors, ultrasonic sensors, laser radar sensors. Most of these sensors use intrusive technologies and require pavement cut; in some cases, lane closure is required, the devices are sensitive to environmental conditions and require an expensive periodic maintenance.

Several solutions have been proposed to solve the problem of an effective sensor placement for traffic counting. For example, in [54] Contreras et al. present a novel approach for studying the observability problem on highway segments, using linearized traffic dynamics about steady state flows. They analyze the observability problem (sensor placement) and propose a method that compares scenarios with different sensor placements. In [15] Ban et al. present a modeling framework and a polynomial solution algorithm to determinate optimal locations of point detectors, for computing freeway travel times. They use an objective function to minimize the deviation of estimated and actual travel times; the problem is discretized in both time and space, using a dynamic programming model, solved via a shortest path search in an acyclic graph. In [116] the performance of the sensors is measured in terms of estimation error covariance of the Best Linear Unbiased Estimator of cumulative flows in the network. Sensors are placed to minimize the sum of the error covariance and of a cost penalizing the number of sensors, using the concept of Virtual Variance. [96] defines a measure of importance for a node in a traffic network and use it to solve the sensor placement problem, by maximizing the information gain (i.e., users' routing choices). It presents a method for finding the optimal number of sensors to be placed, modeling, and maximizing the utility stemming from the trade off between cost, performance, robustness and reliability of the sensor placement. [18] describes some spatial distributions of traffic information credibility and proposes different sensor information credibility functions, to describe the spatial distribution properties. The authors propose a maximum benefit model and its simplified model to solve the traffic sensor location problem. In [14] propose a modeling framework to capture a sequential decision-making process for traffic sensor placement. Optimal sensor deployment for a single application is determined by a staged process or dynamic programming method; sensor locations for new applications can be optimally solved by the DP method considering existing sensors.

Some of the above-mentioned techniques can be used to produce vehicle classification (e.g., rural cars, business day trucks, through trucks, urban cars). Recently, other techniques have been adopted as RFID, Bluetooth, Real Time Location System (RTLS) and Wi-Fi access points [57], [143] . In some cases, the position of the vehicle can be monitored from the GPS position of mobile devices installed on the vehicle itself, or simply by using smartphone navigators (e.g., Google Maps, TomTom, Waze), that provide positions and velocities of the vehicles. In these two cases, vehicle's tracking is authorized by the users (through an informed consent) that install the device or run the mobile application on the smartphone or navigator. RFID is quite unsuitable to detect devices because of the small range of action. Bluetooth can be more suitable but it is expensive, since specific stations to collect the passages are needed. Wi-Fi access points are less reliable in detecting the presence of high speed people as in motorized sources with respect to physical devices, and GPS-related methods. In [119] the Wi-Fi analysis has been used to assess the passages of pedestrians in buildings. In [163], the quality and feasibility of using multiple solutions based on Wi-Fi and Bluetooth for people tracking have been presented providing the evidence that Wi-Fi count may be more reliable. In [75], an early experiment in tracking people flow by exploiting Wi-Fi data has been reported exploiting direct MAC address tracking. In [5], a small scale experiment has been performed for tracking a limited number of people (80 0 0) in well-known and restricted area with 20 AP. The effective precision and assessment were not provided. In [65], a similar experience has been analyses, with the aim of extracting trajectories.

On the other hand, the usage of Wi-Fi Access Points (APs) is addressed as devices to have indication of people flow and density in the city. Wi-Fi solution is viable given the high distribution of mobile devices, the low cost of a Wi-Fi AP, and the fact that a huge number of APs is already installed in the cities. This solution is quite cheap and easy to implement, also considering that many municipalities offer free Wi-Fi connectivity, and the needed coverage can be easily obtained with a small effort adding a few more APs, or just reconfiguring those already present, and thus the proposed approach is used only for selecting those to be reconfigured. Therefore, this Chapter presents mainly two major results:

1. Methodology for identification of the best placement for Wi-Fi Access Points, as detectors for collecting data for user behavior understanding, maximizing the precision of OD matrix computation as one of the most attended results. The methodology aims at limiting the costs to obtain

reasonable data for massive and systematic measuring of the whole city flows by humans. The study and solution have been validated by using the data set introduced in [151] which covers cab mobility traces, collected in May 2008 in San Francisco (USA). This result has been used to identify the most suitable AP in Florence for reconstructing OD and flows.

2. The data collected from the Firenze Wi-Fi network (instrumented for people flow tracking) have been analyzed to derive a number of information and knowledge: (i) most frequent places and hottest areas in the city, represented as heat map; (ii) daily user behavior patterns around AP in the city to understand how the city is used; (iii) OD matrix to extract people movements; and a (iv) predictive model for guessing number of Wi-Fi connections for each time slot and AP (which are directly related to people presences, behavior and flows). This result poses the basis for exploiting the produced model and instrument for early warning. That means as a tool for detecting dysfunctions or unexpected patterns in the city user movements at their early inception.

The proposed AP positioning strategy, combined with the data analysis of Wi-Fi data, constitute an innovative methodology to understand user behavior at low costs in urban areas. Such services include: enrichment of traffic sensor data (i.e., physical road sensors, cellular data), notification tools for alerts or events with huge crowds (e.g., people's flood detection, emergencies, manifestations), development of traffic/people routing and optimization algorithms, resilience management and real time monitoring tools, building of green areas or recreation activities in zone at high density of pedestrians, control of air pollution, and city cleaning, increasing city security.

2.2 User behavior analysis vs data set

User behavior is urban area is represented by trajectories, hottest places also represented as heat maps, Origin Destination Matrices, analysis of regency and frequencies. These results and model can be mathematically obtained with some specific algorithms processing singles GPS traces of the movements. In more details, the OD matrix representing flows among the zones of the city (considered for example as zip codes z or smaller areas) is defined as

$$OD_{n,n} = \begin{pmatrix} z_{1,1} & \cdots & z_{1,n} \\ \vdots & \ddots & \vdots \\ z_{n,1} & \cdots & z_{n,n} \end{pmatrix}$$

where: $z_{i,j}$ represents the total number of traffic counts from z_i to z_j (i.e., in our context how many cabs moved from z_i to z_j) defined as

$$z_{i,j} = \sum_{t \in T} n_t\left(i, j\right)$$

and, T is the set of unique cab traces, $n_t\left(i, j\right)$ is the number of traffic counts from z_i to z_j for the trace t. This means that, if the aim consists in identifying the best position for the sensors (may be Wi-Fi AP as in this case), one should have the data representing the whole set of people movements in the city, that is unrealistic, no one has those data neither the telecom operators. On the other hand, [67] present a nonlinear two-stage stochastic model to compute sensor location (classical traffic flow detector as spires) maximizing the quality of origin-destination matrix (OD) by starting from the traffic flow data. In this case, the authors presented an iterative heuristic solution algorithm, Hybrid Greedy Randomized Adaptive Search Procedure (HGRASP), to find the near-optimal locations. This approach is feasible when the flows are known. The validation of any AP positioning methodology for the people flow count is not a trivial task. In principle, one should install the APs in certain positions and demonstrate, making measures on the real context, that they produce strongly correlated data with the real people flows, among the different areas of the city. Since this approach is very expensive and unfeasible for a number of configurations, an indirect method described has been adopted as described in the following.

2.2.1 Reference data from San Francisco vs AP

Due to the above described difficulties for the analysis,the data set introduced in [151], that includes cab traces in San Francisco, collected in May 2008 has been adopted. The data set reports all the cab traces by providing precise GPS positions for each of them. In Fig. 2.1, the traces are reported on the city map (for a particular day in the time range 8:0 0 a.m. - 9:0 0 a.m.). The total data set consists of 446,079 traces based on about 11.2 million of single GPS points collected by cab movements, not only in the down-town of San

Francisco, but spanning the city's neighborhoods. Areas at higher density are those in the down-town, coherently with what you could have from the movements of pedestrians in the city. This area is covered by about 13 zip code areas.

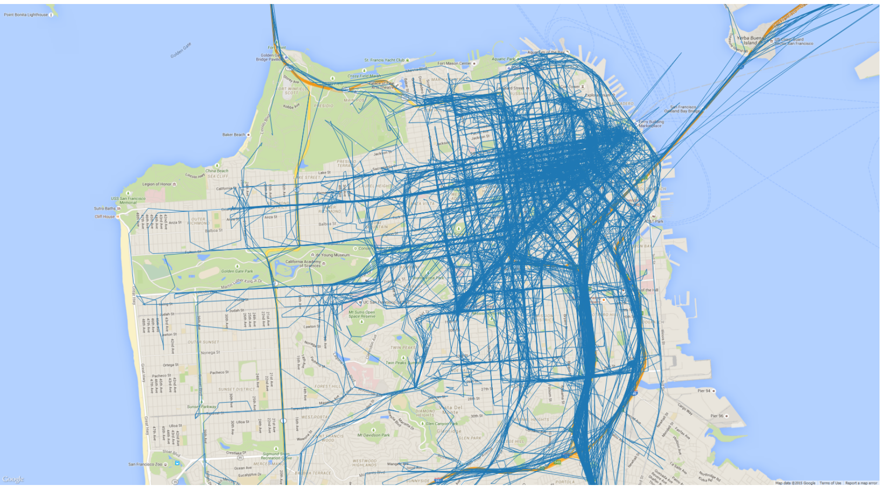


Figure 2.1: Trace flows in San Francisco on a working day of May, 8:00 AM - 9:00 AM

In order to perform an effective data analysis and visualization, some web tools for viewing and comparing flows in different scenarios were developed. At this regard, OD matrix and thus flows among the zip areas are represented with a chord diagram, to put in evidence single and aggregate contributions to the total flow count among the various city zones (in Fig. 2.2 , the chord diagram is reported for the central part of the city with 13 zip code areas). An interactive version of the produced chord based tool is accessible at `http://www.disit.org/6694`. The user can select a time interval in the day to visualize the corresponding chord diagram, which is constituted by circular sectors, each of them representing a city area; passing the mouse over a sector provides additional information about the traffic counts originated from it towards other zip areas. In this manner, it is possible to depict in a compact and intuitive way the traffic flows among the various zones. Additionally, it is also possible to remove a circular sector to simplify the diagram and make easier the analysis of the flows of interest.

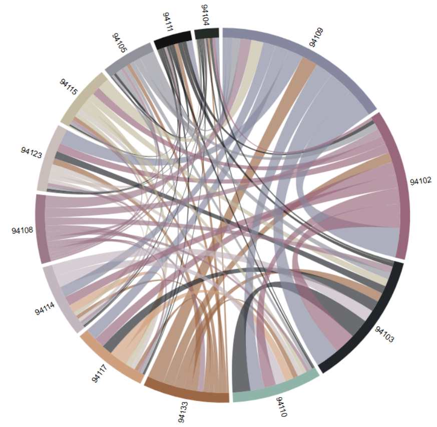


Figure 2.2: San Francisco OD matrix as a chord diagram among the 13 central ZIP areas of the city (real cab flows)

In the case of San Francisco data, the structure of the city and the position of the APs in the down-town is known (see Figure 2.3). The positions of AP in the area have been taken from OpenWiFiSpots (`http://www.openwifispots.com`). They consist of 494 Wi-Fi APs providing city services, from a total of 983 APs at disposal (also located in coffee shops, hotels, restaurants, libraries, bars, book stores, grocery stores). Therefore, and idea could be use the Wi-Fi network to estimate the people flow in the city produced by mobile devices, according to their MAC address or to hash code of the MAC address and other features of the mobile device.

This solution could be implemented by collecting the events of connection and release of mobile devices with respect to the APs (each event reports date, time, device ID to give internet access, and AP identifier). Each AP streams the collected data to a central server which anonymizes the MAC addresses, records the data, and streams the combined multi-streams to the data analytics. In alternative, some of the APs or aggregators of APs may compute the anonymization algorithm, based on a hash code of the identifiers. Once detected the passages of devices on the APs, the OD matrix as well many other information can be can be derived. This means that, these data can be explit to filter out the traces matching with hypothetical position of AP and observing if the obtained OD is still valid to represent the whole OD matrix depicting the actual situation calculated on the basis

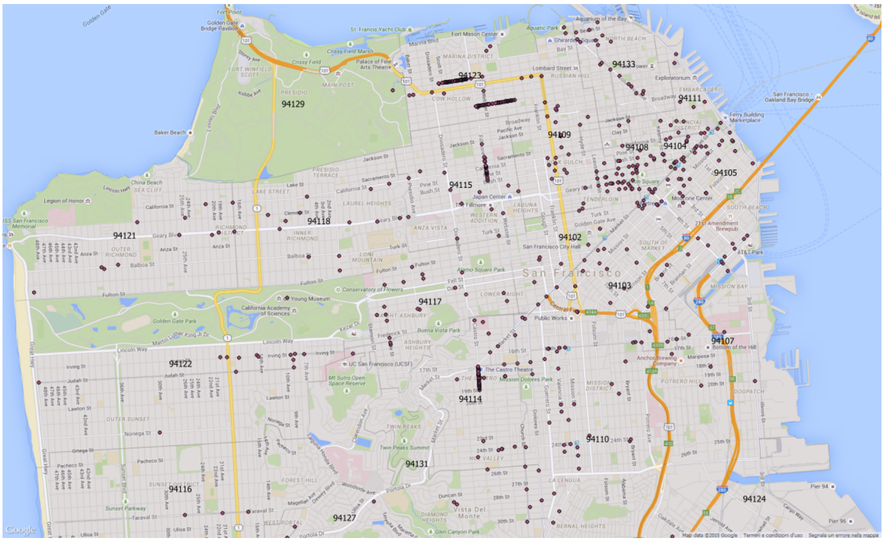


Figure 2.3: Distribution of real Wi-Fi APs in San Francisco

of all traces. This means to produce the AP positioning maximizing the correlation of the estimated OD with respect to the actual.

2.3 Methodology for AP positioning

As a first approximation, is possible assuming to have the possibility of detecting the flows by using the present APs distribution, by capturing the real traces passing within a distance of 25 m from the AP position. The proposed approach can be viewed as a sort of partial simulation based on real data about traffic flow, that is more realistic than producing fully simulated data. It is obvious that the real data captured by the APs would be probably only a part of the real traffic of people passing close to them. On the other hand, it is reasonable to verify that the simulated measures are strongly correlated to the real effective numbers.

As a general consideration, only 1470,091 trajectories were found to intersect with the real APs positions, which in the down-town are 1418,207 with respect to 494 APs. Therefore, in this manner, the available distribution of Wi-Fi APs in San Francisco is assessed, in order to collect people flows related data through mobile devices. Once obtained the observations

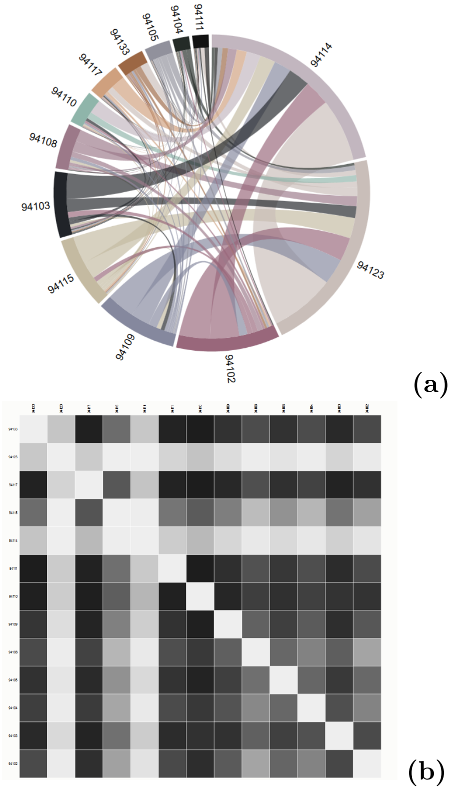


Figure 2.4: (a) Chord diagram of flow counts with real Wi-Fi APs in the city center; (b) Difference matrix among OD matrices of real flows and estimated with real Wi-Fi APs in the city center

by finding the intersections of the traces with the APs, an estimated OD matrix has been produced, as reported by the chord diagram in Fig. 2.4(a). In Fig. 2.4(b), the matrix of difference between the OD matrix of Fig. 2.2 and that of Fig. 2.4 a is reported; the differences between back and forward flows are not perceivable.

The difference matrix of Figure 2.4(b) give the evidence of the difference from the real traffic flow with respect to the flow that is estimated by using the present APs distribution in the city. The differences are reported with a gray-scale (higher is the difference, darker is the matrix element). The two OD distributions are uncorrelated (a correlation of 0.12 has been measured,

Table 2.1: AP models, cc = city-centre, bn = zip boundaries (within 300 m)

Model		Coeff	Std.Err	t-stat	p-val	Corr	APs
Real APs	β	280393.858	19874.972	14.108	0.000	0.446	983
	α	9.448	0.543	17.400	0.000		
Real APs(cc)	β	1598664.580	116546.825	13.717	0.000	0.120	494
	α	1.714	1.141	1.502	0.135		
(a) Random APs(cc)	β	690144.338	75267.849	9.169	0.000	0.835	400
	α	52.921	2.813	18.816	0.000		
(b) High Traffic APs(cc)	β	684144.945	86354.599	12.921	0.000	0.915	804
	α	10.942	0.389	28.114	0.000		
(c) High Traffic APs(bn, cc)	β	1101641.803	86354.599	11.451	0.000	0.687	448
	α	13.586	1.159	11.727	0.000		
(d) High Traffic APs 400(cc)	β	810743.094	70801.471	14.108	0.000	0.835	400
	α	24.429	1.297	18.829	0.000		
(e) Real augmented APs with High Traffic APs(bn, cc)	β	748987.390	58260.615	12.856	0.000	0.892	400
	α	39.960	1.634	24.453	0.000		

see Table 2.1). This result demonstrates the unsuitability of the present distribution of APs in San Francisco for collecting and modeling traffic flows. On the other hand, their placement was not made with the aim of measuring and observing people flows.

2.3.1 Adopting AP positioning models

On the other hand, a more efficient AP positioning scheme should achieve better correlations and smaller standard error, and thus better precision for the estimation of OD matrix (and indirectly people flows in the city). To this purpose, similarly to [67] a set of heuristics have been identified to find a compromise from precision and OD estimation. Thus, a number of different methods for AP positioning and thus for flow observations have been adopted and tested, taking them from the literature of the classical traffic flow observations strategies by humans. Then, it has been possible to start by creating a uniform distribution grid of APs, ideally placed at the middle of each street. In all cases, each AP was considered as a circular area with 50 m of diameter.

The resulting APs set, consisting of 14,959 APs (a number of devices that is surely too high to be affordable), was further reduced using different strategies as reported in the following. Moreover, the reduction is also reasonable since a uniform distribution in all the zones of the city is not feasible. There are many zones in which the flows are very low, at least in the simulated data taken into account. On the other hand, the positioning of the APs in low flow areas is not efficient.

Also, a flow prediction strategy should be able to tell where to place traffic sensors, and how many sensors to use, providing a tuning strategy for selecting the required set of sensors, with the aim of minimizing the number of traffic sensors and the costs of periodic maintenance of the monitoring infrastructure. In this section,some alternative strategies of AP placement have been provided, in order to minimize the number of APs, and to obtain a satisfactory match (i.e., statistically significant) between the real cab data and the data registered by the APs. The possible scenarios for AP distribution are the following.

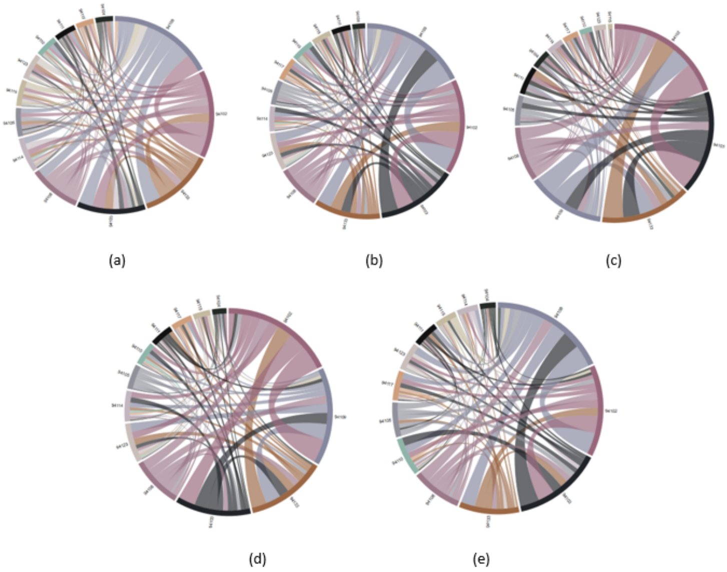


Figure 2.5: Chord diagram of flow counts. Cases as described in Table I: (a) Random APs; (b) High traffic APs; (c) High traffic APs (zip boundaries); (d) High traffic APs (top 400); (e) Real augmented APs.

a) Random APs : identification of the streets with the highest trace flow rate (those that have at least 30 0 0 traces) and then random selection of 400 APs from the AP grid described above (see Figure 2.5 a for the OD matrix). This set of APs is a subset of the set described in case (b).

b) High Traffic APs : identification of the streets with the highest trace flow rate (those that have at least 30 0 0 traces) and then selection of all the APs intersecting those traces, thus resulting in 804 APs (see Figure 2.5 b for the OD matrix).

c) High Traffic APs (zip boundaries) : identification of the streets with the highest trace flow rate (those that have at least 30 0 0 traces) and then, starting from the 804 APs of case (b), selection of those within 300 m from the zip boundaries, thus resulting in 448 APs (see Fig. 2.6 c for the OD matrix). This set of APs is a subset of the set selected in case (b).

d) High Traffic APs (top 400) : identification of the streets with the highest trace flow rate (those that have at least 30 0 0 traces) and then, starting from the 804 APs of case (b), selection of the top 400 APs (see Fig. 2.5(d) for the OD matrix). This set of APs is a subset of the set selected in case b.

e) Real augmented APs with selected high traffic APs (Fig. 2.5(e)): the real distribution of the AP in San Francisco's down-town was integrated with the top 300 AP from case (d) with the highest traffic rate. This set was then cleaned up by removing those APs that were found to be at a distance less or equal than 50 m from the real APs, and removing also intersecting APs, thus resulting in 400 APs (221 real APs, 179 high traffic APs).

The resulting OD matrix for these distributions of APs has been estimated by computing the intersections between the real cab measures with the placed APs, according to a capturing range of 25 m radius. The OD matrix for this configuration was generated by evaluating the traffic counts among the various APs, grouped by the zip code they belong to. The chord diagrams of these scenarios are reported in Fig. 2.5.

2.3.2 Assessing AP positioning models

comparative analysis of traffic flows was conducted, using the above cited set of cab traces, consisting of 11,219,955 unique detection from 536 cabs, with respect to the above described scenarios. With the above assumptions, the real set of APs placed in the city center was used to sample the original data set, by calculating the APs intersections with the cab traces. The OD matrix was calculated from the sampled data set (considering each city zip code as a separate area), reporting the traffic counts among every city's area. This procedure was repeated by choosing the APs with a pseudo random technique, and by placing the APs only in the roads with the biggest amount of traffic. After that, a comparative statistical analysis was conducted for each configuration (see Table 2.1). The traffic flow outcome is predicted with a linear regression, finding the parameters that best fit the data in the linear model

$$y = \alpha x + \beta \tag{2.1}$$

where x is the dependent variable or predictor (i.e., traffic counts as registered by the sensors), and y is the outcome (i.e., predicted traffic counts). Building the model 2.1 using the set of real APs gives a correlation of 0.446 (0.120 using the real APs in the city's downtown) with respect to the real traces.

A number of cases have been assessed following the placement strategies described in Section III. In case (b), the APs have been placed on the roads with the highest traffic rate, producing a model with a correlation of 0.915, and of 0.835 using only the top 400 APs, as described in case (d); using random APs of case (a) gives a correlation of 0.835; using the APs only within 300 m from the areas' boundaries, described in case (c), gives a correlation of 0.687. It is clear from this data that using the real APs set produces noise and doesn't produce a reliable model for flows prediction. Randomly distributing the APs gives a better correlation with the cab traces, while reducing the number of APs and considering only those in the proximity of each area, gives a good correlation while maintaining a limited number of APs. The set of real APs of case (e), integrated with some other APs and cleaned up from some not useful or redundant elements (i.e., mutually intersecting APs), gives a correlation of 0.892. To visualize the results of the various OD models a web interface was developed, with the possibility to view the chord diagram for each computed configuration. The interactive versions of the chords diagrams in which it is possible, for each couple of locations, to see the effective flows (in a way and in the other, for a given time slot of the day) are accessible at `http://www.disit.org/6694` . This approach allowed to identify which are (i) the positions of the new APs to be added (i.e., 179) and (ii) the minimum set of APs already in place that must be used for data acquisition (i.e., 229). The second point allows keeping limited both the network bandwidth and the workload for the estimation of the OD matrix.

A fully mathematical approach could be applied for the identification of the best AP in San Francisco having dense traces, but it would not be suitable for the re-computing it in a new fresh area (without data). In substance if a position of APs is identified in San Francisco just minimizing the error, the position of the AP would not follow any rule that could be re applied in a different city to position the APs or select the APs to be reconfigured. Thus, has been decided to test a set of heuristics and select the best, and thus to use the identified approach to position/select the AP in Florence.

2.4 City user's behavior analysis

The above described AP placing methodology has been exploited in the city of Florence (Italy), for selecting the AP needed for the estimation of city users' behavior. Typically, it can be supposed to derive users' behavior from data collected from the *Telecom* operators. On the other hand, the mobile operators are not authorized in reselling data reporting the fine tracking of their users, even if the mobile/user ID is anonymized. In most cases, mobile operators provide data collected every 15 min, reporting the number of users for each cluster of their cells and without tracking the movements from one cell/cluster to another. Some of them provide OD matrices statistically estimated starting from the described data and thus providing a limited precision in space and time, and not in real time. These facts limit the possibility to use those data to perform a city users' behavior analysis, area clustering and the usage of data for early warning.

On the contrary, the usage of Wi-Fi network can be used for tracking city users' behavior with the needed resolution (in space and time), by accessing to data anonymously and exploiting them according to an informed consent with the users when they connect to the Wi-Fi. The above presented methodology for AP placement has been used on the Firenze Wi-Fi infrastructure to identify the suitable APs to be considered for the analysis, with the aim of reconstructing city users' behavior in space and time. At this regard, Florence offered a free Wi-Fi network (Firenze Wi-Fi) consisting of about 1500 APs. One relevant issue is that Firenze Wi-Fi APs were installed with the aim of providing a good Wi-Fi coverage in the city's centre and in relevant city services as hospital and university.

As a first step, the most active places and areas have been identified to the monitored, on which the above presented methodology would be applied. This action has been performed by interviewing the municipality and by using data collected from mobile App (Florence, Where, What?), available for Android, iOS and Windows Phone stores [20]. That App work with smart city API based on Km4City [134] and provides general information to the city users almost uniformly in the city and on multi-domain since it provides information and suggestions on: public and private mobility, culture, energy, accommodation, restaurant, tourism, free Wi-Fi, bus lines, car parking, pharmacies, ATMs, events, etc. These services are accessible with geo information.

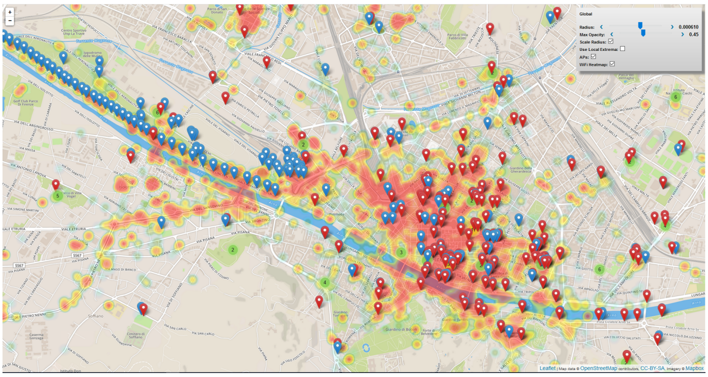
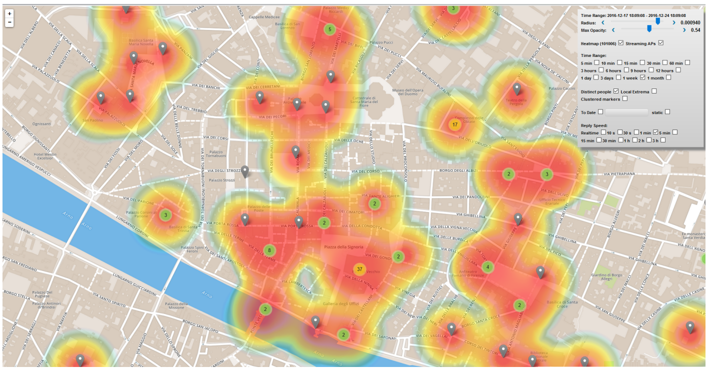


Figure 2.6: Heat-map comparing city users' most frequented places vs the position of the 1500 Wi-Fi APs of the whole network (using a colour gradient scale to discriminate between different densities of measures)

Figure 2.6 reports the heat-map derived from the city users' movements in the city by using the App with overlapped the position of the 1500 AP of the Wi-Fi network. Considering the architectural and environmental constraints of the historical center of Florence (that is part of the UNESCO World Heritage list), you cannot place APs wherever you want: in most cases, the nearest AP has been switched to the predicted one, rather than effectively place the desired AP. The resulted analysis allowed us to select the best points and from these about 345 candidates APs to be configured and used as probes, selected from more than 1500 AP located in the city. The data related to the user behavior tracking via Wi-Fi has been collected in the period from May 2016 to December 2016. They consist of about 56 Million of events of connection and disconnection. Typically, the 60% of connected users are excursionists that stay in the network only for less than 24 h. In the last 6 months, about 1.15 Million distinct users have been detected, which means about 2.3 million of distinct user per year in a city with about 14 million of new arrivals per year and 350.0 0 0 inhabitants. So that about the 16% of people flow has been tracked. Predictions from the positioning methodology with the existing APs data, finding the APs to be added and those that were useless for the study. According to the selected AP, the resulting heat-map describing the distribution of measures performed by the AP is reported in Figure 2.7. The developed tool allows customizing the provided map have been compared, for example varying the radius and the opacity of the heat spots.


Time Range: 2016-12-17 18:09:08 - 2016-12-24 18:09:08
Radius: 0.000940
Max Opacity: 0.54

Heatmap (101006) ☑ Streaming APs ☑

Time Range:
5 min □ 10 min □ 15 min □ 30 min □ 60 min □
3 hours □ 6 hours □ 9 hours □ 12 hours □
1 day □ 3 days □ 1 week ☑ 1 month □

Distinct people ☑ Local Extrema □
Clustered markers □

To Date □ static □

Reply Speed:
Realtime □10 s □30 s □1 min ☑5 min □
15 min □30 min □1 h □2 h □3 h □

Figure 2.7: Segment of the heat-map reporting hottest places detected by using selected Firenze Wi-Fi APs, in Florence downtown

The data analysis allows identifying the hottest places (in terms of events on the APs) as reported in Fig. 2.8, where the names of the locations and the precise latitudes and longitudes have been truncated for safety reasons. On the other hand, they are also well known location to everybody in the world.

Similarly, a number of visual analytics graphs are produced, such as: the numbers of distinct users during the day, the average connection time per AP, the number of working APs in the last minutes, the regency (percentage of new users with respect to the already seen users) and frequency of users. This last view is of particular importance since it allows estimating the number of new users coming into the city. Indeed, it is worth noting that for cultural cities like Florence, newcomers are typically tourists (excursionists) or business people that stay in the city only for a few hours and days.

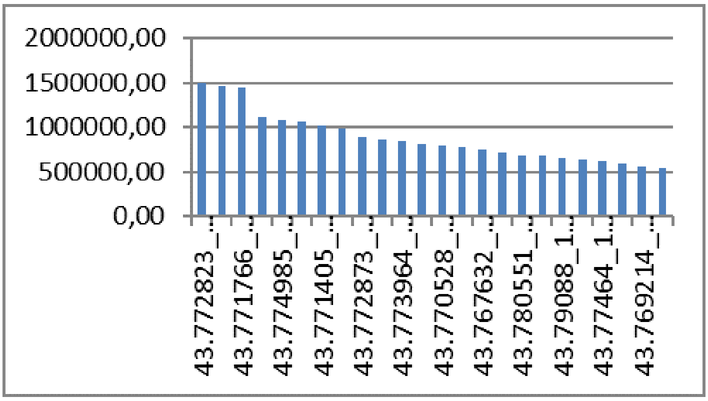


Figure 2.8: Segment of the heat-map reporting hottest places detected by using selected Firenze Wi-Fi APs, in Florence downtown

Every working day the network identifies about 34.0 0 0 distinct users and among them, about the 10% are new users for the net- work in the period. For the present analysis, has been assumed that new users exploit the city up to 10 days before leaving, while old users continue to exploit the city beyond that limit.

Fig. 2.9 reports the users regency found in the range 1-28 days. Every column in the histogram shows the number of distinct users (y-axis) that at most returned in the city within a defined number of days (x-axis). It is evident from this pattern, that most of the users using the Wi-Fi network are exploiting the city for a few days before leaving. This kind of analysis can be performed at large scale (i.e., considering the whole city) or simply by observing the user behavior in some zones of interest. For example, the analysis of regency in the historical city centre (which is normally the most exploited part of the city) can provide valuable insights, since it allows understanding which cultural attractions people prefer to visit, or where and how often they return to them.

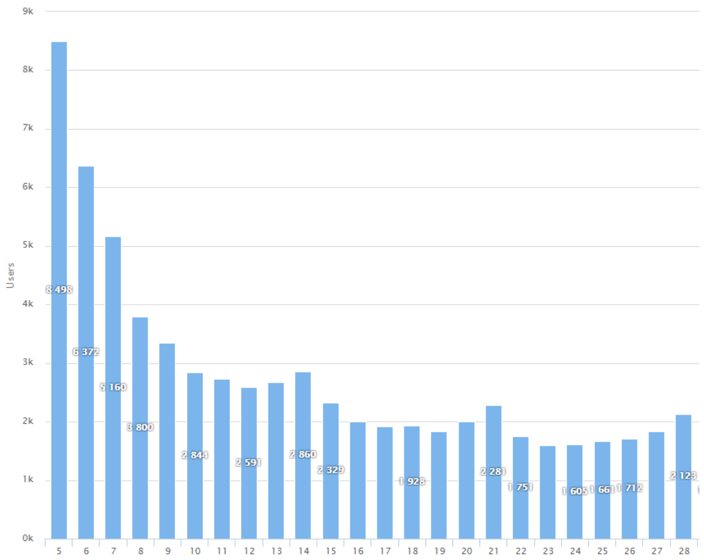


Figure 2.9: Segment of the heat-map reporting hottest places detected by using selected Firenze Wi-Fi APs, in Florence dowtown

2.4.1 Origin destination analysis for people flow

To better understand the movements in the city, it is mandatory to perform flow analysis to effectively evaluate user's behaviour. Since in the downtown the APs are also overlapped this issue has to be taken into account. The measures performed by the mobile APP (as described in first part of Section V) have been also used to define a compromising size for each area collecting accesses to the Wi-Fi. On the basis of the tracked city users among the APs of the Wi-Fi network it is possible to computer the OD matrix according to the origin and destination area defined by the distribution of the APs in the city. On the other hand, the OD matrices are typically quite sparse as one can see in Figure 2.10(a), where the OD matrix for Florence is reported. Figure 2.10(b) reports a new approach for depicting and analyzing the OD matrices. It is a visual analytic approach for depicting an OD matrix as what we call OD Spider Flow in which the analyst may identify the hottest areas of the city as those with larger and darker points/dots. When a dot is selected the graph reports the major (in/out) flows from that origin to the most probable destinations, also providing the percentage of probability on the destination dots. Every flow is depicted with an arrow and a coloured circle reporting the total number of occurrences and their percentage with respect to the total flows. The analysis can be performed for the whole city users or only for the new arriving users (with respect to the last 10 days), for each time slot of the day or for the whole day, for incoming and outgoing flows, and at different level of resolution (zoom). Zooming in/out the map redraws the flows with a different cluster zone, making possible to depict more detailed or aggregated flows between the various zones. The classical OD matrix can be shown as well from the same tool, also calculated with a customizable range within the city's center, for the chosen flow configuration (i.e., cluster area's size, hour of the day, user profile). This kind of derived information can be used for running the services in the city, to plan the cleaning, to distribute the security people, etc.

2.4.2 Understanding city usage from AP data

From the analysis of the OD matrices and/or OD Spider Flows it is evident that different parts of the city are differently used by different city users. AP presents different kind of trends in the usage of the Wi-Fi along the 24 hours and in the different days of the week [101]. For example, it's may possible to have some areas by which the people typically arrive (station) in the morning and leave in the afternoon while they are less accessed at lunch time. For example, some APs could have a huge workload only during mornings or evenings (when people go/back to/from work), others only on late evenings (when people go out for entertainment), others only of festive days etc.

In Figure 2.11, an example of trend for a certain AP along the 24 hour of the day. The trend of Figure 2.11 has been estimated by computing the averaged value per time slot of a certain AP every working day, extracting data from the 56 million of data described above.

In Florence, as in many other touristic cities, the issue is much more complex, since a lot of different city users' kinds (with different aims) use the city at the same time during the working days, and as well as on Saturday and Sunday.

Therefore, in order to tune the services in the city (security, cleaning, transport, etc.), it is very important to infer patterns and analyse city user's behavior. In the present scenario, the major interest is related to understand

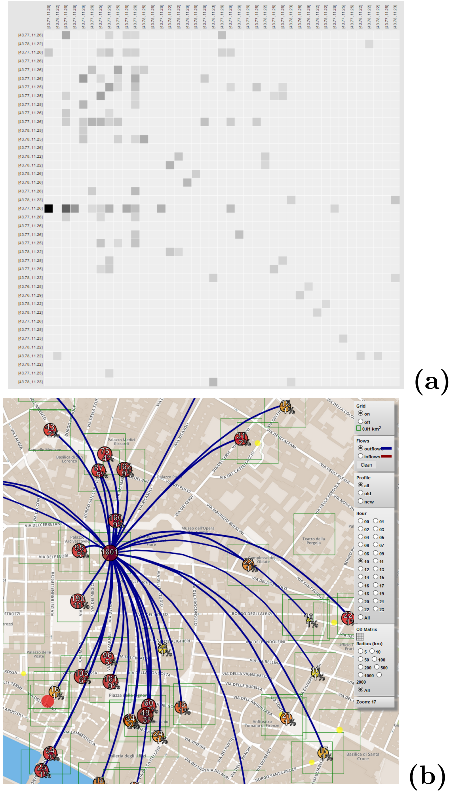


Figure 2.10: OD Matrix for Florence downtown: (a) classical view; (b) advanced proposed view

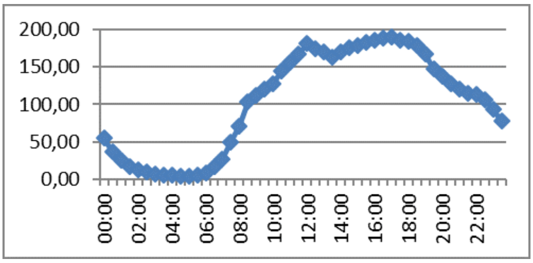


Figure 2.11: Typical AP trend in terms of number of connections along the 24 day, a working day

how the city is used by city users which in turn can be re-conduced to the problem of understanding how APs work and are used. The idea is to exploit some data mining techniques clustering AP on the basis of their normalized temporal pattern. This will allow grouping them in areas and put in evidence the flows

and the service exploitation in the different city's zones. Clustering the APs' behaviors can help to understand if there are zones having a similar usage and exploitation and hence similar flow patterns, and needs in terms of services.

According to the data collected from the Wi-Fi network described at the beginning of Section V, the averaged trend along the 24 hours of the day, for each AP, for each day of the week has been computed. Since the main interest is to find the similar patterns for each AP a Scale Factor and the normalized averaged pattern (from 0 to 1) has been computed. This resulted in 345 APs, on 7 days, on 48 time slot for the day (one every 30 minutes) (from 00:00 to 00:30, from 00:30 to 01:00 and so on until 23:30). A preliminary analysis of AP patterns showed a marked difference between festive and ferial days. For this reason, the clusterization of time series has been chosen by keeping track of their respective day of week, thus considering working days, Saturdays and Sundays as three distinct groups. From the statistical point of view, the temporal pattern for each AP presents an average and an interval confidence for each time slot as depicted in the examples reported in Figure 13.

Since I'm interested in finding similar patterns for the APs, a clustering approach has been adopted to find similarities in time series as in the Dynamic Time Warping [195], and by using different clustering algorithms and metrics to evaluate both the better ranked clustering algorithm and the proper number of clusters. Among the clustering algorithms the results obtained have been compared by using: k-means clustering algorithm minimizes the within-class sum of squares for a given number of clusters [121], [90], hierarchical clustering [176], density-based clustering or subspace clustering. Unlike k-means clustering, hierarchical clustering builds a bottom-up hierarchy, and does not need to specify the number of clusters. For the clustering, the closeness of cluster elements can be determined by using (a) complete linkage clustering (i.e., finds the maximum distance between points of two clusters), (b) single linkage clustering (i.e., finds the minimum distance between points of two clusters), (c) mean linkage clustering (finds all pairwise distances for points of two clusters, calculating the average), (d) centroid linkage clustering (i.e., finds the centroid of each cluster and then calculate the distance between the centroids of two clusters).

2.4.3 AP clustering experimental results

In this section, the comparative analysis among some of the above mentioned different clustering methods is reported. It should be noted that, different clustering techniques and, even for the same algorithm the selection of different parameters or the presentation order of data objects may greatly affect the final clustering partitions. Thus, the adoption of rigorous evaluation criteria is mandatory to trust the cluster results: selection of model and clusters number. As first step, the cluster tendency i.e., the hypothesis of the existence of patterns in the data using the Hopkins statistics [16] has been conducted. Hopkins statistic has been used to assess the clustering tendency of the data set by measuring the probability that a given data set is generated by a uniform data distribution (i.e., no meaningful clusters). Hopkins statistic is equal to 0.2186, thus the data is clusterable. As a second step, two clustering techniques have been adopted and compared using the above described observations and data sets of AP patterns in the 24 hours (Monday-Friday, Saturday and Sunday). The first technique was a sort of K-means clustering algorithm, partitioning around medoids (PAM) which are the most representative elements in the cluster instead of the centroid as in the k-means. PAM approach is also called K-medoids [105]. The second approach is the Model-based Expectation- Maximization algorithm or EM algorithm (EM method) [58], [125]. It is a generalization of the k-means approach that uses an iterative process to find the maximum likelihood (or the maximum a posteriori estimates of parameters, MAP). The algorithm's iteration consists of two steps: the expectation step (E) which, using the parameters' current estimation, calculates a function for the expectation of the respective log-likelihood; and a maximization step (M) which calculates the parameters maximizing the expected log-likelihood from step (E). The estimated parameters are used to calculate the distribution of latent variables in the next iterative step E.

A model-based method was used to evaluate the number of clusters/-groups and the BIC criteria to determine the best model [71]. Under this approach, each mixture component represents a cluster, and group memberships are estimated using maximum likelihood [58]. The maximum likelihood estimator (MLE) of a finite mixture model is usually obtained via the EM algorithm [58], [125]. In the multivariate setting, the volume, shape, and orientation of the covariances can be constrained to be equal or variable across groups. Table 2.2 reports six possible models with the corresponding distribution structure type, volume, shape, orientation, and associated model names.

Table 2.2: Geometric characteristics of mixture models

Model	**Distribution**	**Volume**	**Shape**	**Orientation**
EII	Spherical	Equal	Equal	-
VII	Spherical	Variable	Equal	-
EEI	Diagonal	Equal	Equal	Coordinate axes
VEI	Diagonal	Variable	Equal	Coordinate axes
VEE	Ellipsoidal	Variable	Equal	Equal
VVE	Ellipsoidal	Variable	Variable	Equal

With the Elbow method (as reported in Figure 2.12), the solution criterion value (within groups sum of squares) will tend to decrease substantially with each successive increase in the number of clusters: after 8 clusters the observed difference in the within-cluster dissimilarity is not substantial. Consequently, it's possible to assert that the optimal number of clusters to be used seems to be 7. Note that identifying the point in which a "kink" exists is not a very objective approach and is very prone to heuristic processes. For these reasons, the Gap statistics [179] has been computed to assess the optimal number of clusters in the data. From this analysis reported in Figure 2.13, the estimated number of clusters K=12.

Finally, Figure 2.14 shows the average BIC (Bayesian Information Criteria) values for six different mixture models using the model-based approach over a range of different numbers of clusters [125]. With the VEE mixture model, the maximum average BIC score is reached at 10 clusters. In addition, the VVE mixture model also achieves higher BIC values than the VEE model up to 10 clusters. Therefore, the model-based approach favours the diagonal model which produces higher quality clusters. The BIC analysis selects the VVE model at 10 clusters. Note that although the BIC analysis does not select the best model, it allowed selecting the better number of clusters in this data set.

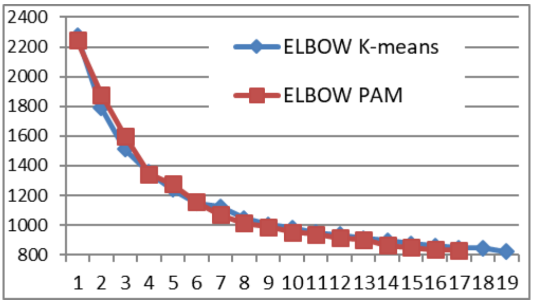


Figure 2.12: Optimal number of AP clusters via Elbow criteria (comparing K-means and PAM): within sum of square function

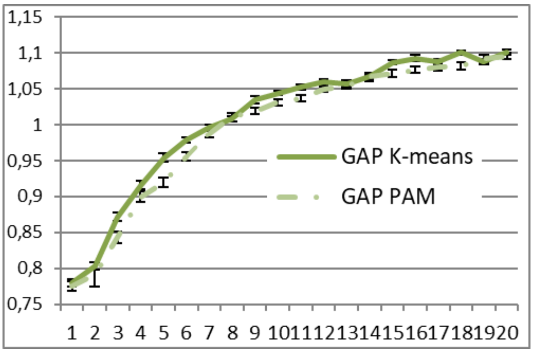


Figure 2.13: Optimal K number of clusters via Gap curve (comparing K-means and PAM)

The Dunn index [61] has been used as a measure to assess the validity of cluster techniques. Dunn index is based on inter-cluster distance and the diameter of cluster hypersphere. It can be seen that PAM clustering performs the best with 12 clusters (Dunn index for PAM is equal to 0.0798, for K-means is equal to 0.0730 and for Model-based is equal to 0.0478). As a final result, the EM algorithm with 12 clusters has been adopted for massive and continuous computing. On this regard, Table 2.3 reports the average standard deviation and the related population of each AP cluster.

In Figure 2.15, the distribution of clustered AP in the Florence map for day kind: Monday-Friday, Saturday and Sunday in which AP of the identical color belong to the same cluster disregarding the day kind. From

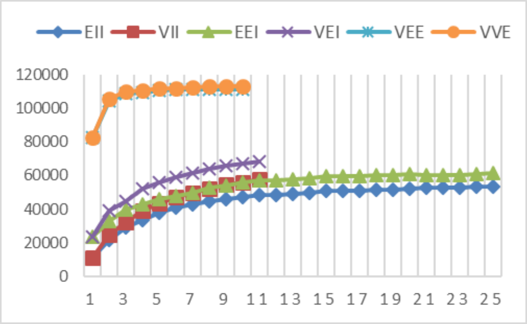


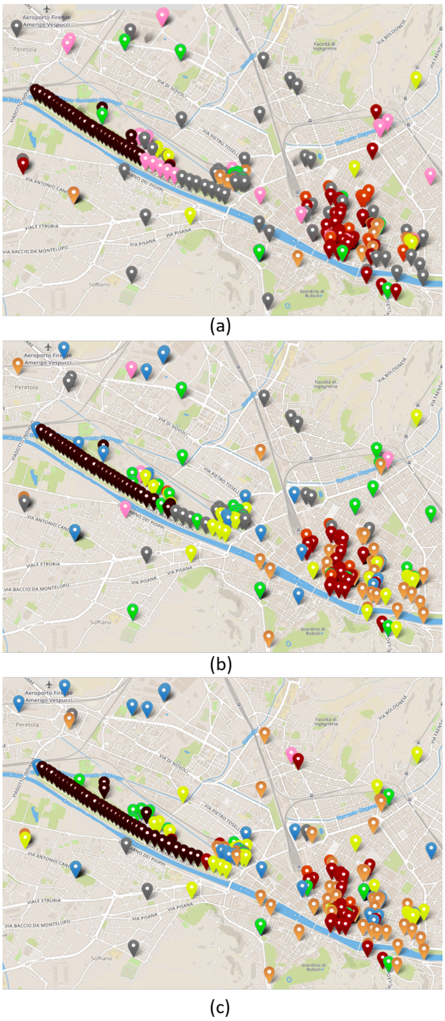
Figure 2.14: Average BIC for mixture models vs K number of cluster, higher values are better, the curves are truncated at the best value for K they found

Table 2.3: Standard deviation and population for AP clusters. W: Working days, Sa: Saturday, Su: Sunday

Cluster ID	Avg.Std. Dev.	Population
1	0.2379	W: 172, Sa: 23, Su: 24
2	0.0849	W: 23, Sa: 43, Su: 43
3	0.0882	W: 8, Sa: 42, Su: 34
4	0.1820	W: 3, Sa: 30, Su: 26
5	0.1059	W: 20, Sa: 15, Su: 14
6	0.0822	W: 38, Sa: 15, Su: 8
7	0.1311	W: 9, Sa: 57, Su: 34
8	0.1374	W: 2, Sa: 23, Su: 55
9	0.1226	W: 4, Sa: 32, Su: 38
10	0.1460	W: 52, Sa: 12, Su: 3
11	0.2487	W: 11, Sa: 13, Su: 21
12	0.1617	W: 1, Sa: 28, Su: 31

Figure 2.15, it can be noticed that group of APs located in the Cascine park (black) is enlarging passing from working days to Sunday, while the cluster of downtown (dark red) is losing some of its APs passing from working days to Sunday. While some of them remain stable: mainly those located in the major attractions for tourists.

Figure 2.16 reports the normalized shapes of the 12 clusters identified



(a)

(b)

(c)

Figure 2.15: Map of AP clusters: (a) Monday-Friday, (b) Saturday, (c) Sunday

which resulted from the best clustering algorithm, the EM. It can be noticed that the second cluster presents APs with relevant activity during the morning and afternoon respecting a break for lunch. Moreover, some clusters provide an evident activity in the afternoon with respect to the morning or vice versa, but with different proportions. A few of them present significant activity also after dinner and in the first hours of the night, as clusters number 1 and 9. So that, it is evident where the city is active during the night.

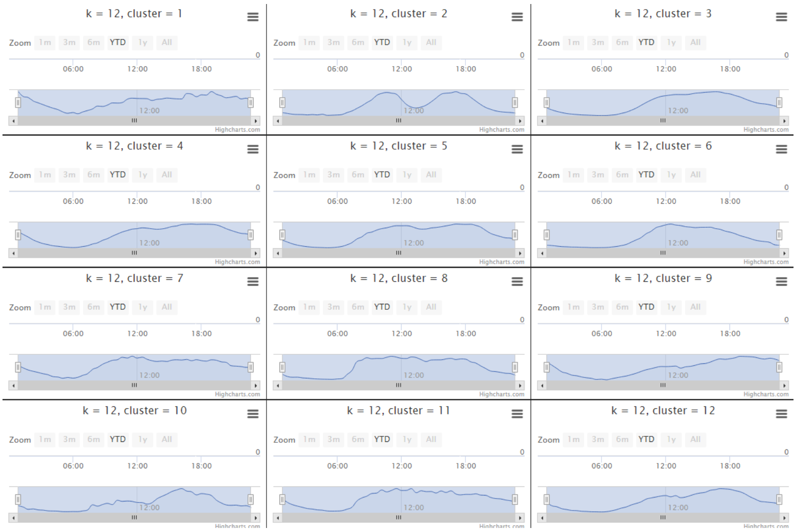


Figure 2.16: The shapes of the AP clusters with k = 12 and EM clustering algorithm

2.5 Predicting access point connections

The ARMA forecasting equation for a stationary time series is a linear (i.e., regression-type) equation in which the predictors consist of lags of the dependent variable and/or lags of the forecast errors. For ARMA [5,1]:

$$\begin{aligned} y_t - y_{t-1} = \epsilon + ar_1(y_{t-1} - y_{t-2}) + \\ + ar_2(y_{t-2} - y_{t-3}) + ar_3(y_{t-3} - y_{t-4}) + \\ + ar_4(y_{t-4} - y_{t-5}) + ar_5(y_{t-5} - y_{t-6}) \end{aligned}$$

Where: $ar_1, ar_2, ar_3, ar_4, ar_5$ are determined during the identification of the model minimizing the root square error during the learning period, ϵ is an independent variable with normal distribution and zero mean.

In Figure 2.17, two examples of AP time series with prediction are reported. Each of them reports: in blue line the average value of the cluster at which the AP belong; the light blue bound describes the interval confidence of the reference cluster of the AP; the red line the actual value of the day; the orange bound describes the interval confidence obtained by the distribution of the value of the AP in the past; finally, the RED segment (second part segment) is the effective prediction by using the ARIMA model. Please note that, the adopted ARIMA model does not take into account the value collected by the same AP in the day. In fact it has been preferred to use the predictive model for detecting dysfunctions and not to follow the most probable next values. The detection of critical situation can be obtained making the difference from those two approaches/estimations.

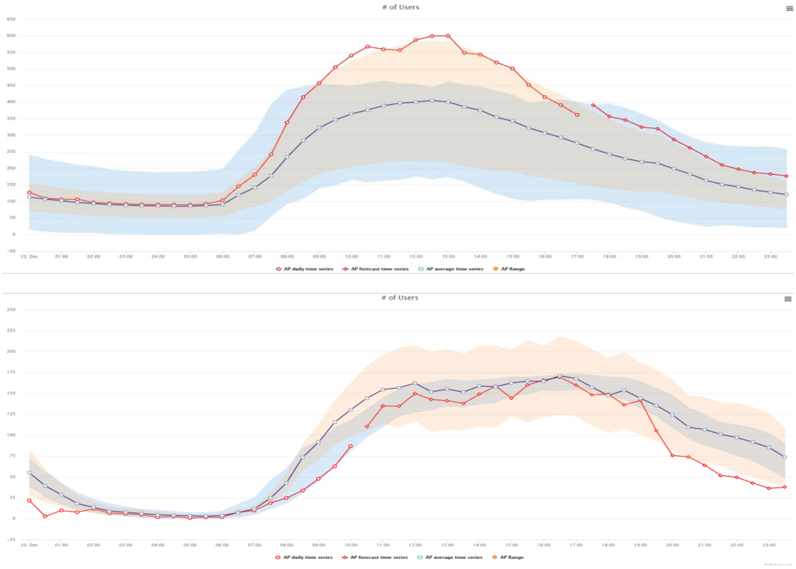


Figure 2.17: APs time series with their respective cluster ranges (see details in the text)

2.6 Considerations

Understanding and predicting city user behavior is one the major topics in the context of Smart City to optimize and tuning city services (security, clean, transport,..) and to be ready in reacting via anomaly detection. In this Chapter, it has been presented a method for AP placement, and a number of algorithms, techniques and solution for estimating city user behavior: heat-map, OD Spider Flow, clustering of AP usage in the city, predictive model. The proposed methodology is general and can be applied to different urban scenarios, in the context of Smart City traffic and people flow assessment and management. It makes use of Wi-Fi AP distributed in the city. A comparative analysis has shown that is possible to have a reasonable precision in assessing city behavior by AP positioning and collecting data from Wi-Fi, as demonstrated by a validation based on real data. The proposed approach allows identifying which are the needed APs to be added, with respect to the APs that are already in place in the city, to exploit the whole infrastructure of Wi-Fi, also for people flow monitoring and assessment. The proposed methodology has been applied to identify significant APs in the city of Florence (Italy). Thus collected data were analyzed to produce usage metrics and studying AP usage. To this end, several clustering techniques have been adopted to identify the better clustering approach for grouping city users' usage trends in the day for each city area. The results have shown that about 12 different major clusters/patterns have been identified. Each AP can be classified with respect to a cluster trend and provides its specific own scale. The corresponding AP data, trend, and cluster can be used for predicting number of accesses and thus city usage, as a well as for detecting unexpected trends incepting in the different places of the city (they may be due to programmed events as well as to detect anomalies as early warning tool). The analysis have been performed by using various clustering algorithms whole calculating different informative criterion to select the best and assess their objective quality. The resulting model proves to be effective for connection to AP forecast in the entire Wi-Fi network and potentially for early warning.

Chapter 3

Available Parking Slots Prediction

This chapter is focused on presenting the research results regarding a solution to predict the number of available parking slots in city garages with gates. With this aim, three different predictive techniques have been considered and compared. The comparison has been performed according to the data collected in a dozen of garages in the area of Florence by using Sii-Mobility National Research Project and Km4City infrastructure. The resulting solution has demonstrated that a Bayesian regularized neural network exploiting historical data, weather condition, and traffic flow data can offer a robust approach for the implementation of reliable and fast predictions of available slots in terms of flexibility and robustness to critical cases.[1][2]

[1] This chapter has been published as "Predicting Available Parking Slots on Critical and Regular Services by Exploiting a Range of Open Data" in *IEEE Access Open Journal, 2018.* [13].

[2] *Acknowledgments:* This work was supported in part by MIUR, in part by the University of Florence, and in part by companies through the Sii-Mobility National Project on Smart City Mobility and Transport under Grant SCN112.

3.1 Introduction

Prediction of available parking spaces is a complex non-linear process whose dynamic changes involve multiple kinds of factors. Parking facilities provide several different working conditions. Some of them are dedicated to a specific facility (football stadium, hospital), others on multipurpose (station, expo, etc.), and others on outskirts of town. Variability and performance are one of the problems to be addressed, together with the precision in critical time slots, which is when the parking is getting full, running out of available slots.

In our cities, the number of vehicles is getting higher and higher if compared to the development of the surrounding urban spaces; thus, the services providing available parking slots are becoming even more relevant for urban mobility management. Drivers are wasting a considerable amount of time while trying to find a vacant parking lot, especially during peak hours and in specific urban areas. Car drivers, in dense city districts, usually spend from 3.5 to 14 minutes to look for a slot [167]; this means spending money and producing pollution, thus affecting the general society costs. Consequently, looking around for available parking spaces may depend on a peculiar number of different reasons: different travel motivations, garage proximity to final destination, price differences among garages, the lack of familiarity with the selected urban area, etc.

Looking for parking slots does not only cause annoyance and frustration to drivers, but it is expected to have a significant negative impact on the efficiency of the transportation system within the urban tissue, and sustainability. To look for an available parking brings forth unnecessary traffic workload and may affect the environment negatively due to an increase of vehicle emissions. These issues are true for parking silos with gates, as well: they can be full in certain areas and time windows; while in other areas, they may become full unexpectedly and/or due to apparently unknown conditions to drivers.

Since a long time now, it is possible to collect real-time parking information i.e., capacity, garage prices, number of empty parking slots in the silos or in the area, thus being able to realize statistics predictive models. Recently, researches have discovered that big data and artificial intelligence may exploit the relevance of other data sources, such as the garage proximity, traffic flow information, and any information related to weather conditions, to calculate precise predictions more reliably.

In the context of monitoring and predicting the parking garage status, a solution to predict the number of available parking slots (not taken) has been analyzed as to parking garages with gates (e.g., silos, or on at, or under station) belonging to two different types: they carry out a regular easily predictable service or they deal with strongly randomized cases (e.g., from suburbs hospital parking to parking locations accomplishing multiple services: stations, theaters, fairs). The approach has the advantage to be robust with respect to critical cases such as when the number of free slots reaches zero, or when some data are missing in the stream.

The proposed prediction model has been created in the context of the national smart city Sii-Mobility research project of the Italian Ministry of Research for terrestrial mobility and transport. It exploits open data and realtime data of the Km4City infrastructure located in the Florence/ Tuscany areas and corresponding to the current Smart City solution (see Section 1.1) .

3.2 State of the art

The car parking activity by a driver is influenced by multiple factors - i.e., the walking distance to destination, driving and waiting time, parking fees, service level, parking size, safety 7, [109], parking price, availability and accessibility [156]. In particular, two important aspects in the parking decision-making process by any driver are: the number of available parking spaces (if known), and past experience in finding available lots. In fact, drivers who are aware about parking availabilities are 45% more successful in their decisions than the ones without such knowledge, when arriving to their parking facilities [38]. Parking facilities can be indoor/outdoor and public/private. In this context, pareto-optimal routes are selected for drivers when planning trips [200]. In more details, parking slots can be located on the street or in parking garages with gates. In terms of prediction models, there is a substantial difference between parking garages and street parking. In fact, in parking garages, it is very easy to count the total number of available slots by considering the tickets released at the entrance gate, and the outputs from the exits. On the other hand, as to streetparking, occupancy could be detected by means of some distributed sensor systems. For such reasons in literature there are two distinct research lines, focused on both street-parking prediction and free/available parking slots inside garages [181].

[181] have also proposed some integrated theoretical models for street

parking predictions, taking into account the effectiveness of both solutions in a central commercial district. Moreover, in [201], identifying where people actually park on the basis of a trajectory analysis has been proposed as a solution. On the other hand, those data have to be accessible. The street-parking problem in San Francisco has been tackled in [49], predicting the occupancy rate (defined as the number of occupied parking spots over the total availability) of parking lots in a given geo-located zone in a future time [49]. The solution works with aggregated parking lots, aiming at reducing errors in parking prediction according to different travel behavior in different regions. On the other hand, [49] discretized the day into 24 intervals, and performed the principal component analysis, PCA, on time series to model the trend of occupancy. Thus, four different predictive approaches (Auto-Regressive Integrated Moving Average approach, Linear Regression, Support Vector Regression, and Feed Forward Neural Network) have been used to investigate the prediction errors. Comparison has shown that Feed Forward Neural Networks produced the best predictive model, presenting a Mean Absolute Percentage Error (MAPE), 1 hour ahead, of about 3.57%. In this case, only well-defined and stationary cases have been addressed using historical data without taking into account contextual data. Therefore, the solution works well only on regular days, which have easily predictable conditions for regular parking clients. On the same path, in [180], an unsupervised clustering approach (Neural-Gas Network [124], [180]) has been adopted on the data to identify the similar street-parking behavior over 24 hours, using a small data sample, and a temporal resolution of 15 minutes. In reality, in [124] a strong variability among the behavior of different street-parking spaces has been presented but what is clearly missing is an effective prediction model. In [99] the solution proposed in [211] (which was a method based on Wavelet Neural Network) has been improved with the aim to predict the availability of a parking lot every minute, in an interval time of 15 hours (from 6:00 AM to 10:00 PM), using a three-days training set and one day as test set. Also, in this case, the predicting precision has been in the range of 3-10% in term of Mean Square Error (MSE). The authors have declared that in critical cases (where available slots are close to zero) the prediction error rapidly increases, and the only way to reduce it is to modify the training set. On the other hand, we would like to stress that it is precisely in critical cases when free slots are getting fewer and fewer, that precision has to be higher, so as to provide a good service for final users; thus, predictive models and services for prediction are much more needed and relevant. As to street parking, in [190] a two-step methodology for occupancy prediction based on sensor data has been proposed: the first step consisted in a real-time prediction scheme based on recurrent artificial neural networks; the second module estimated the probability of finding available parking space in relation to traffic volume, day type and time slot along the day. The resulting MAPE for the prediction at 30 minutes was in the range of 1-4%. Moreover, in [40] a mathematical model has been proposed and it is based on queueing theory and Markov chain to predict parking slots occupancy based on the information exchanged among vehicles, which are connected to an ad-hoc network. The obtained predictive error at 30 minutes (in terms of average deviation of the predicted occupancy) has been in the range of 8%.

In the same thematic area, in [148] a distinction among different sources of data - i.e., parking data, user data, open data, has been presented, thus emphasizing the relevance of open data to provide an independent and sustainable system to search for parking spaces on street. Thus, [148] proposed a prediction model based on using neural network presenting an MSE of 16% without addressing the critical situations of parking spaces with non-stationary attitudes.

In [4] the use of car parking data, pedestrian data and car traffic data has been investigated to predict available on-street car parking in 15-minute intervals in the city of Melbourne. On the other hand, addressing the prediction of free slots in parking garages/silos is a completely different problem with respect to the street parking prediction. In parking lots, the number of offered slots is typically high, and clearly reported at the entrance gate of the garage, and therefore they have a strong appeal to drivers who may arrive all together. They are typically located closer to center of attraction such as commercial centers, hospitals, railway stations, theaters, and multiservice areas, where large events may rapidly overstock the structure. Therefore, the prediction of free/available parking slots in garages is not an easy task. Some of them may have a stable stationary behavior over time (due to the served facilities, such as suburbs hospitals), thus making the job prediction easier. Others may be affected by several factors, which makes the prediction of free/available slots over time much more difficult, especially during critical situation when the parking becomes crammed. Furthermore, the data related to garages are not easily available or they can be available only against payment or when one arrives at the entrance. Therefore, according to [203], the number of available lots of a garage may depend on road traffic flow, weather events, road condition, etc., and the best prediction of available spaces is a combination of short time data and historical information. Thus, a neural network model can be used to predict the available spaces of fourteen garages in Beijing and yet the prediction results were not reported. The authors keep on saying that their prediction results showed problems of performance. [177] and [198] have developed an intelligent parking system without predicting the real number of free/available parking spaces. While, in the discrete choice a model for combining online and historical data for real-time to predict the parking availability of a single garage (665 available lots on four levels) has been used in [40]. Thus, achieving an average error in prediction of 1 hour lower than 3% without addressing the critical condition, when available parking lots are close to zero. A more traditional solution was also proposed in [64], predicting availability in one Pittsburg parking garages (totaling 691 parking spaces) using historical and real-time data, by using multilinear regression model. In [197] the authors proposed a queuing model (well-established continuous-time Markov queue) to describe and predict the stochastic occupancy change of parking facility for a single garage in San Francisco, involving historical data occupancy only.

3.3 Forecasting techniques compared

This section provides an overview of the techniques that have been considered and compared, with the aim of creating a solution to predict the number of available/free slots in parking garages. During our research different techniques have been discharged since they did not produce satisfactory results. Among possible techniques, our choice has been focused on the comparison of the most effective solutions, which are:

Bayesian Regularized Artificial Neural Networks, the Support Vector Regression, the Recurrent Neural Network, and the more traditional statistical approach such as Auto-Regressive Integrated Moving Average approach (e.g., ARIMA).

3.3.1 Artificial Neural Network with Bayesian Regularization

The Artificial Neural Network (ANN) is a very popular technique which relies on supervised learning. Beginning with the very first proponents, they have used ANN as powerful nonlinear regression techniques inspired by theories on how human brain works [27]. The primary application of neural networks involves the development of predictive models to forecast future values of a particular response variable from a given set of independent variables; resulting particularly useful in coping with problems showing a complex relationship between input and output variables. The outcome is modeled by an intermediary set of unobserved variables (hidden neurons), that are typically linear combinations of the original predictors. The connection among neurons in each layer is called 'a link'. A link is stored as a weighted value, which provides a measure of the connection between two nodes, as shown in [76] and [82]. The supervised learning step changes these weights in order to reduce the chosen error function, generally mean squared error, in order to optimize the network for use on unknown samples. ANNs tend to overfit, which means to have trained the NN to fit the noise trend, but without producing a good generalization, as expected by the ANN. However, Bayesian Regularized ANNs (BRANNs) try to overcome the overfitting problem by incorporating Bayes' modeling into the regularization scheme [37]. In general, the overfitting risk increases when a neural network grows in size through additional hidden layer neurons. BRANN approach avoids the overfitting because the regularization pushes unnecessary weights towards zero. The BRANN method is more robust, parsimonious, and efficient than classical ANNs, and the network weights are typically more significant in modeling the phenomena [37]. The BRANN model fits a three-layer neural network as described in [120] and [69]. The layer weights the network, which is initialized by the Nguyen-Widrow initialization method [136], and thus, the model is given by:

$$y_i = g(x_i) + e_i$$

$$y_i = \sum_{k=1}^{s} w_k g_k \left(b_k + \sum_{j=1}^{p} x_{ij} \beta_j^{[k]} \right) + e_i, i = 1, \ldots, n$$

where:

- $e_i \ N(0, \sigma_e^2)$;
- s is the number of neurons
- w_k is the weight of the $k - th$ neuron, $k = 1, \ldots, s$;
- b_k is a bias for the $k - th$ neuron, $k = 1, \ldots, s$;
- $\beta_j^{[k]}$ is the weight of the $j - th$ input to the net, $j = 1, \ldots, p$;
- $g_k()$ is the activation function: in this case

$$g_k(x) = \frac{e^{2x} - 1}{e^{2x} + 1}$$

The objective function consists of minimizing $F = \alpha E_W + \beta E_D$, where E_W is the sum of squares of network parameters (weight and bias), E_D is the error sum of squares, α and β are the objective function parameters.

3.3.2 Support Vector Regression

The SV (Support Vector) algorithm is a nonlinear generalization of the generalized portrait algorithm developed in Russia in the sixties and further developed for decades [188]. This theory characterizes properties of learning machines which allow the generalization of unseen data, thus obtaining excellent performances in regression and time series prediction applications [60]. In the following, the Support Vector Regression model (SVR) with linear kernel has been adopted as a predictive method. The idea of SVR is based on the computation of a linear regression function

$$f(x) = w^T x + b$$

to a given data set

$$(x_i, y_i)_{i=1}^N$$

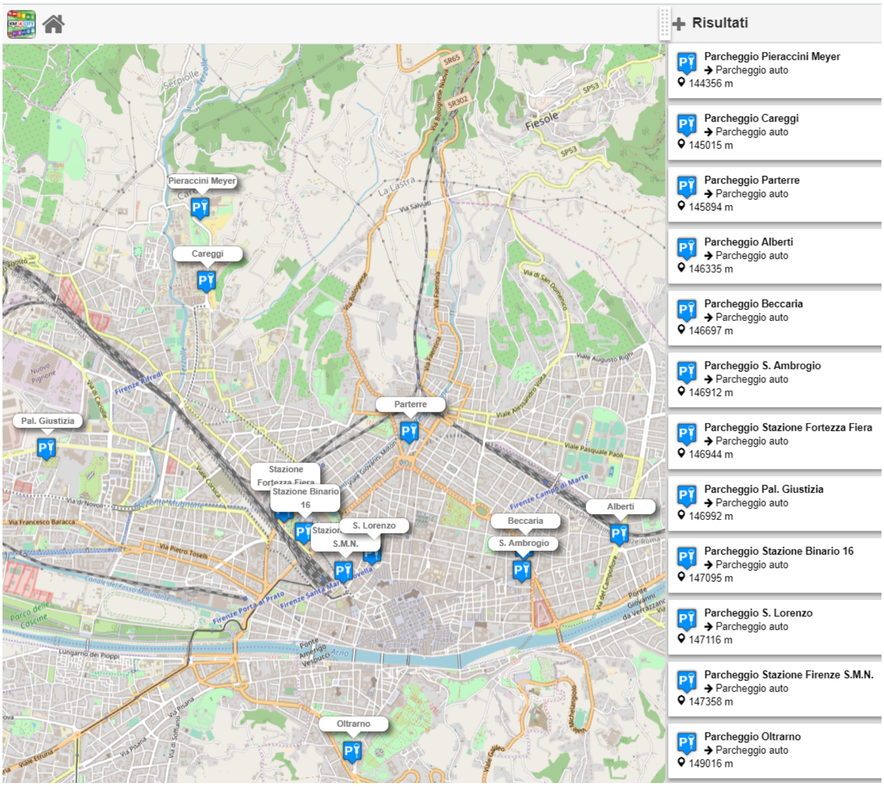
in a high dimensional feature space where the input data are mapped via a nonlinear function. Instead of minimizing the observed training error, SVR attempts to minimize the generalization error bound; so as to achieve generalized performance. The generalization error bound is the combination of the training error and the regularization term controlling the complexity of the hypothesis space [171].

3.3.3 Recurrent Neural Network

Neural Networks have arisen great interest for many decades, due to the desire to understand the brain, and to build learning machines. Recurrent Neural Networks (RNNs) are basically a Feedforward Neural Network with a recurrent loop [53]. They are considered a powerful model for sequential data, and they are applied to a wide variety of problems involving time sequences of events and ordered data. RNN are neural networks consisting of a hidden state h and an output y operating on a sequence of variables $x = (x_1, ..., x_T)$. At each time step t, the hidden state of the RNN is updated by $h_{(t)} = f(h_{(t-1)}, x_t)$, where f is a non-linear activation function. While in principle the recurrent network is a simple and powerful model, in practice, it is hard to train it properly [142].

3.3.4 Arima Models

I have used the Auto Regressive Integrated Moving Average (ARIMA) model as an alternative forecasting method with respect to the above-mentioned techniques. The predictive model has been developed by using Box-Jenkings methodology for ARIMA modeling [30]. ARIMA model is composed of two parts: Auto-Regressive and Moving Average. The Auto-Regressive part (AR) creates the basis of the prediction and can be improved by a Moving Average (MA) modeling for errors made in previous time instants of prediction. The order of ARIMA models is defined by the parameters *(p;d;q)*: p is the order of AR model; d is the degree of differencing, and q is the order of the MA part, respectively; and by the corresponding seasonal counterparts *(P;D;Q)*.



+ Risultati

Parcheggio Pieraccini Meyer
→ Parcheggio auto
144356 m

Parcheggio Careggi
→ Parcheggio auto
145015 m

Parcheggio Parterre
→ Parcheggio auto
145894 m

Parcheggio Alberti
→ Parcheggio auto
146335 m

Parcheggio Beccaria
→ Parcheggio auto
146697 m

Parcheggio S. Ambrogio
→ Parcheggio auto
146912 m

Parcheggio Stazione Fortezza Fiera
→ Parcheggio auto
146944 m

Parcheggio Pal. Giustizia
→ Parcheggio auto
146992 m

Parcheggio Stazione Binario 16
→ Parcheggio auto
147095 m

Parcheggio S. Lorenzo
→ Parcheggio auto
147116 m

Parcheggio Stazione Firenze S.M.N.
→ Parcheggio auto
147358 m

Parcheggio Oltrarno
→ Parcheggio auto
149016 m

Figure 3.1: Map of the main 12 car parks in Florence. As depicted by *Toscana dove, cosa... Km4city* App: `https://www.km4city.org/?app`

3.4 Data description

As mentioned in the introduction, the main goal was to find a solution to predict the number of available parking slots (not occupied) within parking garages controlled by a gate. The Sii-Mobility Km4City infrastructure, collected the data used for the prediction during the period from January 5, 2017, to March 26, 2017. For each car park, the number of available slots has been checked and registered every 15 minutes. Therefore, our study, refers to 12 garages located in the municipality of Florence as depicted in Figure 3.1.

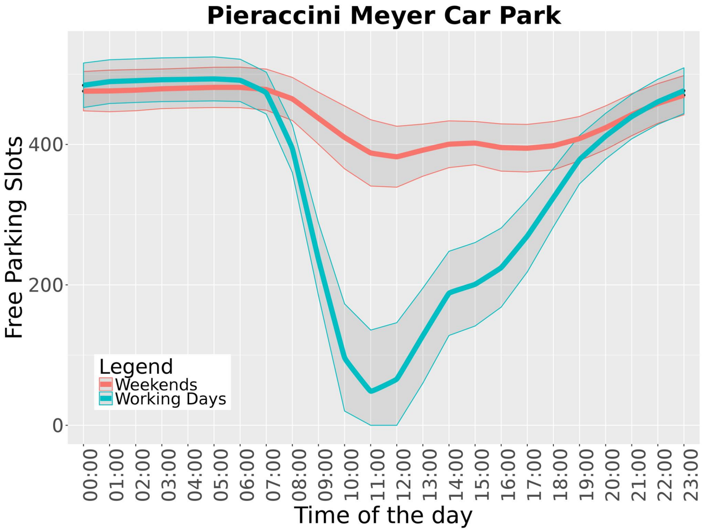
These garages are located in three main different areas of Florence: close to hospitals, downtown (near to the main touristic area) and in the outskirts. The latter are also called park and ride systems, specifically created to stimulate the usage of public transportation. They are meant to provide parking space for commuters deciding to drop their cars out of the city and switch to public transportation.

The above considerations are clearer by analyzing the weekly curves for Careggi and S.Lorenzo car parks (case **(a)** and **(d)** of Fig. 3.2, respectively). In fact, in Fig. 3.2**(a)**, the different trend registered for working day and weekend for the hospital parking is clear. Similarly, the same difference is registered in Figure 3.3**(b)**. Please note that, in both cases, it is possible to observe that Epiphany vacation of the 6th of January, created a trend similar to the weekends. According to [99], it is reasonable to think that changes in patterns between workdays and weekends can be due to different travel purposes: people that usually travel for work on workdays, and for entertainment on weekends [99]. However, considering that Careggi car park is close to the hospital, it's important to remark that car parks have a different trend considering the time windows that the hospitals set up to allow patient visits.

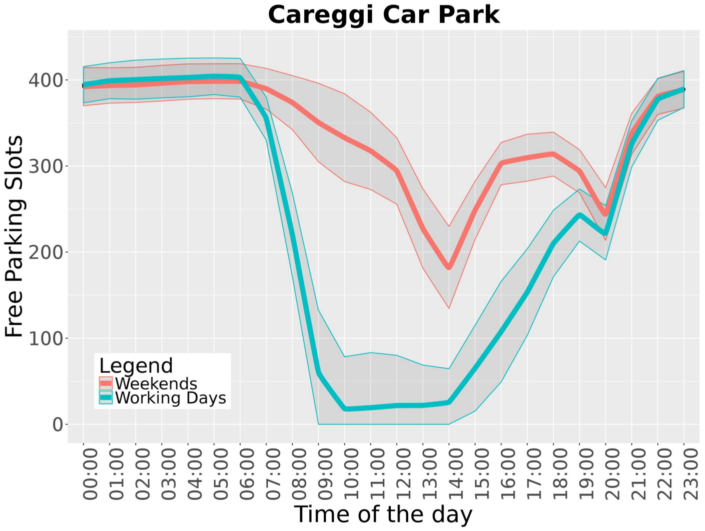
In both daily and weekly curves (see Fig. 3.2 and Fig. 3.3), it is possible to better understand the critical conditions of a garage, i.e., when the available parking slots become close to zero. That is the situation in which the drivers have to be alarmed in advance giving as more precise prediction as possible. The ability of the proposed algorithm to predict when the garage is becoming complete with a significant precision (a small prediction error), together with the capability of handle missing data, can be defined as robustness.

3.5 Features Definition

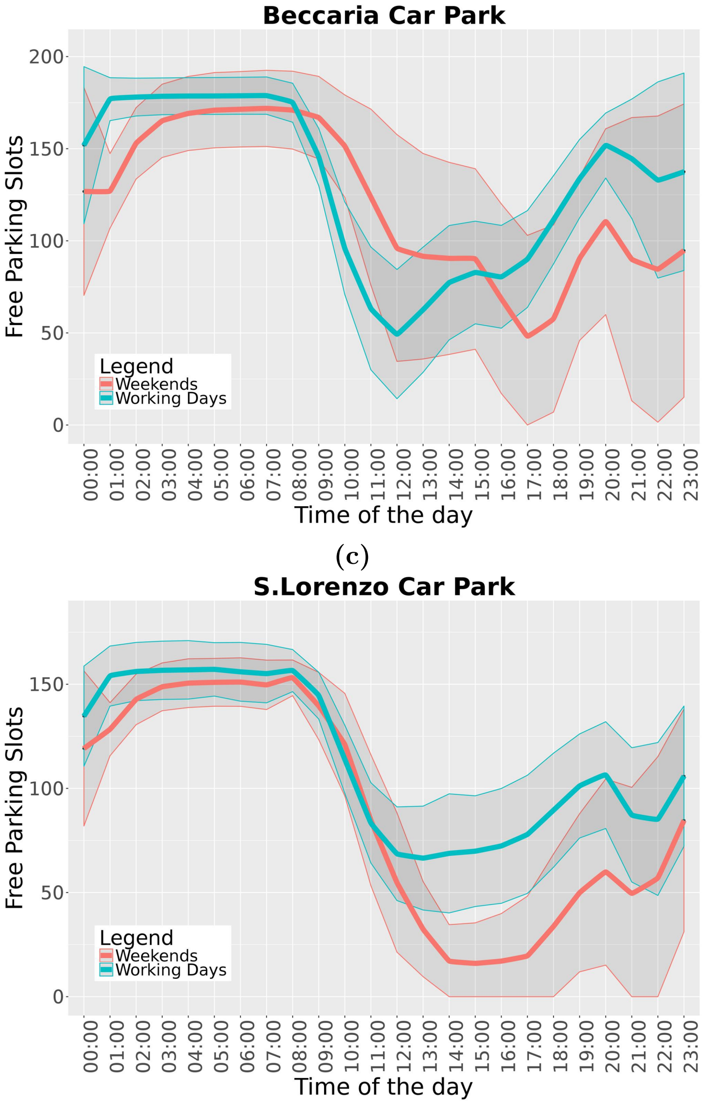
According to the above presented state of the art, there is a substantial difference between a parking garage and a street-parking in terms of distribution of free spaces in the parking area. In the context of street-parking, it may be necessary to make a clustering to understand the free space distribution of an area; thus, aggregating the street-parking areas with the same behavior.On the opposite hand, taking into account garages, the trend of the available slots is very peculiar of the specific contextual conditions of each garage (as depicted in Fig. 3.2). For this reason, the adoption of clustering


Pieraccini Meyer Car Park
Free Parking Slots
0
200
400
Time of the day
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Legend
Weekends
Working Days

(a)

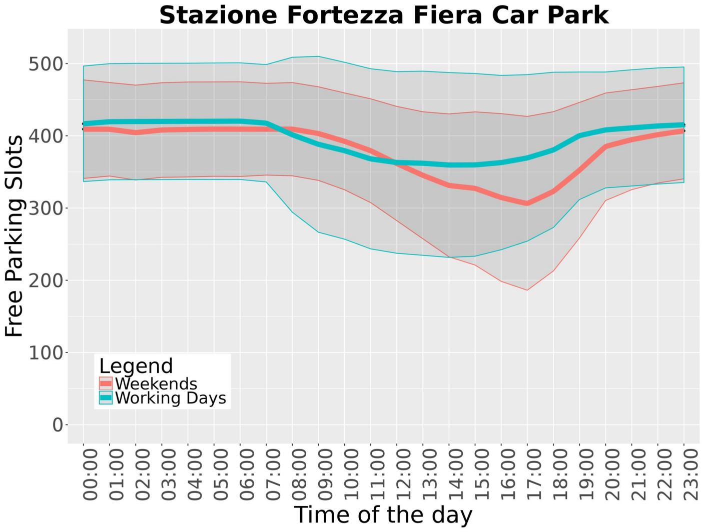

Careggi Car Park
Free Parking Slots
0
100
200
300
400
Time of the day
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Legend
Weekends
Working Days

(b)


Beccaria Car Park
Free Parking Slots
200
150
100
50
0
Legend
Weekends
Working Days
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Time of the day
S.Lorenzo Car Park
Free Parking Slots
150
100
50
0
Legend
Weekends
Working Days
00:00
01:00
02:00
03:00
04:00
05:00
06:00
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00
16:00
17:00
18:00
19:00
20:00
21:00
22:00
23:00
Time of the day

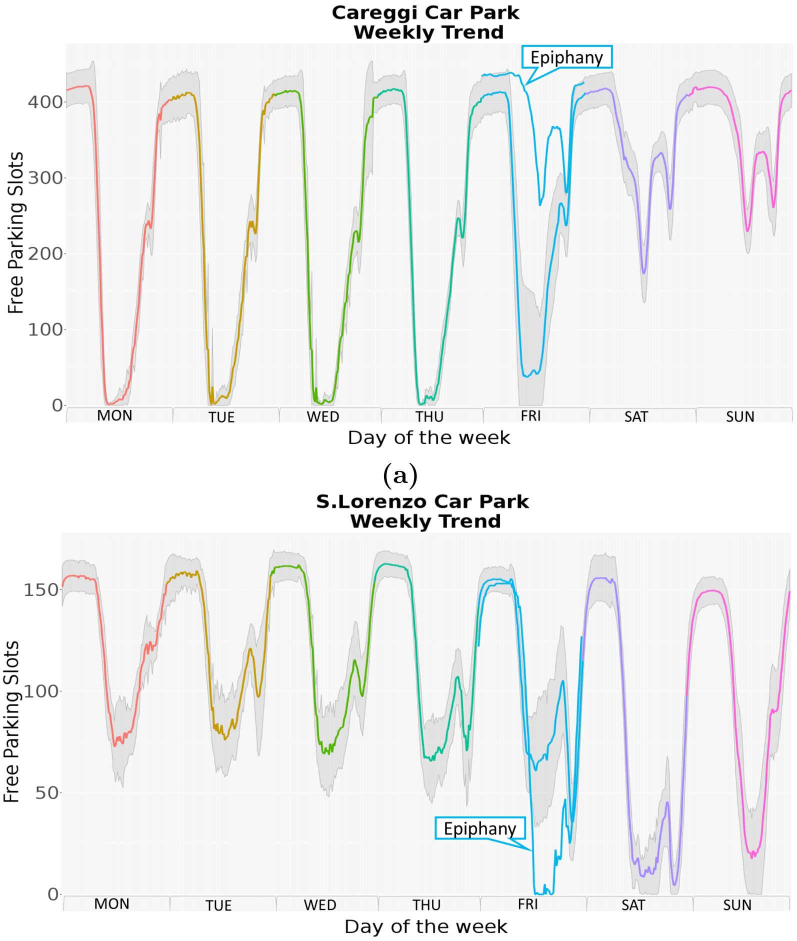
(c)

(d)



(e)

Figure 3.2: Comparison between the actual trend of free parking lots and the predicted trend on the basis of BRANN using baseline features and all features for (a) Pieraccini Meyer, (b) Careggi, (c) Beccaria, (d) S. Lorenzo, (e) Stazione Fortezza Fiera car park for 24 hours.



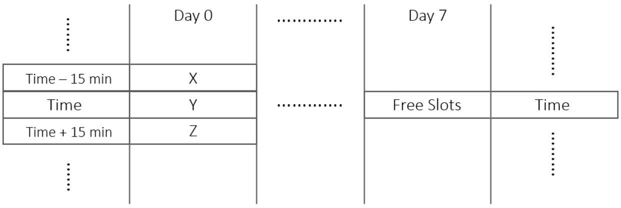
(a)

(b)

Figure 3.3: Weekly curves of free parking slots every 15 minutes (from 05 January 2017 to 26 March 2017) for (a) Careggi car park and (b) S.Lorenzo car park.

approach is not successfully. Moreover, one of the difficulties experienced in identifying a common predictive and precise model for all parking garages, was due to the fact that different parking garages have different behaviors in different days of the week, and period of the day. Some of them may experience critical condition when the available parking slots are close to zero and this is the moment when drivers have to be alerted in advance. According to the above considerations, before evaluating the predictive capabilities of forecasting techniques mentioned in Section III, three groups of features have been identified as possible predictive metrics and are briefly discussed. They have been reported in Table 1. The potential metrics at the basis of the predictive models are discussed in the following beginning with the category they belong to. Features belonging to the Baseline category refer to measures related to the direct statistical observation of garage data and derived information. To this category belong the date and time when measures are taken, working day or not, number of available slots, etc. All the values are recorded every 15 minutes. These variables are used to consider the seasonality of the data which may have different trends - i.e., working days with respect to weekends, etc. When the car parks have the same trend during the same day and time between different weeks, two other features have been included in the model:

- POD: the difference between the actual and previous number of available space at the same time, recorded one week before;
- SOD: the difference between the actual number of parking spaces and the next one at the same time, recorded one week before.



$$POD_{Day7,Time} = X_{Day0} - Y_{Day0}$$
$$SOD_{Day7,Time} = Z_{Day0} - Y_{Day0}$$

Figure 3.4: Construction of Previous observation's difference (POD) and Subsequent observation's difference (SOD) features described in Table 3.1.

Please see Fig. 3.4, at a specific observation of a specific date and time corresponds the POD and SOD of the previous week. Features belonging to the weather are also collected every 15 minutes (i.e., temperature, humidity and rainfall). According to our analysis, the significant values are those related to the hour before any parking time. Therefore, in order to predict the number of available spaces in a garage at 3 pm, the weather features at 2 pm are relevant. In fact, the weather conditions typically influence the decisions on using the car or the public transportation. For example, the expected behavior of citizens when it rains, is to drop the motorcycle and drive a car. By doing so, more parking lots will be taken. On this line, you would suppose to exploit long term weather forecast (6 hours or days in advance) since they could also influence decisions (weather forecasts are accessible on the Km4City smart city). On the contrary, according to our experiments, the weather forecast features are less significant with respect to the real weather features, and thus have been not reported in the table of relevant features. In Table 1, the features and data belonging to Traffic Sensors refer to the values of traffic recorded by the sensors which are located nearby the garage, and mainly on the streets leading to the garage (the distance of influence depends on the density of the city; in Florence case, over 400 meters they are marginally influencing the prediction). These traffic sensors' values are relevant if available for the previous hour with respect to the time of prediction. As described in Table 1, typical values are related to vehicle flow, concentration, average time and average speed. They are estimated every 15 minutes. The metrics adopted for traffic flow estimation are typically the ones accessible from city traffic flow sensors. In this context, the value of traffic flow

Table 3.1: Overview of Features that can be used to describe the context of parking usage with their: category, features and description.

Category	Features	Description
Baseline	Free Parking Slots	Real number of free slots recorded every 15 minutes
	Time	Hours and minutes
	Month	Month of the year (1-12)
	Day	Day of the month (1-31)
	Day Week	Day of the week (0-6)
	Weekend	0 for working days, 1 else
	Previous observation's difference (POD)	Difference between the number of free spaces at time i and number of free spaces at time ($i - 15$ minutes) recorded in the previous week
	Subsequent observation's difference (SOD)	Difference between the number of free spaces at time i, and the number of free spaces at time (i+15 minutes) recorded in the previous week
Weather	Temperature	City temperature measured one hour earlier than Time, in °C
	Humidity	City humidity, measured one hour earlier than Time, in %
	Rainfall	City rain, measured one hour earlier than Time, in mm
Traffic Sensors	Average Vehicle Speed	Average speed of vehicles on the road closest to the parking, over one-hour period (km/h)
	Vehicle Flow	Number of vehicles passed closest to the parking, over one-hour period
	Average Vehicle Time	Average of distance between vehicles, over one-hour period
	Vehicle Concentration	Number of vehicles per kilometer, over one-hour period

is used for assessing the traffic conditions, and thus the average values are satisfactory. On the other hand, as to other applications, such as routing path finding, more precise data and predictions should be used [199] [56]. The traffic sensors which are relevant for each garage may be one or more and they should be chosen taking into account the direction of travel and the most likely route leading to the garage. Traffic sensors are also used as detectors to identify the occurrence of relevant events such as those of Fig. 3.2(e), even if unexpected. An option could be to perform a specific solution able to take into account any planned event. On Sii-Mobility, also the list of the city major events and their GPS coordinates is available. This approach fails to address precise predictions in the event of unplanned occurrences.

3.6 Results

According to the data and considerations reported, the identified challenge was to create a model and tools to predict the number of available parking slots in the garages with a resolution of 15 minutes for the next 24 hours. As a training data set a sample of three months, from January 5, 2017, to the day before the one when observations were carried out, have been selected in the analyzed test set. The test set is made of 96 daily observations (every 15 minutes) recorded during the weeks from March 27th (Monday) to April 2nd (Sunday), i.e., seven test sets were considered to calculate the error on the one-day prediction avoiding noise. Note that, in the real-time application the model has been trained once a day, providing predictive models with 96 values. Predictions are displayed on the Km4City applications in Florence and whole Tuscany region, for one hour in advance every 15 minutes.

3.6.1 Error Measurement Definition

In the literature, most researchers have adopted the MAPE or MSE in order to calculate the prediction error. The identification of the model for measuring the error is very relevant, since it has to work well, even when close to zero. This is related to the particular issue of street-parking predictions where critical cases occur when the available parking slots are close to zero. Measures based on percentage errors (e.g., MAPE) have the disadvantage of becoming infinite or undefined when the observed value is equal to zero. However, as to garage parking prediction, the possibility of reaching zero lots is part of the problem as depicted in Fig. 3.2. They are full every day for several hours (that is, the feature recording the number of available parking lots assumes values very often close to zero). For this reason, the Mean Absolute Scaled Error (MASE) by Hyndman and Koehler [95] has been chosen. The Mean Absolute Scaled Error is calculated as follows

$$MASE = mean(|q_t|)$$

and

$$q_t = \frac{obs_t - pred_t}{\frac{1}{n-1}\sum_{i=2}^{n}|obs_i - obs_{i-1}|}$$

where:

- obs_t = observation at time t,
- $pred_t$ = prediction at time t,
- n is the number of the values predicted over all test sets (96 daily observations per 7 days)

Note that, MASE is clearly independent on the scale of the data. When MASE is used to compare predictive models, the best model is the one presenting the smaller MASE. MASE can be used as measure to define the robustness of the proposed approach. In this case, robustness means the ability of an algorithm to produce quite reliable results in the event of critical cases (e.g., when the number of free parking lots is zero, and/or in the event of missing data in the stream of observations. For this reason, apart from MASE daily prediction, the MASE related to night, morning, afternoon and evening have been calculated.

3.6.2 Kalman Filtering Imputation of Missing Data

One of the main problems related to the robustness of a possible approach to predict the number of free slots, lies in its capability of producing good results in critical conditions - e.g., when slots are close to zero and/or when the data stream of observations is not providing every data continuously. In most predictive algorithms the lack of some observations could become a problem to produce good results in terms of MASE: for example, if the data related to the traffic volume within the selected park area are missing, the prediction error could become higher, as it is based only on weather data and historic data. To overcome this problem, a Kalman Filter [196] has been used for the imputation of missing data in real time. This solution, together with the capability of the model to be precise in terms of Mean Absolute Scaled Error (MASE), has turned out to be robust especially when slots are close to zero. The data from traffic sensors are the most prone to missing data: this can be due to sporadic and discontinuous malfunctions of sensors or network connection. To avoid any consequent prediction error increase, data have been imputed through the Kalman Filter approach. The system can identify not only the sporadic faults of data, but also the faults which need to be recovered only with extraordinary maintenance. In this case, the algorithm is able to provide the guess by using the historical data, weather data and the remaining real time data. In a regular situation, missing data are about 5% of the entire training set.

3.6.3 Prediction Model Results

In the general framework, four different approaches were tested - i.e., BRANN, SVR, RNN and ARIMA model - applied on the features presented above. In detail, the number of input neurons in BRANN model corresponds to the number of the features reported in Table 3.1. Note that, all features are considered as an individual neuron, except Time which has 96 neurons, one for each slot of 15m (‘00:00’, ‘00:15’, ... , ‘23:45’), while a single output neuron represents the predicted value. The model fits a three-layer neural network with three intermediate neurons - i.e., the number of neurons corresponding to the lowest error rate [208]. The processing time comparison, among the models considered above, is also relevant and it is reported in Table 3.3 for each parking garage. Table 3.2 shows that all the approaches can produce predictions every hour for the next hour in a quite small average estimation time. On one hand, in order to produce satisfactory predictions, the ARIMA approach needs to re-compute the training every hour. This is a quite expensive cost of about 9s for each car park. On the other hand, BRANN, SVR and RNN allow their being ‘trained’ once a day, providing predictive models with 96 values in advance with quite precise results. For this reason, the ARIMA solution has been discharged as performed by other researchers in the literature, as reported in Section 1. Note that, the identified ARIMA was $(5; 1; 2)x(1; 0; 1)$ and allowed to perform short-term predictions with a MASE of about 1.2. Our aim was, not only to find a satisfactory solution to make predictions computationally viable and able to suit for several cases, but also to produce satisfactory results in terms of precision in the context of the critical cases discussed before. As a further step, the comparison has been focused by considering BRANN, SVR and RNN on the whole set of car parks in Florence. As a result, Table 3.3 reports the predictive capabilities obtained for reference cases of Fig. 3.1. Table 3.3 reports the comparison in terms of MASE over the predicted week, and a specific MASE estimated for morning, afternoon, evening and night, for each of the predicted numbers of free parking lots. The comparison of the predictive models has been estimated on a training period of 3 months, considering only the features belonging to the baseline category.

Table 3.2: Comparison among model processing time in training and estimation for a single garage.

Training	**Forecasting Techniques**		
	BRNN	**SVR**	**ARIMA**
Average training processing time (sec)	76.3	9.1	9.2
Re-Training frequency	Daily	Daily	Hourly
Training period	3 Months	3 Months	3 Months
Estimation	**BRNN**	**SVR**	**ARIMA**
Average estimation processing time (sec)	0.0031	0.0052	0.0015
Estimation frequency	Hourly	Hourly	Hourly
Estimation length	1 Hour	1 Hour	1 Hour

MASE has been estimated on a testing period of 1 week after the 27th of March for (a) Careggi, (b) Pieraccini Meyer, (c) S.Lorenzo, and (d) Beccaria car parks. This comparison has highlighted that BRANN approach achieved the most reliable results, especially in critical time slots, where the car parking garages risk being full. This fact is also highlighted by the best MASE for BRANN in all reference cases.

An additional analysis has been performed in order to identify the set of combination of feature categories expected to produce the best predictions (see 3.4). The combinations of features have considered: baseline features; baseline and weather features; baseline and traffic sensors; baseline, weather,

Table 3.3: Comparison among predictive models using the features belonging to the baseline category. Darker cells are those showing better values.

Comparison Error	**Forecasting Techniques**	
	BRNN	**SVR**
Careggi		
MASE Night	34.85	**16.29**
MASE Morning	**0.76**	1.42
MASE Afternoon	**1.89**	4.34
MASE Evening	1.99	**1.51**
MASE Daily	**1.87**	2.34
Pieraccini Meyer		
MASE Night	**6.08**	12.83
MASE Morning	**0.86**	1.27
MASE Afternoon	**1.87**	2.91
MASE Evening	**1.36**	1.57
MASE Daily	**1.37**	2.06
San Lorenzo		
MASE Night	**10.33**	11.81
MASE Morning	2.13	**1.91**
MASE Afternoon	**2.70**	3.15
MASE Evening	**2.15**	3.09
MASE Daily	**2.72**	3.21
Beccaria		
MASE Night	9.32	**7.80**
MASE Morning	**0.95**	1.25
MASE Afternoon	2.49	**2.14**
MASE Evening	**2.96**	4.75
MASE Daily	**2.13**	2.67

and traffic sensors features together. The comparison has been performed by both using the BRANN model which turned out to be the one better ranked and estimating R-squared, RMSE, and MASE. As it can be observed from 3.4, the differences among cases are not very relevant. Results suggest that the best choice in terms of precision is still to use a model exploiting

Table 3.4: The results of BRNN model training in terms of R-squared, RMSE and the estimated prediction error MASE for (a) Careggi, (b) Beccaria car parks.

Model Features	**BRANN Model Results**		
	R-squared	**RMSE**	**MASE**
Careggi			
Baseline	**0.974**	**24**	1.87
Baseline + Weather	0.975	**24**	**1.75**
Baseline + Traffic sensors	0.975	**24**	2.04
Baseline + Weather + Traffic Sensors	0.975	**24**	1.87
Beccaria			
Baseline	**0.888**	16	**2.13**
Baseline + Weather	0.890	**15**	2.15
Baseline + Traffic sensors	0.892	16	2.24
Baseline + Weather + Traffic Sensors	0.895	16	2.33

the baseline only. However, extending the assessment to all parking garages results are substantially different as discussed in the sequel.

In fact, while extending the assessment to all parking garages, Fig. 3.5 reports the comparison of the 4 models as compared in Fig.3.4 but related to all garages according to the estimation of MASE for the last week. The comparison stresses that in the cases when the daily trend of available slots:

- is regular (such as cases (a) and (b) of Fig. 3.2, Careggi or Pieraccini Meyer), the 4 models of Table 3.4 are not so much different in terms of result quality.

- presents non-stationary critical conditions (such as Case (e) of Fig. 3.2, Stazione Fortezza Fiera, Palazzo di Giustizia, and other as Parterre), the best model turned out to be the one considering both weather and traffic sensor features together with baseline.

For example, Fig. 3.6 presents the typical comparison of the real daily trend with respect to the prediction using: (i) baseline features only, (iii) the combination of baseline, weather and traffic sensors features, for Careggi

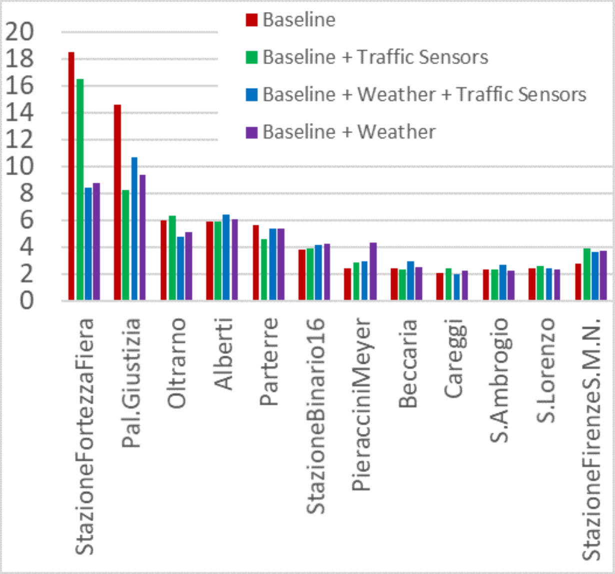
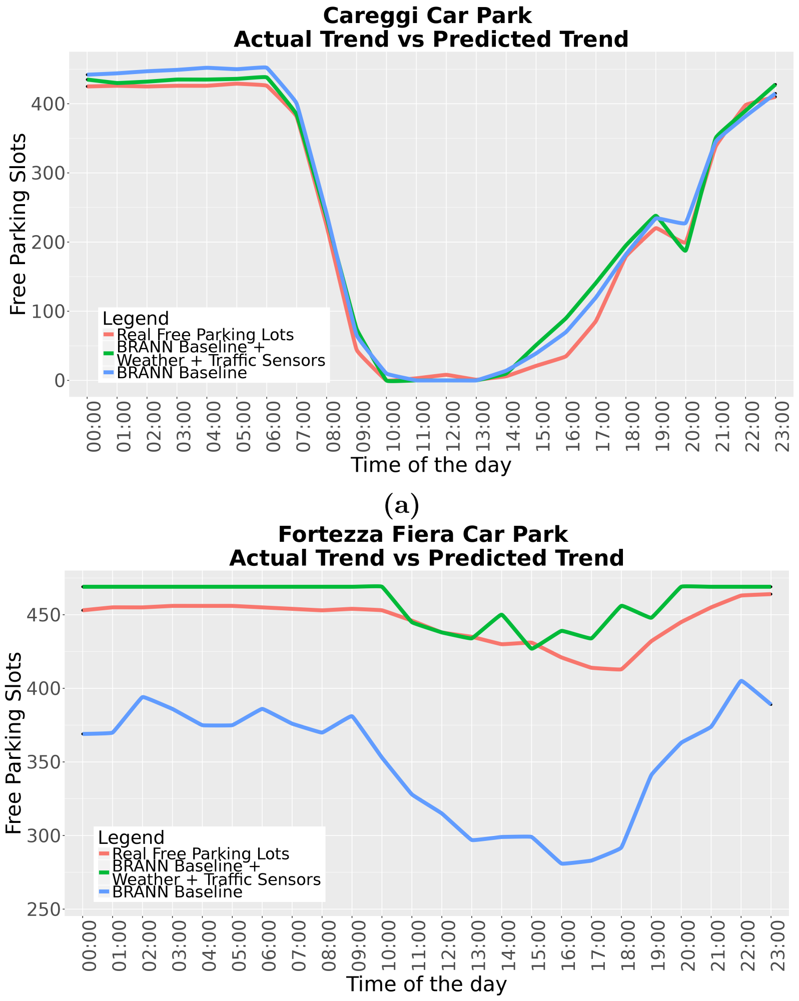


Figure 3.5: Comparison of the predictive models applied to the 12 garages in Florence in terms of MASE assessed in the last week of predictions.

and Stazione Fortezza Fiera car parks. Noteworthy is that the addition of weather and traffic sensor features decreases the mean difference between the real values and the predictions in the Stazione Fortezza Fiera car park.

To highlight the above presented results, Fig. 3.7 reports the analysis of importance for the features listed in Table 3.1. They are listed in order of relevance for the BRANN full model prediction - i.e., the model with all the categories of covariates (the relevance assesses the relationship between each predictor and the outcome is evaluated). In particular, the importance of each predictor is evaluated individually: during the BRNN model training, a LOESS [52] smoother, (i.e., a nonparametric method for regression estimation) is fitted between the outcome and the predictor. To obtain a relative measure of variable importance, the R^2 statistic is calculated for the model containing the considered variables against the null model (intercept only). The resulting histogram depicts that variable Time (of the baseline) is the most relevant to predict the number of free slots for all garages. The second in terms of relevance turned out to be Average Vehicle Time of the traffic sensors features. According to these results, traffic variables are of



(a)

(b)

Figure 3.6: Comparison between the actual trend of free parking lots and the predicted trend according to BRANN using baseline features and all features for **(a)** Careggi, **(b)** Stazione Fortezza Fiera car parks along 24 hours.

primary importance, as already mentioned in [203]. These statements seem to be quite coherent with the finding of [148] for street parking. On the other hand, in making predictions for garages (as in our case) it is easier to

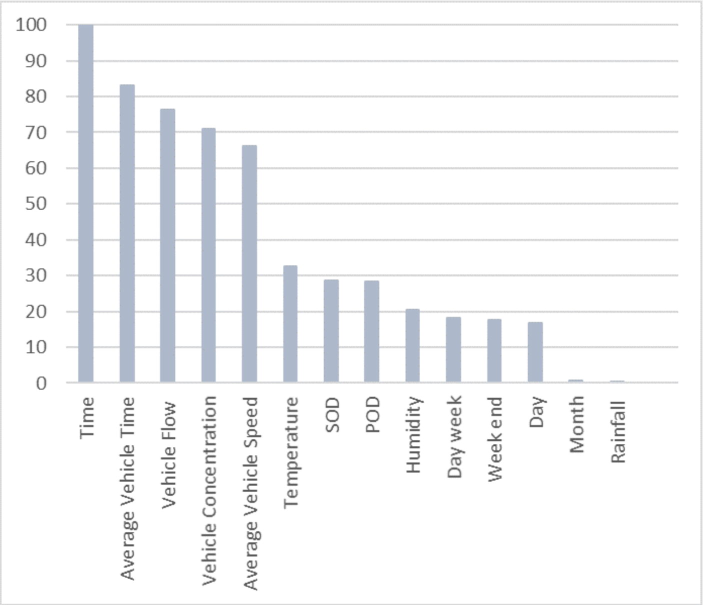


Figure 3.7: Variables Importance of the BRANN full model.

choose traffic sensors related to the car park under investigation - i.e., only the sensors on the streets leading from the path to the garage. Whereas, in street-parking prediction, only the general traffic situation may be of interest. The selection of suitable sensors can be performed only in the cases where data are publicly available, as emphasized in [148]. As a general consideration, solutions to predict available slots in garages in current state of the art, are based only on baseline and stationary conditions [198], [39]. In our case, it's have been demonstrated that exploiting classic historical garage parking data together with traffic and weather features has produced better predictions.

Finally, our results cannot be directly comparable in terms of prediction errors, as just the precision in the event of critical cases has been analyzed. These conditions have not yet been addressed in literature before. Please note that when the parking slots are close to zero, measures based on MAPE and MSE have the disadvantage of being infinite or undefined. For this reason, MASE was the best choice, resulting to be 1.75 in the best case.

3.7 Considerations

Looking for available parking slots is a serious issue in today urban sustainable mobility. The solution can be to provide suggestions to drivers about the parking availability. Suggestions should reach drivers 30 minutes and 1 hour in advance (thus producing a precise time stamp of which time they refer to) to allow their conscious decision-making process. To this end, reliable prediction models are needed. Prediction of available parking spaces is a complex non-linear process involving multiple kinds of factors, as the variety of parking area (downtown, nearby hospital and others on the outskirts, close to theaters, airports, etc.). In fact, a critical factor is the different trend of each garage: provided the aim is to cover a higher number of garages, the precision of the prediction is relevant, especially in critical cases (full garage). In almost all predictive models, the historical data, traffic flow sensors and weather data have demonstrated high predictive capabilities in explaining the number of free parking slots. In parking garages without a recurrent daily trend of available slots, traffic sensors and weather covariates have improved the precision in predicting. The entire approach can be considered flexible, robust to critical cases and robust to sporadic lack of data. The research documented in this Chapter has demonstrated that a Bayesian Regularized Neural Network exploiting historical data, weather condition and traffic flow can be a robust approach for reliable and fast estimation of available slots predictions. The predictive model can produce predictions 24 hours in advance, while they are provided on mobile applications, 30 minutes, 1 hour in advance directly, and if requested also a day in advance as possible general trend.

Chapter 4

Users' Transportation Modality Classification

In this chapter, a number of metrics has been identified in order to understand whether an individual on the move is stationary, walking, on a motorized private or public transport, with the aim of delivering to city users personalized assistance messages for sustainable mobility, health, and/or for a better and enjoyable life, etc. Differently from the state of the art solutions, the proposed approach has been designed to provide results, and thus collect metrics, in real operating conditions (imposed on the mobile devices as: a range of different devices kinds, operating system constraints managing Applications, active battery consumption manager, etc.).[1][2]

4.1 Introduction

With the complete digitalization of the public and private transportation networks, the capability of understanding the users' behavior and the mean

[1] This chapter is under review as "Automated Classification of Users' Transportation Modality in Real Conditions" in *Mobile Networks and Applications.*

[2] *Acknowledgments:* The authors would like to thank the MIUR Smart City national founding and to the University of Florence and companies involved for cofounding. Km4City is an open technology of research of DISIT Lab. Thanks to the collaboration of three major public transportation operators in Italy for their support in putting the solution in operative conditions (ATAF, BUSITALIA and CTTNORD).

of transportation have become important. The presence of GPS, accelerometers, sensors on mobile phones has made possible to create solutions exploiting the users' behavior and context. Information from GPS and accelerometers available on mobile phones can be combined with contextual information regarding city and mobility and transport information to assess and predict user behavior. The understanding of user behavior is the first step for providing suggestions and assistance to people on the move via mobile phones. For example, to push them in taking more virtuous behavior, consume less energy, making more sustainable their transportation, having a healthier life walking more, saving money parking closer. The research addressed in this article aims to understand the users' mean of traveling taking into account contextual data and data coming from the phones. The correct classification of transportation means can be also used for providing suggestions in the context of public or private transportation. For example, it may have sense to suggest getting down the bus at the next stop to walk a bit, or suggest parking the car in different place, respectively, using public transportation instead of the private one. Thus, the above described problem is reconducted to the classification problem of the transportation modality/mean (car, bus, walk, bike, etc.), exploiting real time data coming from the devices and contextual information. Please note that, the contextual data are strongly different in different part of the city, and also change over time, for example busses have different timeline and paths: so that users are moving in the real space.

As described in the following section of related works, the problem of understanding the mean of traveling of users has been many times addressed, but not working in real operating conditions. Most of them, assume data collected from the mobile phones with high rates and high precision, identifying models only taking data in strongly controlled conditions: such as limited number of device type, limited number of users and directly engaged to keep the mobile app running in foreground, etc. This means that those solutions have note addressed real operating conditions. In the presence of real conditions, (i) data are sporadic, (ii) the rate is low and not constant, (iii) the quality of data is not uniform since sensors of different mobile phones have different precision and response, (iv) operating system may push the App in background and may apply energy saving rules to applications. These are just examples of the complexity of understanding the mean of transportation in real operating conditions. This also means that both data and methods have to be completely re-elaborated in the latter case.

4.2 State of the art

The problem of classifying users' mean of traveling has been addressed by a number of approaches in different research areas [155]: *Location Based Services* (LBS), *Transportation Services* (TSc) and *Human Geography* (HG). The LBS solutions aim to understand the transportation meaning in real-time to provide useful information to the user whenever he/she asks. Note that, in LBS approaches the velocity of response has been privileged with respect to the correct segmentation of a trajectory. On the contrary, in the TSc approaches, the correct segmentation of a trajectory is privileged with respect to velocity of response: TSc solutions aim at generating reliable statistics and activity-travel diaries about the transportation mode of a user when performs daily activities [26], [174]. The HG approaches focus on the segmentation of a trajectory into parts with domain-specific semantics: it is common to first split trajectories into segments where the object is stationary or moving.

On the other hand, the aim of the research presented in this article consists of developing a solution to help users during their traveling in real time, LBS solutions have been better analyzed and reported. In LBS, the transportation means' classification is regarded as an online process: an algorithm that provides the current transportation mode of the user in real-time or quasi real-time. Moreover, those algorithms should generate information that can help to understand better how the user is traveling. To this end, different types of data/sensors have been exploited: GPS [173], accelerometer [91], [205], [192] and the combination of GPS and accelerometer [159], [123] [164]. On this topic, [173] have compared five different models using data collected from GPS classifying the users' traveling means in six categories (walk, train, driving, stationary, bus, bike). Their work exploits a transportation network data set with FGIS information together with the real time positions of buses and trains, and they have identified a set of seven features (average accuracy of GPS, average speed, average heading change, average acceleration, average bus closeness combined with candidate bus closeness, rail line trajectories closeness). [173] have used the GPS position sampled every 15s and a window frame of 30s, presenting 92.8% of accuracy by using the Random Forest algorithm. Please note that, today, with the high attention to power safe of the present mobile devices and operating systems, 15s of stable sampling rate is somehow realistic only if the application providing data is in foreground as a navigator, while on all the other cases, is not, so that it is not realistic today, since if the user is walking, moving on in bus or train do not use the navigator, obviously.

[91] have proposed a study that involves only accelerometer data. They have obtained an 80.1% accuracy and an 82.1% recall for seven transportation modes, by using both AdaBoost and Decision Tree (two-stages classification). The authors have collected 150 hours from 16 users considering 3 different devices with accelerometer sampler at 0.01s (100Hz) of frequency. On the same line, [205] have extracted 22 features in the time-domain and 8 in frequency-domain for the evaluation of five classes of transportation modes as motorcycle, car, bus, tram, train and high-speed rail. [205] have compared three different classifiers (Decision Tree obtaining an 84.81% average accuracy, AdaBoost with a 87.16% average accuracy, and SVMs with a 90.66% average accuracy). About 8311 hours of data (100GB) have been used for the learning process, with a sampling rate of sensors of 0.03 sec (30Hz), collected using a single device (HTC One mobile with Android system). [192] have considered a small data-set of 12 hours (5544 samples of six transportation modes) from 7 different users, obtaining a 70% accuracy with a Decision Tree algorithm. [159] have demonstrated that, taking into account of both GPS and accelerometer the accuracy can be improved. The authors have considered five different transportation modes (still, walk, run, bike and motor), gathering 15 minutes of data for each transportation modes (six terminals per person, 1 type of device), and the total amount of data collected across all 16 individuals was 120 hours (a small data set, produced in controlled conditions). They have achieved a 93.6% precision using a combination of Decision Tree and Hidden Markov Model (two-stages classification), with both accelerometer and GPS features involved, using a sampling rate of [164]] made a distinction among different type of non-motorized motion (walking, running, biking), vehicular and random movements, using the accelerometer sensors of mobile. They have used a Decision Tree classifier and have applied a Markov model smoother on top of the output of the Decision Tree. Thus, obtaining a 91.53% precision, based on a small data set, produced in controlled conditions: 50 hours of collected data from 15 distinct individuals (with android devices), an accelerometer sampling rate of 0.01 sec (100Hz) and a gps sampling rate of 5s. On the same idea, [123] have

trained a Decision Tree classification model obtaining an 82.14% accuracy (with a gps and accelerometer sampling frequency rate of 1s and 0.04s respectively), while [154] have obtained a 90.8% accuracy using a Random Forest algorithm on seven different modes of transportation (walk, bike, car, bus, subway, train, ferry), both using a single device. [17] combined GIS features and speed with socio-demographic characteristics and travelers personal preferences to facilitate better transportation modalities detection (stationary, walk, bus/car, rail). The authors focused their research on mobility-affecting disabilities users, achieving an overall a 78% accuracy with the Random Forest classifier.

Recently, [9], have proposed a two-layer hierarchical classifier to predict five classes of transportation mode (car, bus, walk, run, bike), achieving a 97% accuracy. They state that a hierarchical approach can increase the accuracy with respect to the traditional classification algorithms. As first step, they applied a multi-class classifier to select the two transportation modes with higher probability, given the test data. As second step, they used a binary classifier to discriminate between the chosen pair: the binary classifier is specialized in the pair of modes identified in the first step and uses a specific feature subset. [9] have used the GPS, accelerometer, gyroscope and rotation vector sensors, sampling the data with the highest possible frequency, during the workdays and working hours, only using two different devices. In particular, an accelerometer frequency rate of 0.01s (100Hz) and a gps frequency rate of 0.04 sec (25Hz) were applied. [160], evaluated the transportation mode detection through a three-steps algorithm: a segmentation, a fuzzy rule transport mode detection and a consistency correction. The authors have achieved a 75% accuracy considering five transportation modes, walk, bike, bus, car and train, tracking users position at least every 4s without information about public transport opportunities. Thus, also in this case, the rate for data acquisition is incredibly high and unrealistic for actual applications (with a regular gps sampling rate of 1s). [204] have presented a Convolutional Neural Networks (CNN) based method to automatically extracting features for the identification of transportation means, thus achieving a 98% accuracy to distinguish between train, bus, car, metro. The authors have conducted the experiment using a single device (Android smartphone) and collecting 200 hours of transportation data and a accelerometer sampling frequency of 0.01s (100Hz). In Table 1, a summary of the state of the art solutions for understanding the travel means is reported. The comparison is putting the attention to a number of aspects that may enable that solution to work on real operating condition or not. And in particular on: (i) the data exploited (both sensors and contextual data if any), (ii) the sampling rates in seconds or sampling per second (regular and/or unregular), (iii) the number of users involved in the training, (iv) the number of features used, some features have to be computed on client and/or server side since the mobile device has not all the contextual information accessible in real time (GIS information), (iv) the number of device types the variability of sensors quality of devices connected with different operating systems and versions, and (v) the precision accuracy obtained. Almost all the state of the art solutions adopted very high rates for GPS data acquisition, with limited number of devices. Thus, the energy consumption and the corresponding network bandwidth for data transmission are issues that are limited by the present mobile operating systems, that apply energy safe rules on all the applications. So that, those solutions are almost unfeasible in real operating conditions. Mobile operating systems allow to keep the high rates (in the order of seconds) only when applications are running in foreground. On the other hand, some of the operating systems, to save energy force the services into the mobile App to sleep and wake up only after few minutes.

4.2.1 Research aim

The research aim has been to realize a solution overcoming the previous solutions at the state of the art to classify the transportation modes to deliver at the users, personalized assistance messages for:

- sustainable mobility, to incentivize ecological transportation choices, suggesting alternative public mean of transport (bus/tram) instead of the private car/motorbike. This is feasible only having a tools for identifying who is taking the private car to with a relevant frequency;

- healthy suggestions, better and enjoyable life, to stimulate users in dedicating a part of their time and moving needs to exercise their body. For example, suggesting getting out of the bus in advance, park in other locations, etc.;

- implementing city strategies to change city user attitudes. For example, to: reduce the number of vehicles in certain areas, to increase the usage of public transportation in specific time slots, to stimulate tourists in taking diverse paths in the city visits, and to select less busy parking areas, etc [11], [13].

With this pourpose, the real-time identification of a private transportation mode (car or motorbike) has a central role in assistance messages delivery. Therefore, according to the above described real operating conditions, the techniques have to produce high classifications accuracy to identify transportation modality of a user, in the presence of (i) large discontinuities samples of data (from sensors and sporadic communications to the central computation modules), (ii) relevant differences which may be due to the different kind of mobile phone features in terms of sensors and precision. The capability of working on real operative conditions, it also mandatory to avoid the App to be identified by the power safe procedures that are nowadays installed into the mobile phones to reduce battery consumption. Therefore, the proposed solution overcome the above mentioned solutions at the state of the art, for the aspects focused on sensor energy consumption factors and real conditions. The solution has been tested on a real application (delivered to the users via official App stores such as Google Play Store, Apple App Store, and accepted by common users, see “Tuscany where what...” on the stores). As described in the following, it is capable to cope with the constrains introduced by terminal manufactures on battery usage for background and foreground services. Moreover, no restrictions on the modality of mobile device usage have been imposed, differently to what has been imposed in the state of the art experiments in which the devices have been asked to keep: (i) the application running in foreground to get more precise GPS data, (ii) the device in a proper position/orientation during the usage; and/or to (iii) use specific devices.

4.3 Architecture and Data Collection

The proposed solution relies on a client-server architecture, where the mobile application can be installed on different operating systems (Android, iOS and Windows devices), with different versions [11]. The sensors' values collected on the mobile device (client-side) are sent to the server that enriches them with additional context information derived data (GIS, geographical information system and knowledge), etc., as described in the sequel. At the same time, the server executes the real time classification algorithm to compute the transportation mean classification for each user. The information is stored on server as a report on the preferred user's travel mean. On this basis, it is possible to set up complex strategies that may be triggered when the user behavior reaches certain specific conditions - i.e., to assist and/or engage the users in their daily activities (even rewarding them, in the cases of virtuous behavior; for example, when a suggestion has been followed). For example, a strategy for stimulating the city users may be based on a firing condition which sends a suggestion to all city users that take their private car to perform the same trip path at least 3 times per week, and at the same time the trip could be easily performed by using public transportation. Thus, the system may inform those city users of the possible alternative, and some of them may follow the suggestion. As a result, by exploiting the user behavior analysis, the solution may detect the acceptance of the suggestion by detecting of change of behavior and may automatically reward the user with a bonus or discount, and deliver congratulations. See [12], on rules and strategies. Figure 4.1 provides a high-level overview of the software architecture and its main components.

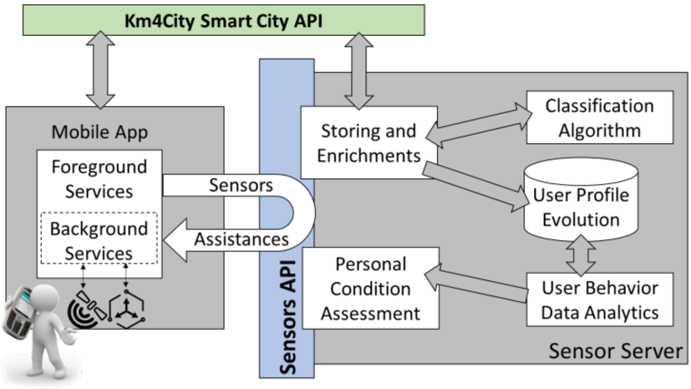


Figure 4.1: System architecture

The **Mobile App** may execute services in background or just in foreground depending on the operating systems. Therefore, different data collection strategies are used on different devices and operating systems. Android devices accept the background processes; thus, the data collection service may work in background remaining always active. In other operating systems, running background services are allowed, therefore, data are collected when the App is in foreground. Both services (foreground and background) follow the same workflow of operations independently on the implementation languages: Java for the background service on Android and JavaScript for other operating systems, since the Mobile App has been developed in Apache Cordova.

The data collected on the mobile applications are called by us as "***Sensor Data Package***" (considering the mobile device as a sensor), and are: (1) positions and movements of the user's device through sensors' and derived information such as GPS latitude and longitude, speed and acceleration; (2) device characteristics as the type of operating system, application version, device model, to provide also statistics on the mobile application usage and to highlight the most widespread configuration; and (3) the user characteristics (language, user profile) to personalize the mobile user experience. The collection of Sensor Data Package aims at gathering a correct GPS position at a given minimum refresh time of 30s, while it is managed by the operating system power safe strategy. As location/position data for the mobile, the GPS position is preferred. On the other hand, some devices and operating systems derive the position from the network connection (cellular connection and/or Wi-Fi hotspots), or by using some fused/mixed strategies, which actually combine all of them. In some cases, the GPS is not available, and this may mean that the user has disabled the location detection, the GPS. Thus, the **Location Measure kind** can be GPS, Network (cellular position or Wi-Fi) or mixt. If, at least one location mode is active, two different cases may occur: (1) it is possible to detect the location of the device at that time (for example, the user is inside a building where the GPS signal fails to retrieve information from satellites) or (2) the detected location is not up to date (for example, a position is obtained by the device, then the user entered inside a building: the position may be the last obtained and sadly it is too old). In both these cases, a position update is required: if the updated value is not accessible, a simply ALIVE message is sent to the server, communicating the other values of the Sensor Data Package. In the absence of a local measure, the accelerometers information could be collected anyway from the device and thus sent to the server.

The Sensor Data Package is collected in a buffer and sent by the mobile application to the server periodically by the active background or foreground services. When the sending of the last packages is successful, the data sent are deleted from mobile device.

In the Figure 4.2, the protocol's operations performed by the background

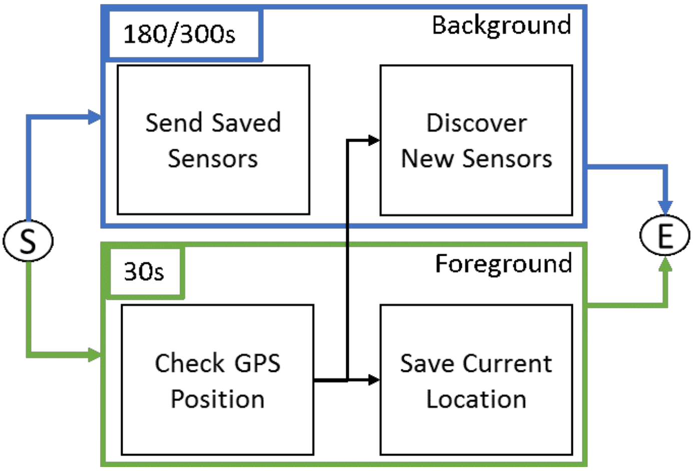


Figure 4.2: Data Recovery Service Timing

service are shown. The block called "Save Current Location" stores (every time the GPS position is available) a list of data used to calculate further features of the terminal movements at a given time, whenever these data are sent to the server-side. Both background and foreground services perform the same operations in their corresponding operative conditions of the App.

The **server-side process** (see Figure 4.1) collects all the data sent by the mobile applications and computes several features for enriching each the single record of data (package) associated with a given timestamp of measure. Among the computed features, the distance/proximity of the GPS coordinated of the mobile with respect to the railway, or bus lines, or highway, cycling path, and/or parking zones. This kind of geographical information is retrieved from the Km4City knowledge base in Big Data technology by using the Smart City API [134], interface to semantically integrated information for applications and services [22]. Other computed features are the average velocity of the last period, etc., as described in the sequel.

Finally, the process on the server puts in execution the classification algorithm to retrieve the user's transportation mean. This information is also stored point by point for further processing, and to eventually produce contextual assistant messages to be delivered to the user's device by some rule editor according to the strategies identified by the municipality or by some city operator [12]. On the other hand, the classification approach of transportation mean has to predict the traveling means for the next time slot, in order to provide suggestions in time and not only with the delay due to the data collection - sending - and computing process.

4.4 Classification techniques compared

This section presents an overview of techniques considered to create a solution for classifying the transportation mean of the users in the move. During the experiments, several unsatisfactory techniques have been tested. Among the possible techniques, only the comparison of the most promising approaches has been presented: Random Forests, Extremely Randomized Trees, Extreme Gradient Boosting, Super Learner and Hierarchical approach. Please note that Classification Trees are machine-learning methods which are used for constructing exploration, description, and prediction models. The usage of Classification Trees approaches, i.e., Extremely Gradient Boosting and Random Forests methods, have potential advantages in the predictive model's construction. Classification Trees are free of distributional assumptions and they can handle different types of responses, such as categorical, numeric, multivariate, censored and dissimilarity matrices; they are invariant to predictors monotonic transformations; the presence of missing values in the predictors are handled with a minimal loss of information. On the basis of the above described properties, the adoption of classification and regression trees - i.e., Extremely Randomized Trees or Random Forest methods which are free of distributional assumptions potentially provide an advantage for the construction of predictive models. As further step, the multi-class problem has been divided in a collection of binary classification problems: they were analyzed using the Super Learner algorithm, combining the different learning techniques above. Moreover, a Hierarchical approach has been proposed and compared with the above approaches based on a single multi-class classifier. Therefore, for completeness, a short overview of the above-mentioned approaches is reported in the next subsections. In Figure 3, a schema of the processes adopted is reported for both classic classifiers (a) and Super Learner (b).

4.4.1 Random Forest

Random forests algorithm has been proposed by Breiman [34] as an improvement of the Tree Bagging approach. In Random Forests, a different bootstrap sample from the original data was used to construct each tree. For each tree of the collection, each split is determined by a randomly chosen subset of predictors; this procedure reduces the correlation between predictors of the individual trees, and each tree has the same expectation.

4.4.2 Extreme Gradient Boosting

Gradient Boosting [73] is a way to reduce the variance, respect to other decision tree methods. The Boosting Trees algorithm is an evolution of the boosting methods application. Boosting methods [72] performs classifications by weighted majority vote, and they have the advantage to fit many trees of different dimensions to reweighed versions of the training data. In Gradient Boosting, small regression/classification trees are built sequentially from the gradient of the previous tree loss function (pseudo-residuals), and in order to produce an incremental improvement in the model, at each iteration, trees are built from random sub-samples of the dataset. Basically, given a modality i with a vector of covariates X_i , a K additive functions is used to predict the output of the tree ensemble model.

$$\hat{y_i} = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F$$

where F is the set of all possible trees. The f_k function maps the value in x_i to a certain output, at each step k. Extreme Gradient Boosting tries to minimize the regularized object as follow:

$$\mathcal{L}(\phi) = \sum_i l(\hat{y_i}, y_i) + \sum_k \Omega(f_k)$$

where

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda||\omega||^2.$$

In the above equation, l is a differentiable complex loss function (Mean Square Error). while the second term penalizes the complexity of the model in terms of number of leaves in tree T and vector of scores on leaves ω to avoid overfitting. In general, boosting procedure outperforms the random forests.

Extreme Gradient Boosting in [48] is an efficient and scalable implementation of the proposed Gradient Boosting framework by Friedman [74].

4.4.3 Extremely Randomized Trees

The Extremely Randomized Trees is a tree-based ensemble method for supervised classification and regression problems. It involves the randomization of both attributes and cut-points choices during the splitting of a tree node. In the extreme case, trees are totally randomized and the structures of them are independent of the learning sample output values. The strength of the randomization can be tuned to problem specifics by the appropriate choice of a parameter [78]. With respect to random forests, the Extremely Randomized Trees idea is to drop the use of learning sample bootstrap copies, and instead of trying to find an optimal cut-point for each one of the K randomly chosen features at each node, it selects a cut-point at random [78]. From a statistical point of view, the idea to drop the bootstrap copies leads to an advantage in terms of bias because the cut-point randomization has an excellent variance reduction effect.

4.4.4 Super Learner

In this case, the multi-class problem can be divided into a binary classification problems collection: considering 4 classes (C_1, C_2, C_3 and C_4), it is possible to adapt binary classifiers for C_1 vs C_2 or C_3 or C_4, C_2 vs C_1 or C_3 or C_4, C_3 vs C_1 or C_2 or C_4 and C_4 vs C_1 or C_2 or C_3, and then combine the results to make a past classification on the highest probability estimate. Note that, the approach presents a certain flexibility for the classifier for the different categories. In general, it is not possible to know a priori which learner will produce the best performance for a given prediction problem [187], [152], and the relative performance of various learners depends on the true-data generating distribution. The idea is to apply a set of candidate learners to the observed data and choose the optimal learner for a given prediction problem by using a cross-validation risk approach. Thus, the learner algorithm with the minimal cross-validation risk is selected. Super Learner performs asymptotically to produce the best possible weighted combinations among the set of candidate learners considered [185], [186]. Therefore, considering $L(n)$ a collection of learners $\hat{\Psi}_l$, $l = 1...L(n)$, in parameter space Ψ. Super Learner is defined as

$$\hat{\Psi}(P_n) \equiv \hat{\Psi}_{\hat{L}(P_n)}(P_n)$$

where $K(\hat{P_n})$ indicate the cross-validation selector.

$K(\hat{P_n})$ selects the best performing learner in term of cross-validated risk:

$$\hat{L}(P_n) \equiv \arg\min_l E_{V_n} \sum_{i,V_n(i)=1} (y_i - \hat{\Psi}_l(P^0_{n,V_n})(X_i))^2$$

In particular, $V_n \in \{0,1\}^n$ indicates a random binary to split the learning data set into a training set $\{i : V_n(i) = 0\}$ and validation set $\{i : V_n(i) = 1\}$. The empirical probability distributions of the validation samples and training samples are denoted by P^1_{n,V_n} and P^0_{n,V_n} respectively [187]. In the performed experiments, the Super Learner has been applied with 10-fold cross-validation to select the optimal learner given the following set of candidates: Random Forest, Gradient Boosting, Extremely Randomized Trees.

4.5 Data and Feature Definition

The features taken into account by the classification algorithm have been selected from a larger set considered during the preliminary analysis and experiments. The process of features reduction has been performed by assessing their relevance in the contexts of the algorithms tested as partially mentioned in Section 3. The aim was to identify the smallest subset of features without reducing significantly the precision of the travel mean's classifications. As a result, Table 4.1 includes the selected metrics and the features, classified in 4 categories, collected from the mobile as **Sensor Data Package**, with those computed from the server-side to be used by the classification algorithm. Some of these features can be used for both users' traveling mean classification, and for creating firing conditions for implementing strategies. In Table 4.1 "Where" can be: "D" when the measure is produced on the Device, and "S" when is computed on server-side. Each measure is collected/referred at a given **Day and Time**, and from this value can be easily derived from the device or from server if the day is a working day or not (**Non-Working Day**). The same approach can be followed to detecting the **Time Slot** in which the measure has been collected. The Time Slot strongly influences the attitude of the city users to move by using different means.

Table 4.1: Overview of Sensor Data Package feature measured at a given time from the mobile or computed on server-side.

Category	Metrics	Description	Where
Day/Time Baseline and GPS	Day and Time	Day and Time of the sample package	D
	Non-Working day	1 if weekend or vacation, 0 if it is a working day	D/S
	Time Slot	Slot of the day (morning, afternoon, evening, night)	S
	GPS latitude and longitude	Position of the device in GPS coordinates	D
	Accuracy	GPS Sensor's Accuracy from the mobile device	D
	Location Measure kind	Types of Location measure: GPS, Network, Mixed/Fused	D
	Speed	Speed as provided by the GPS driver of the mobile (as m/s)	D/S
	Average Speed	Average speed of the measures collected in the last two minutes	D/S
	Phone Year	Year/age of the terminal	D
	BDS	Availability of a BDS compliant GPS Sensor	D
	User Type	User Type: commuter, citizen, students, tourist, etc.	D/S
Accelerometer	Average linear magnitude of acceleration	Average of the acceleration magnitude calculate on five measurements	D
	Linear acceleration of X-axis	Acceleration of the device along the X-axis, purged by Earth gravity	D
	Linear acceleration of Y-axis	Acceleration of the terminal along the Y-axis, purged by Earth gravity	D
	Linear acceleration of Z-axis	Acceleration of the terminal along the Z-axis, purged by Earth gravity	D
Proximity	Rail Line	Bool indicating if the device is in proximity of a rail line	S
	Sport Facilities	Bool indicating if the device is in proximity of a sport facilities	S
	Tourist Trail	Bool indicating if the device is in proximity of a tourist trail	S
	Green Areas	Bool indicating if the device is in proximity of a green areas	S
	Bus/Light-rail Line	Bool indicating if the device is in proximity of a bus line or a light-trail line	S
	Cycle Paths	Bool indicating if the device is in proximity of a cycle path	S
Temporal window	Previous speed	Speed of the device of the previous 12 minutes slot	S
	Previous average speed	Average speed on the measures collected in a 12 minutes time slot	S
	Previous median speed	Median speed on the measures collected in a 12 minutes time slot	S
	Speed distance	Speed (m/s) calculated on the distance between two consecutive coordinates and the time passed between the observations	S

As described in Section II, the information about the user's movements is collected from the device sensors. If the user has the mobile application in foreground, the data are sent to the server every 1 minute and 30 seconds (sending interval). This interval can be reduced by the user (via the setting of the App) to an update up to 30 seconds, to have a more accurate assistance. If the App is not used, the data collection is performed in background modality, thus the measures and sending rates may become up to 3/5 minutes, forced by the operating system/device, which in some cases can hibernate the App. Therefore, in order to make the solution viable in real conditions (differently from the state of the art solutions), a set of strategies and robust classification algorithms have been put in place. Among them, techniques for filtering noise and GPS errors, and for smoothing the sequence of the user locations (user trajectory) have been used.

A Sensor Data Package l_i represents the user context at a specific time t_i and is composed by the GPS latitude and longitude (according to a Location Measure kind), speed, and accuracy of the measure plus a list of N additional features *(feat-1...feat-n)*:

$$l_i = \{latitude_i, longitude_i, speed_i, accuracy_i, feat - 1_i, ..., feat - n_i\}$$

user trajectory t_{ir} is a sequence of l_i that describes the movements of a user to move from l_i to l_r:

$$t_{ir} = \{l_i, ..., l_r\}$$

A segment s_{uv} is a trajectory t_{uv} in t_{ir} where a user keeps the same mobility mean:

$$l_i \to mobility - A \to l_u \to mobility - B \to l_v \to mobility - C \to l_r$$

where $l_u \to mobility - B \to l_v = t_{uv}$

The distance between l_u and l_v can be approximated by using flat-surface formulae between the two coordinates $< latitude_u, longitude_u >$ and $< latitude_v, longitude_v >$.

A measure of the terminal Speed can be directly retrieved from the GPS sensor (for example, every 30 seconds or at the rate imposed by the device). On the other hand, the above mentioned Average Speed of Table 4.1 is calculated over the sequence of l_i in the same sending slot from the mobile device, to cut out eventual errors coming from GPS sensor. If the mobile App is in foreground the Average Speed is computed every 2 minutes (4 measures of 30s, if any). If the mobile App service for collecting data is in background, and 2 minutes passed before a measure is available (probably the operating system put the application in hibernate mode). The service tries to wake up whenever it is possible (if the operating system on the device allows us to wake the service up), to retrieve a bounce of new l_i to calculate a more precise Average Speed. If at a given time, the measure of Speed is not available, a *valid* value may be obtained on the server-side by using the distance between the two last GPS coordinates and the current refresh time. Figure 4.3 overviews the scenario.

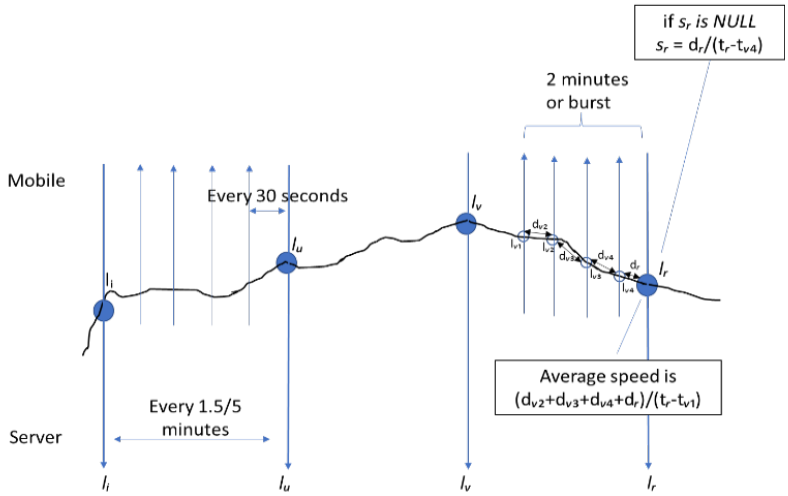


Figure 4.3: Speed and average speed.

The Location Measure kind is an important feature to understand the location measures reliability. Usually, the measures obtained and marked as “GPS” by the mobile device are quite accurate, even if they suffer time by time of well-known problem of shading (e.g., urban canyoning) or blocked (under the bridge) [128]. The location measures, labeled as “Network”, resume the position from the location of the available Wi-Fi hot spots or GSM/4G/5G in the mobile connection; while those marked as “Mixed” modality is obtained by the operating system by merging the previous strategies according to different algorithms that may depend on the operating system kind, sensor kind, etc. The Location Measure kind strongly depends on the factory settings of the device, that makes very difficult to force a pre-determinate modality from the App. On the other hand, the approach permits the users to optimize the battery use and to have more precise positioning in critical cases, thus the switch among modalities is not under control of the App. The Accuracy of the GPS measure is reported in meters from the device and can be used from the classificatory algorithm to eventually discharge entries.

Terminal model and its characteristics are also tracked and passed to the classification algorithm. Thus, the Phone Year of production of the device and the characteristics of the GPS sensors strongly influence the reliability of measure and thus have and have been considered as variable, differently from the state of the art solutions. Old terminals usually support just A-GPS modality, meanwhile new ones' support also GLONASS and BDS standards. We have been also capable to experiments on new GALILEO compatible sensors available on new Samsung models. As a result, there is evidence that the type of the sensor influences the accuracy of location retrieval [113].

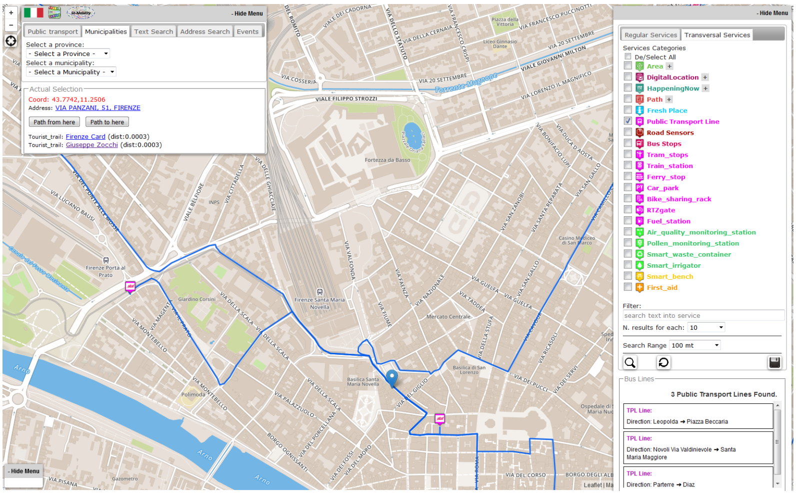
4.5.1 Accelerometer Features

Values from the **Accelerometers** of the terminal/device are always available and are sampled. Using the linear acceleration of the device avoids taking measures influenced by device orientation (horizontal or vertical, in the hand or in the pocket). Not all the mobile devices provide this information (some of them just return the non-linear values, influenced by the gravitational acceleration, and orientation, thus needing a de-rotation). On the other hand, almost all the relatively new devices already have this aggregated measurement available (for Android 8 [Android-cdd]). Phone Year variable allows us to take this into account. Thus, the three measures of linear acceleration on three axes have been considered aggregating five consecutive acceleration measures for computing an average magnitude as:

$$Average\ Linear\ Magnitude\ of\ Acc = \sum_{k=1}^{5} \frac{\sqrt{acc_{x_k}^2 + acc_{y_k}^2 + acc_{z_k}^2}}{5}$$

4.5.2 Distance Feature

On the server-side, the Sensor Data Package collected from the devices via the App are enriched by computing and, in most cases, exploiting the Km4City knowledge base of the City via Smart City API. This allows to retrieve contextual information about the closeness of the device/user with respect to: Railway Line, Sport Facilities, Tourist Trail, Green Areas, Bus/Light-rail Line, and Cycle Paths. The closeness features are binary values that specify if the location is closer to those structures, in the range of 30mt. This derived information is very valuable for understanding some transportation means. For example, to be close to a Rail and/or Bus/Light-rail line for a number of points of a trip permits to infer bus/train modality (train, bus and light-rail run just in their closeness) with a very high probability. On the other hand, the closeness to a cycle path cannot directly infer that a user is using a bike because the user can be in its proximity by a car and with similar speed or a bike can run also away for the cycling path (see Figure 4.4).



Select a province: - Select a Province -

Select a municipality: - Select a Municipality -

Actual Selection

Coord: 43.7742,11.2506

Address: VIA PANZANI, 51, FIRENZE

Tourist_trail: Firenze Card (dist:0.0003)

Tourist_trail: Giuseppe Zocchi (dist:0.0003)

Services Categories

- □ De/Select All
- □ Area +
- □ DigitalLocation +
- □ HappeningNow +
- □ Path +
- □ Fresh Place
- ☑ Public Transport Line
- □ Road Sensors
- □ Bus Stops
- □ Tram_stops
- □ Train_station
- □ Ferry_stop
- □ Car_park
- □ Bike_sharing_rack
- □ RTZgate
- □ Fuel_station
- □ Air_quality_monitoring_station
- □ Pollen_monitoring_station
- □ Smart_waste_container
- □ Smart_irrigator
- □ Smart_bench
- □ First_aid

Filter: search text into service

N. results for each: 10

Search Range 100 mt

Bus Lines

3 Public Transport Lines Found.

TPL Line:
Direction: Leopolda ➔ Piazza Beccaria

TPL Line:
Direction: Novoli Via Valdinievole ➔ Santa Maria Maggiore

TPL Line:
Direction: Parterre ➔ Diaz

Figure 4.4: Bus-line in proximity computation.

4.5.3 Temporal Window Feature

Besides having instantaneous measurements about device/user's mobility, the speed values in the last 12 minutes time-frame is also computer on server-side, as well as the average and mean value between these measurements. This allows to reduce the noise overcoming disruptive mobility conditions mainly related to traffic congestion or temporary signal absence. This is also due to the fact that the service for collecting data on the mobile device runs on a real application (foreground/background) conforming to the policy of "energy saving" of the user to have shortage of data for up to 3/5 minutes. So that, in real conditions, it is very important to avoid battery drainage warning, that may stimulate the user to un-install the App from the device. In order to perform an addition refinement on speed measures, mean and median speed and distance between GPS coordinates are also computed. The User Type specified by the user in the App during installation or setup permits contribute to the classifications and to the strategies. The **User Types** are: citizen, commuter, student, tourist, etc. We noticed that different profiles present a different approach in everyday mobility and, so on the transportation mode they normally use.

4.5.4 Results From Classification/Prediction Models

According to the above described data, the challenge was to predict the transportation mode, whether an individual is stationary, or is walking, or moving on a motorized private transport (car or motorbike) or using a public transport (tram, bus or train). The experiment has been conducted on about 30K observations, collected from April to August 2017 on **38 different users and 30 different kinds of devices**. Note that, each user can use the mean of transport they want. When the mode of transport is changed, the user was asked to notify the change to the App for creating the learning set and for validation. As mentioned above, no restriction was imposed on how the phone should be held during movement (foreground/background, on hand or bag, etc.). Unlike the experiments reported in the literature, most of the data were collected in the background because the phones were kept in pocket or bag, in fact there is a non-conformity in the frequency distribution of the collected data. In details, the frequency average is equal to 180 seconds and the variance is equal to 13240 seconds. The frequency distribution of the sampling period is reported in Figure 6.

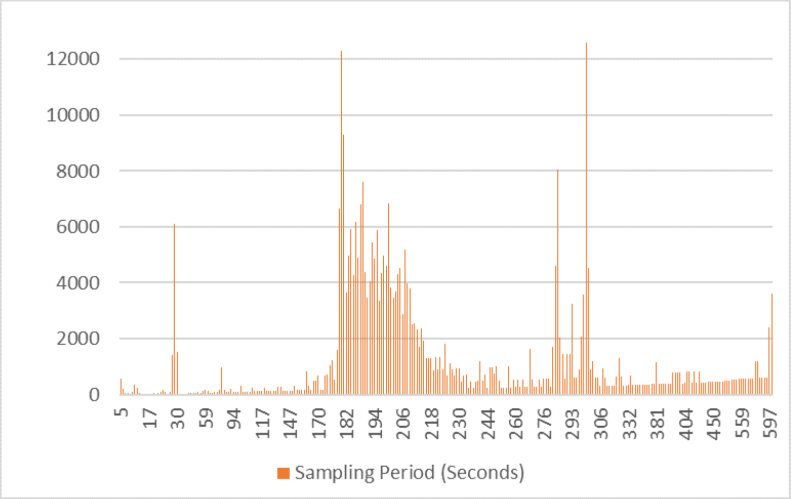


Figure 4.5: Frequency Distribution of Sampling Period.

The training set has been created by randomly selecting the 80% of the collected data, while the test set was the remaining 20%. In the general framework, three different approaches were more successfully considered - i.e., Random Forest (RF), Extremely Randomized Trees (Extra-Trees), and the Extreme Gradient Boosting procedure (XGBoost). Those approaches have been tested by using the above presented features/metrics (see Table 2), classified by categories as: baseline and GPS features, accelerometer features, distance features and temporal window features. The comparison among those models has been reported in Table 3, in terms of resulting data. From the comparison, it is evident that all the approaches are capable to produce satisfactory predictions (the accuracy for each model exceeds 90%) for the identification of the transportation means. According to the results reported in Table 3, the differences among the different approaches are not very relevant, while the results suggest that the **Extra-Trees** resulted to be the better-ranked approach in terms of accuracy and $F_1 score$. In Table 4.5.4, the $F_1 score$ is reported: $F_1 score$ has been used to measure the models' performances. This is a measure to evaluate the robustness of a model for making predictions, as a compromise between precision and recall:

$$F_1 score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$Precision = \frac{\#correctlyclassifiedistancesintoclassi}{\#istancesclassifiedasclassi}$$

$$Precision = \frac{\#correctlyclassifiedistancesintoclassi}{\#istancesbelongingtotheclassi}$$

Table 4.2: Classification Models Comparison on four classes of transport mode: stationary, non-motorized, private transport, public transport.

Classifier Models	**Accuracy**	**Precision**	**Recall**	F_1**score**
Gradient Boosting	0.947	0.773	0.828	0.800
Random Forest	0.942	0.774	0.869	0.819
Extra-Trees	**0.953**	**0.827**	**0.869**	**0.847**

According to this our first result, the Extra-Trees algorithm achieves an accuracy of 0.953, and a precision of 0.827. It should be remarked that, these results have been obtained and can be produced by observing data coming from a large range of devices and a variable sampling rate (up to 5 minutes). The model produce allows to understand if a user is moving with a public or private transport. On the contrary, in [159], a precision of 0.937 has been obtained by using a single device, Nokia n95, and a constant sampling rate of 60s, which is not realistic with present mobile operating systems. With the classification method presented in [159] was only possible to know if a user is moving with a motorized vehicle. The same considerations apply to: [173] where data come from three different devices and they are taken with a constant rate of 15s achieving a precision of 93.7%; and to [205] achieving a precision of 91% with accelerometer sensor data only, without distinguishing the type of motorized transport. Moreover, Table 4.5.4 reports the assessment of the results performed for each traveling mean classification for the Extra Tree procedure according to our first result. The traveling mean class with lower accuracy is Walk. This is probably due to the fact that, it is not easily to understand if a user is walking or not, since the GPS sensors accuracy is very noisy in indoor scenarios, with frequent jumps passing from the different modalities: wifi-mixed, etc.

Table 4.3: Extra-Trees Prediction Model: Statistic by class.

Extra Trees Model	**Stay**	**Walk**	**Private Transport**	**Public Transport**
Sensitivity	0.978	0.731	0.869	0.917
Specificity	0.901	0.988	0.987	0.996
Pos Pred Value	0.977	0.770	0.827	0.936
Neg Pred Value	0.904	0.985	0.990	0.994
Balanced Accuracy	0.940	0.859	0.928	0.956

4.5.5 Combining with Super Learner

Subsequently, with the intention of improving the precision, the Super Learner algorithm has been applied by dividing the multi-class problem into four binary classification problems, with 10-fold cross-validation, to estimate the risk on future data and select the optimal learner given the set of candidates above: Extra-Trees, RF, XGBoost. Then the results have been combined to make a classification on the highest probability estimate. Taking into account the classes of transport modality above (i.e., stationary, walking, private transport, public transport), **four different binary classification models** have been constructed:

(A) stationary vs walking, private transport, public transport;

(B) walking vs stationary, private transport, public transport;

(C) private transport vs stationary, walking, public transport;

(D) public transport vs stationary, walking, private transport.

The results for each binary classification model are reported in Tables 4.5.5. Where: RCV risk is a measure of model accuracy or performance (at lower value corresponds to a lower risk and higher accuracy).

The obtained results have been combined on the highest probability estimation. The complete statistic by class of Super Learner algorithm is reported in Table 4.5.5. Please note that, the coefficient columns indicate the weight of each individual learner in the overall ensemble, and the weight values are always greater than or equal to 0 and sum to 1.

Table 4.4: Super Learner results on each binary Classification Model: Couples A to D.

Method	**RCV risk**	**Coef**
Extra-Trees	0.0282	0.5391
RF	0.0287	0.0562
XGBoost	0.0300	0.4047

(A) Stationary vs Walking, Private Transport, Public Transport.

Method	**RCV risk**	**Coef**
Extra-Trees	0.0234	0.6277
RF	0.0259	0.0091
XGBoost	0.0252	0.3632

(B) Walking vs Stationary, Private Transport, Public Transport.

Method	**RCV risk**	**Coef**
Extra-Trees	0.0213	0.6857
RF	0.0235	0.0000
XGBoost	0.0239	0.3143

(C) Public Transport vs Stationary, Walking, Private Transport.

Method	**RCV risk**	**Coef**
Extra-Trees	0.0087	0.6296
RF	0.0108	0.0000
XGBoost	0.0096	0.3704

(D) Private Transport vs Stationary, Walking, Public Transport.

Table 4.5: Binary Classification Models combination based on the highest probability estimate: Statistic by class.

Super Learner Model	**Stay**	**Walk**	**Private Transport**	**Public Transport**
Sensitivity	0.990	0.662	0.857	0.927
Specificity	0.892	0.993	0.990	0.996
Pos Pred Value	0.975	0.831	0.865	0.953
Neg Pred Value	0.955	0.982	0.989	0.994
Balanced Accuracy	0.941	0.828	0.924	0.961

This is **our second result**. In this case, the average accuracy has been of **0.96**, precision of 0.865, and a recall of 0.857; with a $F_1 score$ equal to **0.861**.

Therefore, the results obtained by using the Super Learner overcome those of the Extra-Trees multi-class model (see Table 4.5.4), compared in terms of average accuracy and $F_1 score$, and those from the literature. For each class of transportation modes, three different algorithms have been compared in terms of RCV risk. Please note, that Extra Trees has obtained the lower risk in all the binary classifications. Therefore, according to Tables 4.5.5 and 4.5.4, there are no significant differences in terms of Balanced Accuracy between the Super Learner approach and Extra Trees multi-class model, except for class *Walk* in which Extra Trees model is better ranked in terms of accuracy and sensitivity. *For this reason, the Extra-Trees model could still be the best choice.*

4.5.6 Assessing the Influence of Features

A comparison in terms of accuracy, precision and recall of the Extra-Trees multi-class approach has been computed considering four combinations of the different categories of data (as reported in Table 2):

- baseline features and distance feature;
- baseline, distance feature and accelerometer features;
- baseline, distance feature and temporal window features;
- baseline, distance, accelerometer, temporal features together. (**Full Model**)

This set of combinations of feature categories permits to assess the flexibility of our approach in real operative conditions, where a variety of devices have to be supported, since not all devices support the full combination of categories. The results obtained by using different subsects of feature categories are reported in Table 4.5.6. Please note that the differences among the different cases for feature categories are substantial. The results suggest that the best choice in terms of precision is still the usage of model exploiting all the categories together, thus demonstrating that the model is flexible and resilient with respect to the device kind. Please note that the Boolean value detecting a close transportation line (i.e., proximity feature in the table) improves the classification effectiveness: the accuracy passed from 0.91 to 0.92 and higher.

Table 4.6: **Extra Tree Model results** on four classes of transport modality (stationary, non-motorized, private transport, public transport) considering four combinations of the different features.

Model features categories	**Accuracy**	**Precision**	**Recall**	F_1**score**
Baseline and GPS	0.910	0.682	0.751	0.714
Baseline and GPS + Proximity	0.924	0.739	0.691	0.715
Baseline and GPS + Proximity + Accelerometer	0.926	0.814	0.744	0.777
Baseline and GPS + Proximity + Temporal window	0.949	0.805	0.787	0.787
Baseline and GPS + Proximity + Accelerometer + Temporal window	**0.953**	**0.827**	**0.869**	**0.847**

In Figure 4.6, the features listed in Table 2 are reported in order of importance across the classes for the prediction of the Extra-Trees Full Model, (the model with all the categories of covariates). The distribution of relevance suggests that the variable Location Measure kind (i.e., GPS, Network or Fused/Mixed) is the most relevant for predicting the class of transportation mode, due to the fact that in Stay mode (during the night) usually the service is kept in background (thus the terminal operating system use a specific location provider to save battery usage). The relevance of each predictor has been evaluated using the ROC curve analysis [66]. For multi-class outcomes, the problem has been decomposed into all pair-wise problems. The area under the curve has been calculated for each class pair (i.e., Stay vs Walk, Walk vs Private Transport etc.). The maximum area under the curve across the relevant pair-wise AUC's is used as the variable importance measure of a specific class.

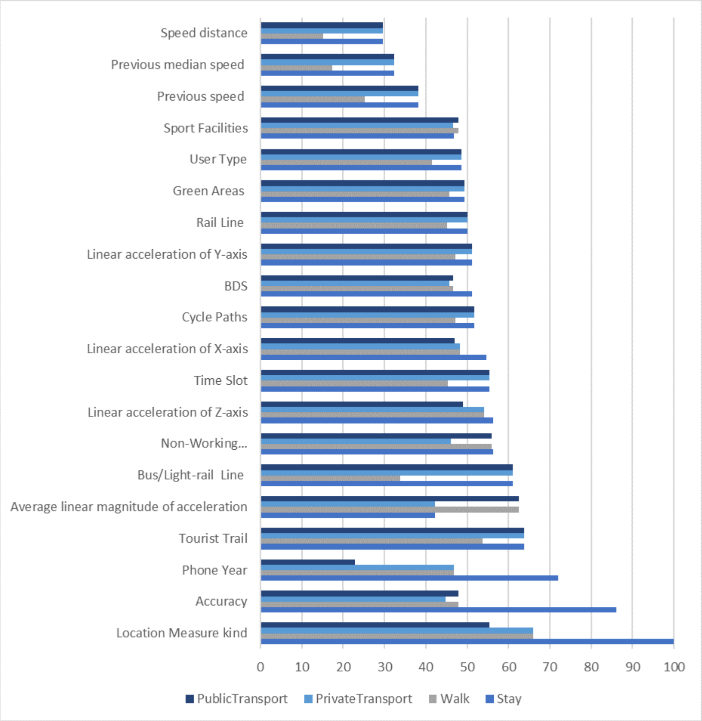


Figure 4.6: Variables Importance across the classes of the Extra-Trees full model.

4.5.7 Hierarchical Approach

The above presented and adopted learning models reflect the one-step algorithm methodology. In this section, those models with a two-step hierarchical approach in terms of final accuracy and testing execution time are compared. The hierarchical approach can be considered a combination of the Extra-Tree multi-class classification and the Super learner algorithm. As first step, the Extra-Tree multi-class classifier (as reported in Table 3), from which the two transportation means with higher probability is considered. Subsequently, as second step, the Super learner approach has been used to discriminate between these transportation means. A threshold has been used to decide which class can be considered directly correct at the first step: if the probability of the class is higher respect the considered threshold (0.90), the transportation modality is regarded correct without proceeding to the second step. In this case, it can be difficult to know a priori which machine learning method will work best [152], especially if the two transportation modes that have to be discriminated can variate among the different combination of transportation mode pairs: Super learner can be a solution to compare different approaches and find the best one or the best combination.

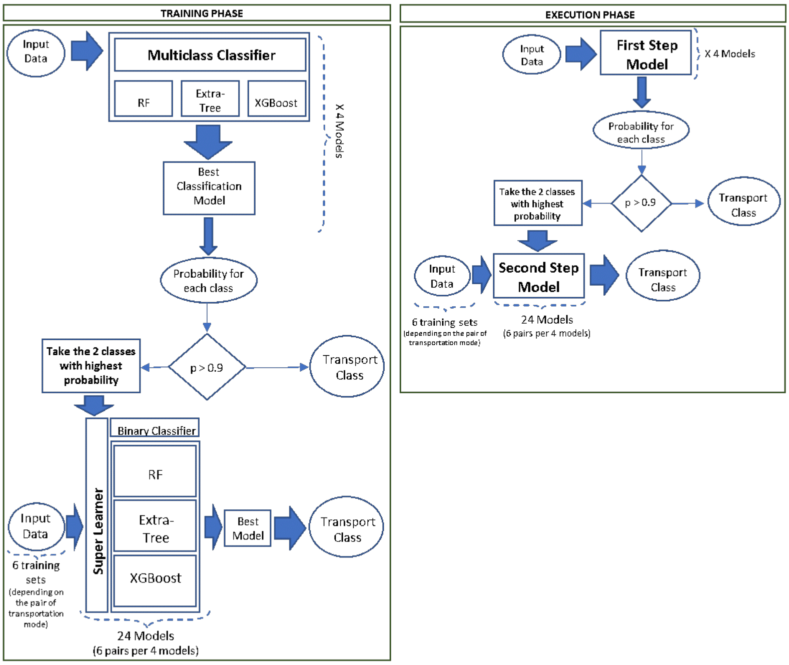

TRAINING PHASE
Input Data
Multiclass Classifier
RF
Extra-Tree
XGBoost
X 4 Models
Best Classification Model
Probability for each class
p > 0.9
Transport Class
Take the 2 classes with highest probability
Binary Classifier
Super Learner
RF
Extra-Tree
XGBoost
Input Data
6 training sets (depending on the pair of transportation mode)
Best Model
Transport Class
24 Models (6 pairs per 4 models)
EXECUTION PHASE
Input Data
First Step Model
X 4 Models
Probability for each class
p > 0.9
Transport Class
Take the 2 classes with highest probability
Input Data
Second Step Model
Transport Class
6 training sets (depending on the pair of transportation mode)
24 Models (6 pairs per 4 models)

Figure 4.7: Scheme of the hierarchical approach in training and execution.

The considered machine learning algorithms are the Extra-Tree, the Random Forest and the XGBoost. As reported in Figure 8, the classes with higher probability respect to the threshold are not considered in the step two of the hierarchical model: the classification has been considered correct and no corrections have been made in the second step using the second learning approach (Super Learner). In the first step, the Extra-Tree algorithm have achieved a 97.6% precision for the classes with probability higher than the threshold. In the second step, six new binary classification models have been created, one for each combination between pairs of the transportation modes selected during the step-one (the classes with a probability lower than the threshold, i.e., Stay-Walk, Stay-Private Transport, Stay-Public Transport, Walk-Private Transport, Walk-Public Transport, Private Transport-Public Transport).

Table 4.7: Two-steps hierarchical approach confusion matrix (Extra-Tree and Super Learner considering Baseline, GPS, proximity, Accelerometer and Temporal window features).

Two-Steps Hierarchical Approach		**Predicted**			
		Stay	**Walk**	**Private Transport**	**Public Transport**
Actual	**Stay**	**0.98**	0.30	0.09	0.03
	Walk	0.01	**0.60**	0.02	0.01
	Private Transport	0.01	0.07	**0.87**	0.07
	Public Transport	0.00	0.03	0.01	**0.89**

Table 4.5.7 reports the confusion matrix related to the hierarchical approach (Extra Tree & Super Learner). Note that, Super Learner algorithms takes a weighted average of the learners using the coefficients/weights, with 10-fold cross-validation. For the two steps hierarchical approach, average accuracy is 0.94, while precision and recall are 0.786 and 0.869 respectively. It is also interesting to note that the average accuracy of each combination of transportation modality significantly decreases respect to the first step. This can be due to a loss of information from the first to the second step. In fact, after the application of step-one (Extra-Tree algorithm), for the classes with a probability below the threshold, the achieved accuracy is higher respect to the average accuracy calculated during the step-two (80.8% and 77% respectively). Moreover, the percentage of transportation modalities correctly classified during the step-one is of the 75%, while the remaining 25% of the test sample is further classified during the step-two.

4.5.8 Real Condition Scenario VS Hierarchical Approach Limitations

Several considerations have been already presented about the critical aspect of working on real operating conditions. Battery drainage and the opportunity to support a contextual service for the users, even with the application in background mode, drove our research, despite little decrease of accuracy and precision. A client-server architecture has been designed to support a finer classification, using GIS data easier available on the server side (avoiding user terminal network bandwidth usage to eventually download from remote) and to support technologies to aggregate information cross-terminal and user agnostic. Implementing a central server-side classification algorithm leaves open also the chance to auto-update scenario with feedback provided directly by the user. However, a real condition scenario can be affected by some limitations that cannot be solved either if a hierarchical approach is applied. This is due to the fact that the phone/user characteristics can be manifold, e.g., the presence of accelerometer information, the different type/-generation of gps sensor, the presence of information related to the temporal window, etc. For this reason, the classification model has to be flexible and the training data set has to be as much as possible various (e.g., any kind of generations, manufactures, years, characteristics, etc.) without any restriction. The application of a two-steps approach, as demonstrated in the previews subsection, may lead to a loss of accuracy due to a loss of information and can be more time consuming in terms of execution time and number of different training models. In detail, during the second step, six different training models have to be executed, one for each combination between pairs of the transportation modes (selected during the step-one), considering that the classes of transportation means are four. In addition, a specific model has to be created depending on the characteristics of the device and of the users, considering four combinations of the different categories of data (reported in Table 2). Therefore, 4 different training models during the first step, and 24 different training models during the second step (6 transportation modality pairs combinations per 4 categories combinations) have to be computed.

4.5.9 Final Solution

In the previous subsections a comparison between different solutions has been presented and discussed. On one hand, a two-steps hierarchical approach has been proposed. In the first step a multi-class classifier algorithm has been adopted to classify the transportation modalities. After the first classification, the classes with a probability lower than a threshold of 0.90 (prob ¡ 0.90) have been re-classified in the second step, while the classes that have a probability higher than 0.90 are considered as correct and excluded from the re-classification test set. During the second step six different binary classification models have been trained, one for each pair of transportation modality. On the other hand, a single step classification model has been presented and different models have been compared. The Extra-Tree algorithm can be considered as the best and final solution: it was found to produce the best performance in terms of average accuracy (0.953) and time consuming. In detail, four different models have been trained to make the approach as flexible as possible. The necessity of this flexibility is because the solution has to be applied in a real condition scenario, for different phone/user characteristics, in any pseudo real-time context. The advantage of this solution is not only in terms of accuracy but also in terms of number of training models (4 different models vs 4+24 different models in the hierarchical solution).

4.6 Considerations

This research has been focused on presenting a solution to create a classification system that uses mobile devices' sensor values and GIS data (user contextual information) to identify the transportation mean of users: stationary, walking, on a motorized private transport (car or motorbike) or in a public transport (tram, bus or train). The goal has been to define a solution for sustainable mobility, delivering to the user useful personalized assistance messages. A number of metrics and features have been chosen as the baseline and GPS, the distance, the accelerometer data and the temporal windows data. The research documented in this Chapter demonstrated that a one-step multi-class classifier solution was found to produce the best performance in terms of average accuracy and time consuming if compared to a hierarchical approach. In detail, the Extremely Randomized Trees exploiting all the discussed above data can be a robust approach for reliable, precise and fast estimation of transportation means. The proposed solution overcome those of the literature since it presents a solution that is capable to produce reliable results in real conditions (i.e., real-time applications and background modality of operations) with a real set of devices and in particular: (i) addressing a large number of devices providing different features, different GPS sensors, different accelerometer sensors, etc., (ii) working with time variable samples of the data that may be due to the different operating systems, energy saving setting, etc., which are not under control of the App and thus are a strong constraint to realize real applications, background/foreground modality of operation; (iii) exploiting a number of different features and obtaining results with higher precision and accuracy. For these reasons, features related to the type of phone, e.g., the presence of accelerometer, phone year, location provider etc., have been considered in the prediction model, contributing to perform corrections in the model. The prediction model proposed has been created by exploiting open and real-time data using the Sii-Mobility (national smart city project of Italian Ministry of Research for terrestrial mobility and transport, `http://www.sii-mobility.org`) solution based on Km4City infrastructure http://www.km4city.org in the Florence area, Italy. The solution is deployed as an additional feature on Smart City Apps in the Tuscany and Florence areas for sustainable mobility, which is now in place for stimulating the private mover toward a more sustainable mobility with the collaboration of three major public transportation operators: ATAF, BUSITALIA and CTTNORD. Most of the computations were conducted in R Statistical Environment (`https://www.R-project.org/`), and then implemented in real time.

Chapter 5

Enviromental Data Network and Automated Analysis and Representation

This chapter is focused on presenting a system to carry out automatic real-time statistical data analysis from environmental sensors positioned in Smart Cities. The main objectives are to provide services independently on the number of sensors, on their position, and to provide services on a large number of devices in a modality that can be understood by everybody. The sensors network can also be enriched by devices hosted by city users. The environmental data collected from personal devices, have been used to provide informative view regarding environmental data. In particular, the data results are computed through a specific IOT Applications exploiting data analytics. The IOT Application can manage the data analytic process by choosing input data, the time interval to be analyzed to produce the resulting heatmap which are saved into a GeoServer for further visualization via Internet on mobile App and Dashboard. Furthermore, heat-maps interpolation errors trends have been used to detected devices dysfunction related to a bad trend over time. Such anomalies may often be useful to alert the user about a problem on the device by sending *them warning messages.*[1]

5.1 Introduction

The public is increasingly aware of the health and economic costs of air pollution. Poor air quality is linked to over three million deaths each year, and 96% of people in large cities are exposed to pollutant levels that are above recommended limits. The costs of urban air pollution amount to 2% of gross domestic product in developed countries and 5% in developing countries [112]. For these reasons, most of the cities and regions are increasing their attention to the real-time monitoring of environmental and weather parameters [153], [104]. Cities have an interest in understanding how much pollution affects the quality of the air that citizens breath in order to properly regulate urban mobility and give to all the awareness that they are living in a city that is increasingly technological and oriented towards focusing on citizens' health and thus quality of life. In the past, the usage of environmental data sensors were in most cases limited to the public administrations, to main devoted institutions. They had, in most cases, also difficulties in make public the information regarding air quality, for the lack of capabilities to provide answers at the city users eventual questions. The main difficulties have been due to the objective difficulties in understanding the meaning of the measured parameters, which may strongly depend on the specific position in which the sensors are located. For example, providing the values of two sensors for a whole city it does not mean that the measured values are valid for the whole city as well.

On the other hand, in the past, it was too complex and expensive to have a large network of sensors measuring the air quality in each point of the city. In reality, it is reasonable to have large differences if the air quality parameters are measured on a main street with high traffic rather than in a garden just on the back of the house located on the same main street. Recently, there is a wider understanding of the meaning of the environmental parameters (for example, PM_{10}, $PM_{2.5}$, CO, CO_2, SO_2, O_3, H_2S, NO, NO_2, NO_x, etc.), and how much they are influenced by the city structures, how the high values are provoked, which is the dynamic of their diffusion/propagation,

[1] This chapter has been presented as "Environmental Data Network and Automated Analysis and Representation" at the *5th Italian Conference on ICT for Smart Cities and Communities, i-Cities, Pisa, 2019.*

etc. Today, there is also the direct interest and the economic possibility of a number of city users to buy and install their own sensors.

The intention of putting the resulting data at disposal of the community can lead to a double benefit: from one side, the advantage of knowing the values of the measured parameters on their premise; on the other side, to have a global view of the data of the city with a denser sensor network. Even if some of low cost sensors are of low quality, the increased number of them, and the procedures for their calibration can compensate. The city users are also interested in hosting sensors to take decisions on their activity in the city on the basis of the measured data. For example, opening the back windows in certain cases, rather than the front ones, or getting out in the garden with the baby or dog, or to choose the path for cardio running.

5.2 Real Time Data Analytics architecture

The Snap4City solution presented in Section 1.1.3, allow to ingest and manage Big Data coming from IoT devices, providing a set of visual tools enabling the production of interactive dashboards for data analytics and supporting decision-making processes. In terms of Data Analytics, in Snap4City platform has been developed a specific architecture able to manage and analyze data automatically and to create data visualization tools (e.g., monitoring dashboards or heat-maps) in real-time 5.1. In the platform context, Data Analytics would mean to provide Micro-Services/nodes exposing a number of services to be exploited into the IOT Applications for Smart Cities that are obtained as: *IOT App = Node-RED + Snap4City Micro-services.* In fact, Snap4City suite provides generic Data Analytics Micro-Service where Data Analytics algorithms developed in R-Studio can be put in execution, according to specific guidelines. A collection of more than 150 Smart City Micro-Services has been also developed as Nodes for Node-RED programming environment. Node-RED visual programming philosophy allows the creation of event driven data flow applications where the exchanged messages are in JSON format. On the other hand, also periodic processes can be developed by scheduling one or more internal timers. This means that users can develop IOT Applications as Node-RED flows, exploiting both Push and Pull data protocols, with the same visual programming environment. In detail, the data analytic part is developed using the statistical IDE R Studio. The integration of R is performed using the package *Plumber* that converts the existing R code to a web API. The run time engine of Node-RED is executed on Dockers, so as the Plumber package. Each *plumber* node creates a single thread docker that contains an R script.

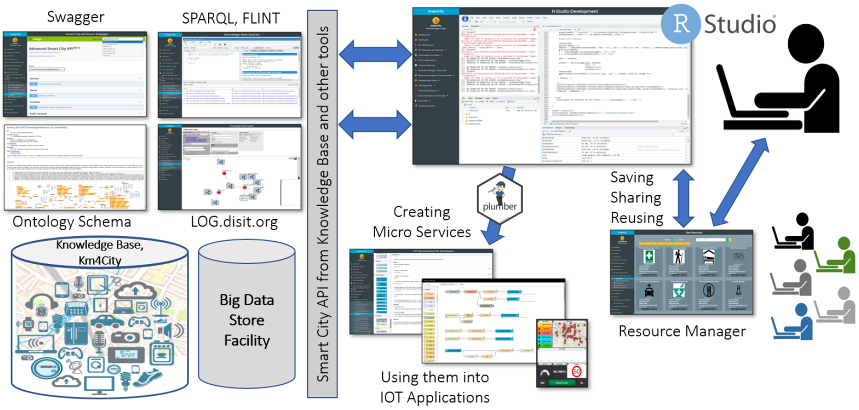


Figure 5.1: Snap4City Real Time Data Analytics architecture

5.3 City of Helsinki's Use Case Scenario

City of Helsinki's use case scenario is aimed at environmental monitoring. The reason for air pollution monitoring is to increase the degree of innovation, enabling stakeholders through data collection to develop a variety of completely new data-driven services for citizens. The environmental monitoring use case has been primarily performed in a new smart district called Jätkäsaari, a small connected island to the South from the City center. In addition to 20.000+ future inhabitants and workplaces for 6000 people, including various hotels and office facilities, Jätkäsaari also encompasses the main part of Helsinki's passenger harbor. However, geographical features and historical development of infrastructure significantly limit connections to the mainland and obstruct traffic in the area. The city of Helsinki aims developing new smart mobility solutions focused on this hot-spot. The large construction sites, the intensive and obstructed traffic, and the growing population create environmental challenges in Jätkäsaari. Thus, there is a need for a platform that can integrate data from different sources and services, provide tools for data analytic, and enable development of new innovative services to easily and quickly get reliable information about the current state of the air quality and other environmental indicators in different parts of Jätkäsaari.

There are ongoing activities to measure air quality near construction sites performed by Helsinki Region Environmental Services (HSY). During March-October 2018, the polluting effects of construction sites are being measured at another Helsinki Smart District Kalasatama. The concentration of inhalable particles (PM_{10}) is being measured near Kalasatama School and other sensitive targets, such as day care centers, playgrounds, primary schools, senior citizens' housing and services, and hospitals. Smaller inhalable particles (less than 10 μm in diameter, PM10) are not noticeably visible but they can cause problems with health. The measurement results describe the air quality in a residential area with several major construction sites in the immediate vicinity. Respiratory particulate concentrations are high when the daily average exceeds the limit value of 50 $\mu g/m^3$. Air quality is poor when the hourly rate is above 100 $\mu g/m^3$.

For better monitoring the area under development and innovation, citizens and business owners have installed their own sensors in order to create a denser sensor network. Note that, although personal sensors have not yet achieved their market potential, applications are becoming a mainstay of research by showing people's exposure to environmental factors [150].

The environmental data collected from IOT Brokers, included IOT Devices hosted by city users, and from data providers have been used to (i) provide informative view to city users regarding environmental data via some mobile App, and to (ii) provide detailed information about the Environmental data to city officials for decision making. To this end, a number of heat-maps and dashboards have been created to provide data results in real time in the hands of city officials.

5.4 Interpolation Technique

The creation of heat-maps for particulate matters (see Fig.5.2) is based on a gridded bivariate interpolation for irregular data [3]. The bivariate interpolation method consists of five procedures: (1) triangulation (i.e., partitioning into a number of triangles) of the $x - y$ plane; (2) selection of several data points that are closest to each data point (sensor) and are used for estimat-

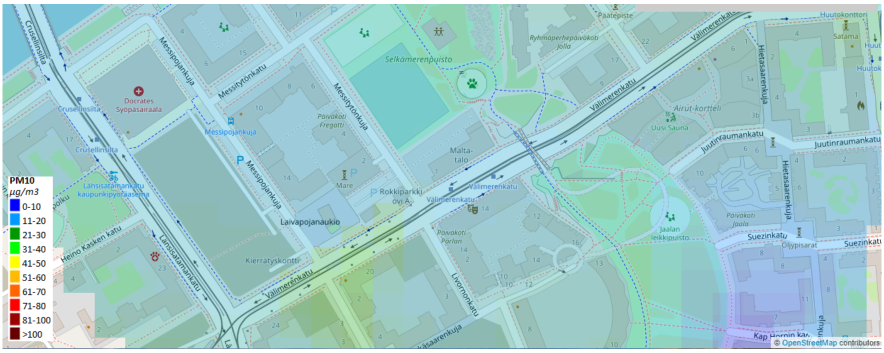


Figure 5.2: Air Quality PM10 interpolation heat-map for a small area of Jätkäsaari Island

ing the partial derivatives; (3) organization of the output with respect to triangle numbers; (4) estimation of partial derivatives at each data point; and (5) punctual interpolation at each output point [3]. More precisely, for a unique partitioning of the plane, the $x - y$ plane is divided into triangles by the following steps. First, determine the nearest pair of data points and draw a line segment between the points. Next, find the nearest pair of data points among the remaining pairs and draw a line segment between these points if the line segment to be drawn does not cross any other line segment already drawn. Repeat the second step until all possible pairs are exhausted. The z value of the function at point of coordinates (x, y) in a triangle is interpolated by a bivariate fifth-degree polynomial in x and y:

$$z(x,y) = \sum_{j=0}^{5} \sum_{k=0}^{5-j} q_{jk} x^j y^k.$$

The coefficients of the polynomial are determined by the given z values at the three vertexes of the triangle and the estimated values of partial derivatives (i.e. z, z_x, z_y, z_{xx}, z_{xy}, and z_{yy}) at the vertexes, together with the imposed condition that the partial derivative of z by the variable measured in the direction perpendicular to each side of the triangle be a polynomial of degree three; at most, in the variable measured along the side. [3].

5.4.1 Validation

In Jätkäsaari Island about 25 devices have been taken into account for the interpolation. Each device performs two different measures of particulate matters (PM_{10}, $PM_{2.5}$). For each particulate matter sensor has been computed the interpolation for heat-map creation, with a resolution of 2×2meters. In this way, final users have a real-time overview of the area quality situation within the smart zone and close to their homes.

For each sensor measure, the interpolation accuracy has been evaluated in terms of percentage error. The error evaluation of the interpolation approach is based on the exclusion of data from a different selected air quality sensor. More precisely, for each time t we are going to estimate the error between the calculated interpolated value $\hat{z}_i(t)$ in the position which locates the selected i-th sensor respect to the measured/real-time air quality value $z_i(t)$ from the i-th sensor. The percentage error of the i-th sensor e_r^i at time t is calculated as

$$e_r^i(t) = \frac{|z_i(t) - \hat{z}_i(t)|}{z_i(t)} \times 100.$$

The accuracy of the whole approach is estimated by considering the same procedure for each data sensor at time t. Then, the percentage system error per time slot is evaluated as $\frac{1}{S}\sum_{i=1}^{S} e_r^i(t)$ where S is the data sensors number. About 3 weeks data has been used for the interpolation errors evaluation. The error measures have been computed for: (1) week-ends and working-days, without considering different devices and time slots; (2) week-end and working days on each device; (3) week ends and working days per time slots; (4) week-end and working days on each device per time slots. In case (1) the PM_{10} absolute percentage errors are 0.78 and 0.73 for week-end and working-days respectively, while the $PM_{2.5}$ absolute percentage error is 0.99 for the working-days and 0.87 for the week-end.

5.4.2 Anomaly Detection of Sensor Dysfunctions

Measurement errors can be caused by a variety of factors and some countermeasures are needed to be taken accordingly. When a sensor error occurs, it is important to examine the cause of measurement errors thoroughly in order to implement anomaly detection systems. This is important for ensuring stable quality in measurement and in the creation of interpolation map.
Errors in measures can be manifold.

- Errors caused by the measurement system: calibration error; measurement errors originating in the measurement system; deterioration of measurement accuracy over time (deterioration caused by wear in consumable components).
- Errors caused by the user: bad positioning of the devices, mishandling of the measurement system, different degrees of skill of the users; user-specific methods of reading the scale.
- Errors caused by environmental conditions: deformation of the measurement target caused by rapid changes in air quality measure; measuring in locations with varying air quality measure levels.

Errors caused by the measurement system or environmental conditions can be easily identified as peak respect to the average trend of the measure. When a device is left in the user's hands, one of the most likely errors that may occur can be a dysfunction due to a bad positioning of the device. Identifying this type of error is important because an alert message can be sent to the user once the dysfunction has been detected. An idea of countermeasure is to use the validation error as detectors of device dysfunctions: it's possible to understand anomalies on devices just comparing error trends on each devices. If the error trend is higher than the error confidence interval, it's likely to be a problem on the device. Once checked the error trend, the second step is to monitor the error on the other sensor (pollutant measure) installed on the same device. If the second measure trend error is similar to the first one, the presence of a dysfunction on the device is highly probable. This error control is quite different from a simple real-time measure trend control. A positive/negative peak on trend can be due to multiple factors and is possible to detect it just comparing the device with a nearest one while, in this case, the detected dysfunction is related to a bad trend over time. Such anomalies may often be useful to alert the user about a problem on the device by sending them warning messages. To check possible dysfunctions, for each time slot t, all the point in the area of interest have an estimated/interpolated air quality value. The interpolation error and the confidence interval are computed every 24 hours on the basis of the validation method presented in Subsection 5.4.1. The confidence interval for the average error has been computed considering a period of 3 weeks (working days and week ends distinctly).

Basic Computational Approach

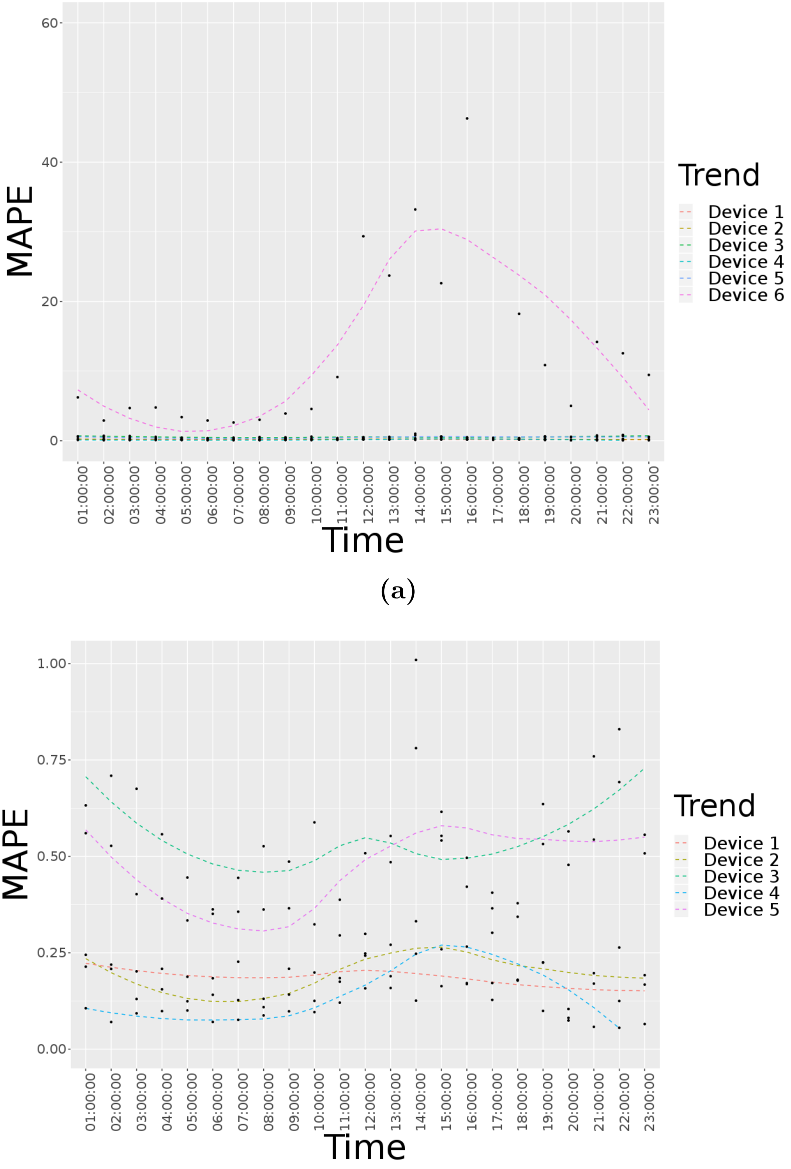
A computational approach for detecting dysfunction in real-time observations can be executed according to the following steps:

```
Input: $S$ = Sensor Number
Input: $z_i(t)$ real-time value of the i-th sensor at time $t$
for each time $t$ do
  for $i = 1$ to $S$ do
    compute $\bar{z}_i(t)$ average value of the i-th sensor at time
    compute $CI_{\bar{z}_i(t)}$ 95% confidence interval for the average value
    compute $\hat{z}_i(t)$ interpolated value in the i-th sensor location
    compute $e_r^i(t)$ error interpolation measure in the i-th sensor
    compute $\bar{e}_r^i$ average error interpolation measure
    compute $CI_{\bar{e}_r^i}$ 95% confidence interval for the average error
    if $|z_i(t) - \bar{z}_i(t)| > CI_{\bar{z}_i(t)}$ then
      print High probability of Error in Measure in the i-th sensor
      mark i-th sensor on the map
    end if
    if $|z_i(t) - \bar{z}_i(t)| < CI_{z_i(t)}$ and $|\bar{e}_r^i| > CI_{\bar{e}_r^i}$ then
      print High probability of device dysfunction
      save The i-th sensor coordinates
      send Alert message to the user
    end if
  end for
end for
```

Figure 5.3 shows PM_{10} interpolation error trends in terms of absolute percentage error. In Fig. 5.3(a) the trend of the device with dysfunction (Device 6) and other five devices' error trends are compared. In 5.3(c) trends for the five devices without any dysfunctions are compared.

5.5 Considerations

The environmental data collected from devices hosted by city users and from data providers, have been used to provide informative view to city users regarding environmental data via some mobile App, and to provide detailed information about the Environmental data to city officials for decision making. In particular, the use of a personal device gives the possibility of city users to better monitoring a specific area. Further, the intention of putting the resulting data at disposal of the community is a double benefit: on one hand, is possible knowing the values of measured parameters on their premise, for the other hand is possible to have a global view of data from the city with a denser sensor network, also thanks to the creation of interpolation heatmaps. However, the use of personal sensors has some disadvantages. For example, it is not possible to check if the device is correctly positioned (e.g., inside the house instead of outside). To solve this problem, a solution can be monitoring the trend of interpolation mean absolute percentage errors. To check possible dysfunctions, one week interpolation data has been used for the errors evaluation, and the detected dysfunction was related to a bad trend over time. Such anomalies may often be useful to alert the user about a problem on the device by sending them warning messages.



(a)

(b)

Figure 5.3: Air Quality PM_{10} working days interpolation error trends per hour in terms of mean absolute percentage error for (a) six personal devices including the device with a dysfunction; (b) five personal devices

Part II

Social Media Data Analysis for Citizen

Chapter 6

Predictive models for retweeting

This chapter is focused on presenting the research results regarding a solution to predict and understand retweet proneness of a post on Twitter (tendency or inclination of a tweet to be retweeted). Several features extracted from Twitter data have been analyzed to create predictive models, with the aim of predicting the degree of retweeting of tweets (i.e., the number of retweets a given tweet may get). The main goal is to obtain indications about the probable number of retweets a tweet may obtain from the social network. The usage of the classification trees with recursive partitioning procedure for prediction has been proposed and the obtained results have been compared, in terms of accuracy and processing time, with respect to other methods.[1][2]

6.1 Introduction

In recent years, social media have become an important communication tool and instrument for monitoring preferences of users, as well as making predictions in a number of contexts. Many social media platforms allow rapid

[1] This chapter has been published as "Assessing the reTweet proneness of tweets: predictive models for retweeting" in *Journal of Multimedia Tools and Applications, 2018* [135].

[2] *Acknowledgments:* The work has been developed in the context of smart city projects of the European Commission RESOLUTE H2020 (`www.RESOLUTE-eu.org`), in which the capacity of communicating information is fundamental for advertising, promoting alerts of civil protection, etc.

multimedia information diffusion, and thus they may be used as a source of information for viral advertising and marketing, early warning, emergency response and, more generally, for promoting and/or informing many users. Among the various platforms, Twitter.com has a very large user base, consisting of 1.3 billion of accounts and hundreds of millions of users per month. Twitter users can produce a post (i.e., a 'tweet'), about any topic within the 140-characters limit and can follow other users, in order to receive their tweets/posts on their own twitter web page, as well as on the mobile App. Twitter plays an important role in spreading information, allowing people to communicate and share contents in a fast manner. The posts made by a user are displayed on his/her profile page, and they are also brought to the attention of all his/her followers. It is also possible to send some direct private messages to other users without provoking diffusion. Another solution to enhance the diffusion and the echo of tweets is to include in a tweet including a direct mention of a user; this can be done by using the *'@'* prefix such as *'@ usernickname'*. In this case, the @ *usernickname* user is stimulated by receiving a notification. Therefore, the information conveyed in a tweet is diffused among the social network users through retweets of the former tweet, thus echoing the original message to the followers, hence producing a chain of messages since the retweets are also echoed. A retweet represents the echo of an original tweet made by one user that has been automatically forwarded by Twitter.com to the followers of the retweeting users (a part for eventual promotions performed by *Twitter.com* for featuring the most important tweets when they are getting on the list of the most appreciated). In the world of Twitter, the effectiveness of a tweet is frequently measured in terms of retweet count, which is the number of times the tweet has been retweeted [147]. It gives a measure of the number of reached audience and/or appreciation.

There is a growing interest, both in research and commercial fields, for influential strategies and solutions for seeding and diffusing information. Twitter offers to business users the possibility to integrate its analytics with audience measurement tools and services, such as Nielsen Digital Ad Ratings (DAR) and ComScore validated Campaign Essentials (vCE). Overviews of predictive methods exploiting tweets have been proposed in the works of [169], [122], [206]. In most cases, the predictive capabilities of Twitter data have been identified by using volume metrics on tweets (i.e., the total number of tweets and/or retweets associated with a Twitter user or presenting a certain hashtag). However, in specific cases, a deeper semantic understanding of tweets has been required to create useful predictive capabilities. Thus, algorithms for sentiment analysis computation have been proposed to consider the meaning of tweets by means of natural language processing algorithms. Moreover, the adoption of techniques for segmenting, filtering or clustering by context (e.g., using natural language processing for avoiding the misclassification of tweets talking about flu), or by users' profiles (e.g., age, location, language, and genre) may help to obtain more precise results in terms of predictability. On the other hand, the aim of this Chapter is to study the retweet proneness of a tweet, which we define and refer in the following of the Chapter as the capability to be retweeted, including a quantitative measure of the number of retweets a given tweet may get (which can be considered as the potential degree of being retweeted). This Chapter is focused on presenting a study on identifying and assessing the most representative metrics which can be used to predict the degree of retweeting of a tweet (i.e., the number of retweets a given tweet may get). According to the literature, the tweet features can be related to the tweet, to the author of the tweet and thus to the network of relationships of the tweets' author. The study is grounded on the analysis of tweets datasets collected in different areas in the last 18 months, for a total amount of about 100 million posts. By analyzing the datasets with the aim of identifying the best predicting model allowed us to identify also the main characteristics of tweets to predict the degree of retweeting. Please note that, according to the state of the art reviewed and presented in the following section, the identification of models for estimating of the degree of retweeting of a tweet has been only partially addressed in the literature; a few efforts are mainly focused on identifying parameters to guess the probability of retweeting, and/or to study the cascading effected through the network.

To our knowledge, the aims not simply predicting the probability for a tweet to be retweeted, rather to go a step further, which is predicting and estimating the degree of retweeting. Moreover, the proposed analysis identified additional relevant metrics/features, with respect to those proposed in the reviewed literature, such as the publication time of tweets and the number of users who added a given tweet's author to a list, as discussed later in more detail. The motivation for establishing the probability of prediction of a tweet is related with the value of the tweet itself and the value of the advertising service that may have produced it. The estimation of the probability to be retweeted is a measure of the effectiveness of a tweet and it is somehow a more precise measure of the concept of tweet virality, that tends to assess only tweets and their context to create huge volumes of retweets.

6.2 State of the art

In this section, the predictive capability of Twitter data has been reviewed with the aim of providing a better view of the context in which the research has been developed, and the impact of the obtained results. In the work of [170], a solution for predicting results of football games has been proposed, taking into account the volume of tweets. Opinions pools and politic elections predictions have been proposed to be correlated with the volume of tweets by using Sentiment Analysis techniques in [139]. Different models based on volume of tweets and other means have been also used for predicting purposes: voting results in [25] and in [183], economics [28], [50], marketability of consumer goods [166], public health seasonal flu [1], [110], [168], box-office revenues for movies [10], [117], [127], crimes [193], book sales [86], recommendations on places to be visited [47] and weather forecast information [85], [84]. Moreover, Twitter-based metrics have been used to predict and estimate the number of people in some location, such as airports, the so-called crowd size estimation by the work of [29], as well as to predict the audience of scheduled television programmes, where the audience is highly involved, such as it occurs with reality shows (i.e., X Factor and Pechino Express, in Italy) [55] as discussed in Chapter 7. Other adoptions of Twitter have been used to perform risk analysis [98].

In general, a Twitter user could find a tweet worth sharing, and therefore he/she may retweet it to followers. There is no upper limit to the number of times a retweet (re-post) operation can be performed. Hence, multiple levels of retweeting can be identified (considering the retweet of an original tweet as the first-level). A user could actually retweet a formerly retweeted post to his/her followers, and his/her followers can do the same again and again. In this way, retweets became a popular mean of propagating information through the Twitter community, as they may get viral propagation when volumes of retweets become high. Most studies about the assessment of the retweeting capability of tweets (proneness of a given tweet to be retweeted) try to analyze retweeting behaviors and, thus, to discover the features that may help Twitter users (i.e., the tweets' authors) in creating tweets which are more effective in collecting retweets. In the literature, different models have been proposed to shed some light on what kind of factors are likely to influence information propagation in Twitter.

Various motivations for retweeting behaviors have been explored in [31]. They found that the most influential users can retain significant influence over several different topics. In the works of [108] and [46], the relationships between the number of followers of Twitter users and their influence and lists of the most influential Twitter users, compiled according to a variety of metrics (including retweet count), have been investigated. Kwak et al., have ranked users by the number of followers and by PageRank, and found the two rankings to be similar. They have analyzed the tweets of top trending topics and reported on the temporal behavior of trending topics and user participation. [46] have examined three types of influential users, performed in propagating popular news topics. Hansen et al. investigated the features of tweets that garner large numbers of retweets, analyzing a dataset of 210,000 tweets about the 2009 United Nations Climate Change Conference, as well as a random sample of about 350,000 tweets from 2010 [89]. [93], studied the dynamics of user influence across topics and time, as well as the problem of predicting the popularity of messages as measured by the number of future retweets. The study was conducted by classifying tweets in four categories according to the number of retweets they received (0, $\leq$ 100, [100, 9999], $\geq$ 10,000), formulating the prediction task as a classification problem. Moreover, they used a multi-class classifier, training it on one week and testing it on the next week for creating a short-term prediction. [133] used a similar technique to predict the probability that a tweet receives any retweets. They proposed a predictive model to forecast the likelihood for a given tweet of being retweeted, based on its contexts; furthermore, they deduced what are the most influential features that contribute to the likelihood of a retweet on the basis of the parameters learned by the model. In the work of [175], a number of features that might affect the probability of tweets to be retweeted ('retweetability', e.g., retweet proneness of a tweet) has been examined by using the principal component method and logistic regression models. The aim was the assessment of the probability of a tweet to be retweeted without assessing the degree of retweeting. Amongst the features that can be computed for each tweet, the presence of URLs and hashtags in the tweet body have been proved to present a strong relationship with retweetability. The experiment has been computed on a small dataset of 10 K observations, and the achieving prediction accuracy is not reported. [147] have defined the 'influence' as the ability of a user to spread information in a network, assuming that the retweet count may measure the popularity of a message on Twitter. The influence of a user could be also estimated by the average number of retweets collected by all tweets of the user. In that paper, the authors demonstrated by simulation that the probability to be retweeted is modeled by a power law function and the capacity of the most influential authors depends on their number of followers. [146] have proposed a model called retweet patterns (i.e., the retweet propagation trend). In that case conditional random fields have been used, taking into account three types of features: tweets features, users features and relationship features (which incorporates the perspectives whether the tweet may be simultaneously retweetable for two users). They have constructed the network relations for retweet prediction, and have demonstrated that conditional random fields can improve prediction effectiveness by incorporating social relationships, compared to those baselines that do not take into account such feature. [130] have computed both Naive Bayes and Support Vector Machine models considering two classes: tweets retweeted less than 30 times and tweets retweeted more than 100 times (massively retweeted tweets). The aim of their study was to detect those tweets that are massively retweeted in a short time, however without addressing the problem of predicting the potential number of retweets. They also used the principal component analysis to evaluate relevant features that could have an impact in detecting some retweeting proneness, without proposing a model for assessing the degree of retweeting, thus presenting only an exploratory descriptive approach. [207] have measured the popularity of a tweet through the time-series path of its retweets, by using a Bayesian probabilistic model. They have used the user ID of the original tweet and retweet authors, the number of followers and the word contained in tweets to predict the future retweets. Uysal and Croft [184] proposed a predictive model for estimating the likelihood of retweeting for a given user and tweet by using a logistic regression model. [202], used a factor graph model to investigate the retweeting behavior focusing on those features related to the user profile and to the content of a tweet. [41] focused their research on predicting the expected retweet count of a tweet by studying three types of features: content based features (presence or absence of hashtags), structure based features (as followers count, friends count, statuses count), as well as multimedia and image based features (the distribution of color intensities, perceptual dimensions, responses of individual object detectors). They have used the logarithm of retweet count for a given tweet as the response variable, and three different types of regression: linear, SVM with a Gaussian kernel, and Random Forest. The experiments produced better results with Random Forest, providing a RMSE score of 1.297 in log scale, very close similar performances have been obtained with SVM. They identified the Followers counts to be the most correlated feature. A common drawback found in content-based predicting tools reviewed in the literature, is represented by the 140-character constraint imposed by Twitter, which makes it difficult to identify and extract content-based predictive features [41]. [141] have treated the retweet prediction as a binary classification problem. They have used a multi-class classification for ranges of cascade sizes, in order to directly predict the logarithm of the retweets volume. For each day in the testing period, they have trained a Random Forest classifier to predict the future volume of retweets for tweets appearing on the day. The experiments have been compared by using the AUC (area under the precisionrecall curve) demonstrating the dependency of the model with respect to the user feature (e.g., followers counts), hashtag used popularity, user network features. [36], have provided a comparison of the performance for different learning methods and features, in terms of retweet prediction accuracy and feature importance, to understand what kind of tweets would be retweeted, by using as response variable a dummy variable representing the two states of being retweeted or not retweeted. They have found that Random Forests method archives the best performance. Moreover, they have found and included among the best features the following ones: number of times the user is listed by other users, number of followers, and the average number of tweets posted per day. On the same line, [100] and [209] have treated the retweeting behavior prediction as a binary classification problem, achieving an accuracy of 0.85 and 0.789 respectively. [114] have proposed a two-phase model to predict how many times a tweet can be retweeted in Sina Weibo microblog. In the first step, they have built a multi-classification model, while in the second step a regression model on each class has been constructed. They have achieved a high Mean Absolute Error of 58.22%, using the combination of Random Forest model and Least Median Squared Linear Regression model. However, discussion about the importance of each considered features is not reported. [68] have tried to consider user's different behaviors in different roles for the purpose of retweet prediction. They argue that the retweet prediction model might give better prediction accuracy results when the difference between the behavior of the author and retweetters is considered, determining the topic of interest of a user based on his past tweet and retweet.

6.3 Assessment framework for retweet modeling by using Twitter vigilance outcomes

According to the above presented state of the art, retweeting is a powerful mechanism to diffuse information on Twitter. The number of retweets of a tweet can be considered as a measure of how much the produced tweet has been effective in propagating the information, which is one of the major motivations for tweeting on Twitter.com. The proposed study aims at identifying the values of tweets' features which may determine the *degree of retweeting* and, as a side effect to understand the mechanisms which may determine retweeting in Twitter. The main goal is to create a predictive model for assessing the *degree of retweeting*, and thus to classify tweets in terms of certain classes for their *degree of retweeting.* The computational process at the end is performed through the following steps as depicted in Figure 6.1, and better described in the following sections:

1. Collection of the data from Twitter.com by crawling them by using Twitter Vigilance platform and tools on the basis *searches* and *channels.* The platform allows computing simple metrics for counting tweets/retweets for search and channel, extracting relationships among users, etc.

2. Selection of predictors/features from collected data and metrics.

3. Computation of potential predictors: a statistical criterion is applied to identify the statistically significant features. The use of an exploratory method is a crucial issue not only for ranking the variables before the construction of a prediction model, but also to give the phenomenon's first interpretation and to understand the underlying data structure.

4. Computation of a predictive model for the assessment of the binary probability to be retweeted or not.

5. Computation of a model to predict the *degree of retweeting*. The results have been obtained by comparing several different computational alternatives and approaches and selecting the better ranked and the most relevant metrics as described in the following.

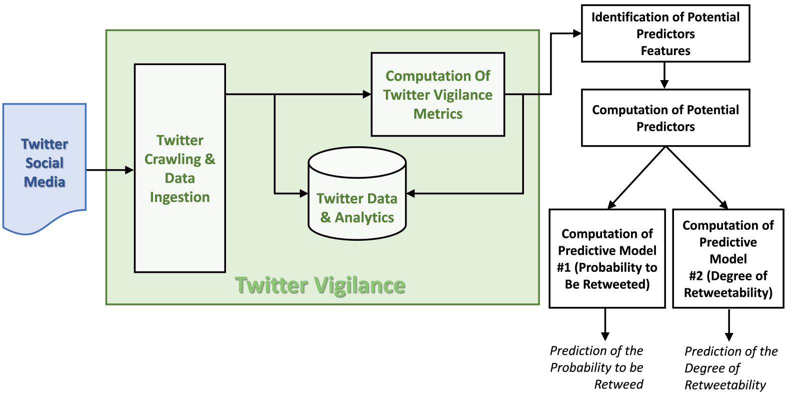

Twitter Social Media
Twitter Crawling & Data Ingestion
Computation Of Twitter Vigilance Metrics
Twitter Data & Analytics
Twitter Vigilance
Identification of Potential Predictors Features
Computation of Potential Predictors
Computation of Predictive Model #1 (Probability to Be Retweeted)
Computation of Predictive Model #2 (Degree of Retweetability)
Prediction of the Probability to be Retweed
Prediction of the Degree of Retweetability

Figure 6.1: Workflow of the overall process carried on by the proposed framework, from Twitter data ingestion to the computation of the predictive model

According to the previous statements, we have adopted Classification And Regression Tree (CART) models to understand the relevance of variables and to construct a model for predicting the probability to be retweeted and the *degree of retweeting*.

6.3.1 Collection of the datasets

Three datasets have been considered for the analysis. The first includes 100 Million of tweets (100M data set) related to 45 different Twitter Vigilance Channels covering many different topics but collected on the basis of a large number of search keys on Twitter.com API (which can be mainly related to terrorism, weather, mobility and transport, politics, city services, health and drugs, tourism and city, TV events, etc., see Figure 6.2 for details) from a larger set of 200 million data sets (as defined in Section 1.1.2, from April 2015 to June 2016). The second set includes 100,000 randomly selected tweets (100 K data set) from the 200 million dataset. The third includes 500,000 randomly selected tweets (500 K dataset) from the 200 million dataset. All data sets have been used to perform an exploratory analysis, a classification and a regression tree model. From the 100 M dataset, the 61% of the tweets are in English, the 12% in Italian, the 9% in Spanish and the remaining tweet are in many other languages. In Fig. 6.2, details of the distribution of collected posts are illustrated, showing the most numerous (covering almost 90% of the whole collected data set) search queries used for data ingestion (i.e. hashtags, citations, keywords etc.) grouped in their pertaining Twitter Vigilance channels; actually, as described in Section 1.1.2, a Twitter Vigilance channel can be considered as a thematic categorization of a set of semantically similar search queries. However, it is worthy to be noticed that the analysis and estimation of the degree of retweeting performed in this work are not dependent from the topic or subject.

6.3.2 Identification of potential features/metrics

In the second step, a set of features/metrics has been identified from the literature, by considering the information available on Twitter data, and by performing a qualitative analysis of twitter mechanisms by using a metric identification approach and methodology, such as GQM (Goal, Question, Metric). Such an approach has been followed considering that it would be desirable to identify metrics that may have some predictive capabilities in explaining the degree of retweeting.

The identified metrics are reported in Table 6.1, in which some metrics can directly refer to data and information contained in the single tweet, while other ones are derived from the author that has produced the tweet. A first set of metrics concerns the content of the tweet, and includes the number of Hashtags, Mentions and URLs contained in the message, the number of Favorites obtained by a tweet. A second set of metrics is about the tweet authors, and includes information regarding the user who posted the tweet: the number of days since the author created the Twitter account and the number of tweets posted since the creation of its own account (Statuses). A third set of metrics is related to network connected to the author: the number of users who follows the author of a tweet (Followers), the number of friends that author is following (Followees) and the number of other users that have listed the author in some of their own lists (Listed Count). A part of the identified metrics has been also used in [175], where a simple descriptive and Principal Component Analysis have been provided without deriving a predictive model. In [36], a comparative analysis of several methods has

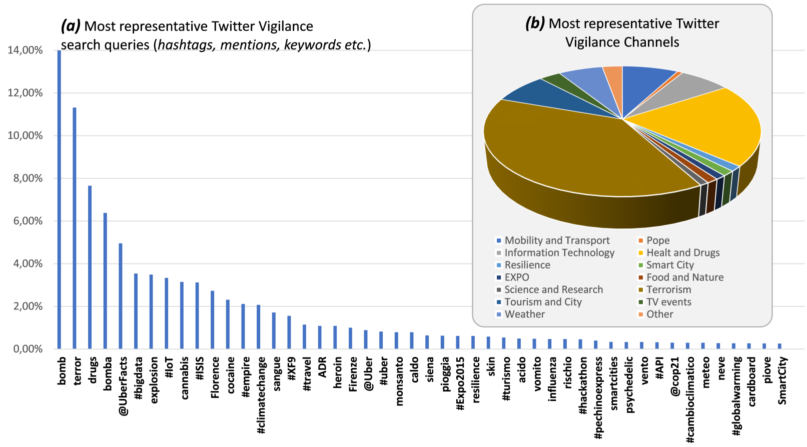


Figure 6.2: Distribution of collected posts dataset, showing the most frequent search queries (a), grouped by their pertaining Twitter Vigilance channels (b)

been proposed without considering all metrics we identified, and without addressing the prediction of the degree of retweeting. In the proposed analysis, we have specifically addressed metrics such as: Publication Time and Listed Count. The Publication Time metric should consider the classical claim stating that a higher probability of retweeting could be achieved if the tweet is published when the audience is on-line. The Listed Count metric should consider the reputation of the author, which is an additional level with respect to be just followed by another user. In addition to the metrics reported in Table 6.1, we also collected the Retweet Count (i.e., number of retweets obtained by the tweet), which can be considered, in our case, the target of our prediction models and not a real metric.

6.3.3 Computation and understanding of potential predictors

In the third phase, all the metrics have been extracted for the above-mentioned data sets. Figure 6.3 reports the percentage of the distribution of Retweet Count for the 100 million data set.

Tweet metrics	**Description**
URLs count	# of URLs in the tweet
Mentions count	# of mentions/citation of Twitter users in the tweet
Hashtags count	# of hashtags included in the tweet
Favorites count	# of favorite obtained by the tweet
Publication time	Local hour H24 in which the tweet has been published in the day according to the author' local time.
Author of tweet metrics	**Description**
Statuses count	# of days since the tweet's author created its Twitter account
Days count	# of tweets made by the tweet's author since the creation of its own account
Author network metrics	**Description**
Followers count	# of followers the author of the tweet
Followees count	# of friends the tweet's author is following
Listed count	# of people added the tweet's author to a list

Table 6.1: Considered features/metrics from the tweet information.

Then, Principal Component Analysis (PCA) has been applied. PCA is an exploratory technique for multivariate data, applied as a structure analysis method typically used to reveal the underlying structure that maximally accounts for the variance in datasets. The basic goal of PCA is to describe variations in a set of correlated variables, $x_T = (x_1, ..., x_q)$, in terms of a new set of uncorrelated variables, $y_T = (y_1, ..., y_q)$, each of which is a linear combination of x variables. The new variables are derived in decreasing order of importance in the sense that y_1 accounts for as much as possible of the variation in the original data amongst all linear combinations of x. Then y_2 is chosen to be uncorrelated with y1 and to account for as much as possible of the remaining variation, and so on. The new variables defined by this process, $y_1, ..., y_q$, are the principal components [63]. The first few components will account for a substantial proportion of the variation in the original variables, and they can be used to provide a lower-dimensional summary of these variables. To identify the optimal number of factors,

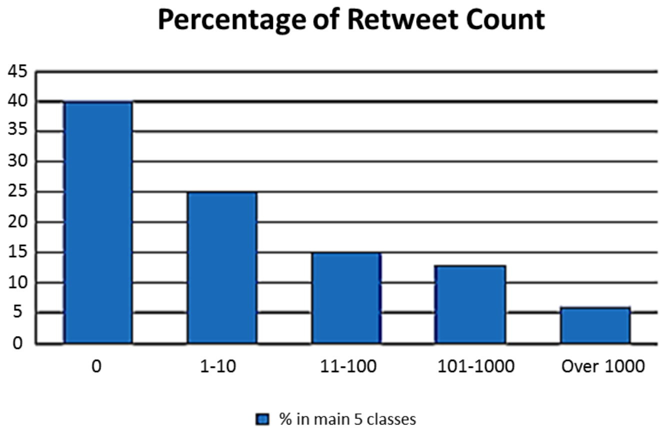


Figure 6.3: Percentage of the retweet count distribution in main 5 classes.

several informal and more formal techniques are available [102]. The most common procedures to choose the number of components/metrics to retain are the following:

- Retain just enough components to explain some specified large percentage of the total variation of the original variables. Values between 70% and 90% are usually suggested, although smaller values might be appropriate as q or n (the sample size) increases [63].
- The Kaiser criterion [103] recommends retaining only factors with eigenvalues greater than one.
- The screen test of Cattell [45], recommends plotting the eigenvalues and finding a place where the smooth decrease of eigenvalues appears to level off to the right of the plot. The number of components selected is the value corresponding to an 'elbow' in the curve, i.e., a change of slope.

PCA provides a first general idea about the internal structure of the data in a way that best explains the variance. PCA is performed on a representative random sample of 100 K observations with the eleven features (see Table 6.1), also including in this case the retweet count as performed by [175] on smaller number of variables. Table 6.2 reports the importance of factors extracted by PCA in descending order of variance. In the second column of Table 6.2, the eigenvalues that represent the variance for each factor are reported. corresponding percentage of the variance is shown in the third column of the table. With respect to our analysis on a 100 K tweet dataset, according to the Kaiser Criterion and to the screen test (see Fig. 6.4), the right number of principal components to be considered as relevant is five. The first five factors account for the 58.77% of the total variance. In [175], only 3 main PCA with an eigenvalue greater than 1 have been identified, explaining the 44,34% of the variance (Kaiser criterion), and considering only 10.000 tweets. In the work of [130], 4 main components have been identified, explaining the 56.34% of the variance considering 6 million of tweets, not sampled from a larger data set.

Factors	**Eigenvalue**	**% Variance**	**% Cumulative Variance**
1	1.9545	17.7681	17.7681
2	1.3748	12.4979	30.2659
3	1.0777	9.7976	40.0636
4	1.0335	9.3959	49.4594
5	1.0248	9.3164	58.7758
6	0.9623	8.7485	67.5243
7	0.9523	8.6576	76.1819
8	0.9339	8.4899	84.6717
9	0.7679	6.9808	91.6526
10	0.5976	5.4325	97.0851
11	0.3206	2.9149	100

Table 6.2: Importance of principal components.

In Table 6.3, the principal components loading for the features of Table 6.1 (plus *Retweet Count*) are reported. The component correlations of the original metrics are graphically depicted in Figures 6.5, 6.6, 6.7 and 6.8. Each feature in Table 6.2 is mapped into a vector in the factor map. The vector represents the correlation between the feature and the principal components (the axis of the graph). Factor 1 carries more than 17% of the total variability of the data set (Table 6.2), and this variability is mainly explained by the covariates *Favorite Count*, *Followers Count* and *Listed Count.* This first factor is strongly different with respect to the one identified by the Kaiser

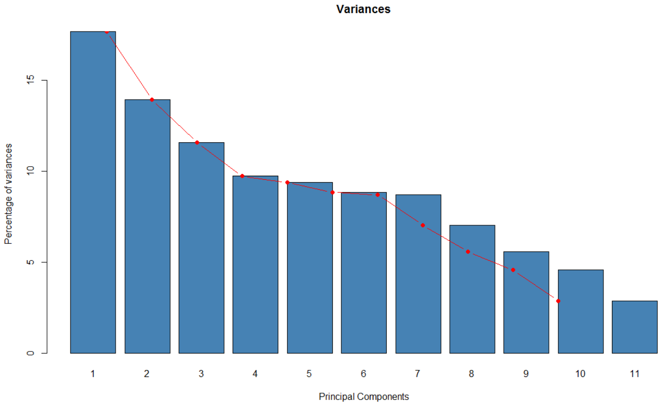


Figure 6.4: Distribution of the percentage of variance from PCA analysis.

criterion [103], since the *Listed Count* metric (which is dominant) was taken into account in that article.

Metrics	**PC1**	**PC2**	**PC3**	**PC4**	**PC5**
Retweet Count	−0.1623	0.4346	0.1635	−0.0026	−0.1009
Favorites Count	−0.6294	0.3908	0.1922	−0.1128	−0.1880
Followers Count	−0.7599	0.2736	0.0522	−0.0983	−0.0857
Followees Count	−0.1336	−0.0907	−0.4627	−0.2494	0.1182
Listed Count	−0.8431	−0.1549	−0.0498	0.1500	0.1871
Statuses Count	−0.4256	−0.5016	−0.3781	0.2795	0.2410
Hashtags Count	−0.1585	−0.5661	0.4377	−0.0517	0.0309
Mentions Count	0.0394	0.2194	0.0786	−0.1607	0.7697
URLs Count	−0.1288	−0.5483	0.2539	−0.3388	−0.3248
Publication Time	0.0076	−0.0728	0.3639	−0.5186	0.3707
Days Count	−0.0370	0.0070	−0.5072	−0.6604	−0.1691

Table 6.3: Principal component loadings.

The variability of Factor 2 (12.5%) is carried by the negative correlation of *Hashtags Count* (−0.5661) and *URLs Count* (−0.5483), while Factor 3 explains about 9.7% of the total variability, and it is represented by *Followees*

Count feature. Component 4 explains almost 9.3% of the total variability, and it is negatively correlated with the Publication Time of a tweet and the age of the author account (Days Count). Please note that also the *Publication Time* was not considered in Kaiser Criterion. The *Mentions* feature (0.7696) is mainly carried by Factor 5, and it explains the same proportion of variability of Component 4.

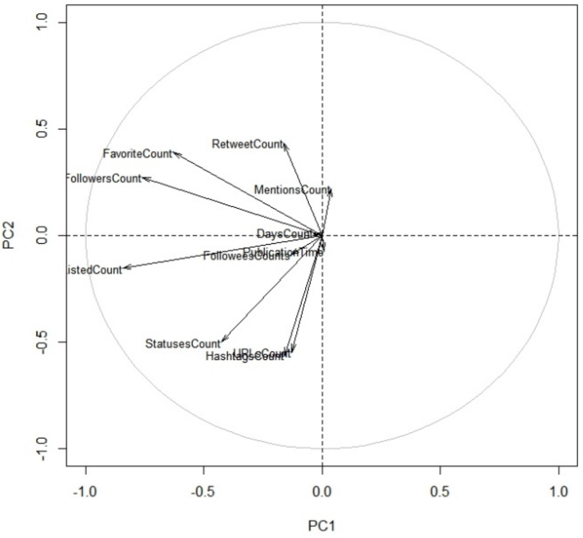


Figure 6.5: PCA factor map with factor 1 and factor 2.

PCA allowed to sort the features according to the impact on total variability, as well as to understand the correlation among the metrics and the number of retweets. According to the analysis results, the most relevant metrics are: *Mentions Count* (76.9% of Factor 5 total variability); *Listed Count* (explains the main variability of Factor 3 sharing it with *Followers* and *Favorite*); *Hashtags* (that explains the main variability of Factor 2, sharing it with *URLs Count*, *Statuses Count* and *Retweets Count*); *Days Count* (that explains the main variability of Factor 4, sharing it with *Publication Time*).

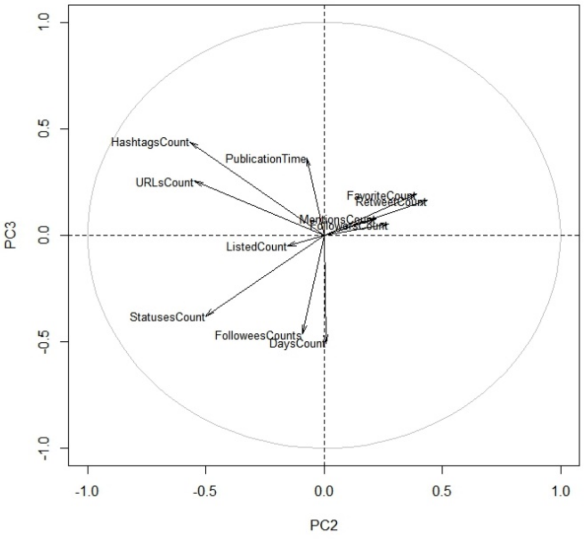


Figure 6.6: PCA factor map with factor 2 and factor 3.

6.4 Predicting the probability to be retweeted and the degree of retweeting of a tweet

In this section, before to present the analyses performed, a presentation of the considered classifications methods is provided. Then, the different analyses are reported. As a first phase, as reported in Section 6.4.2, a binary classification has been performed to create a model to identify tweets that have a probability to be retweeted, and thus the most relevant features that may determine the model. As a second phase, Section 6.4.3 presents the model for predicting the degree of retweeting of tweet. Also in this case, the most relevant features for the prediction have been identified.

6.4.1 Analysis of the considered classification methods

Classification Trees are machine-learning methods for constructing prediction models from data, and they have been widely used for the data exploration, description and prediction purposes. Trees have many properties,

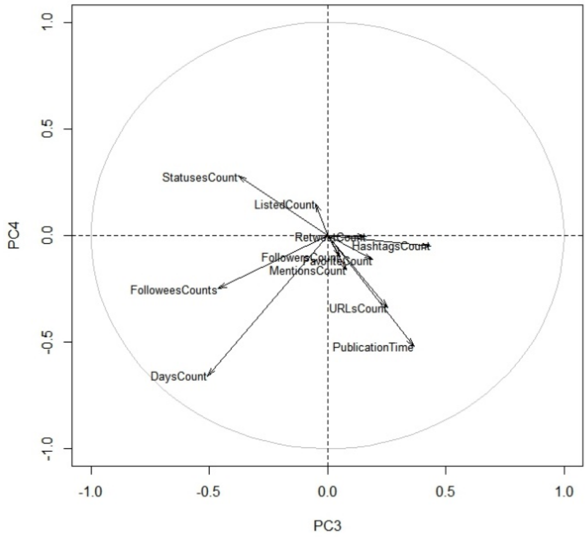


Figure 6.7: PCA factor map with factor 3 and factor 4.

including their ability to handle various types of response such as numeric, categorical, censored, multivariate, and dissimilarity matrices; trees are invariant to monotonic transformations of the predictors; complex interactions are modeled in a simple way; besides, missing values in the predictors are managed with minimal loss of information. Thanks to these properties, the use of classification and regression trees (i.e., a recursive partitioning method that is free from distributional assumptions), has potential advantages to construct predictive models. In this section, a short recall of the methods considered and compared for creating a suitable predicting model to estimate the degree of retweeting for single and/or groups of tweets is reported. Recursive partitioning procedure models (RPART) are defined by recursively partitioning the data space, and defining a simple local prediction model for each resulting partition. This can be represented graphically as a decision tree, with one leaf per partition [32]. The model can be written in

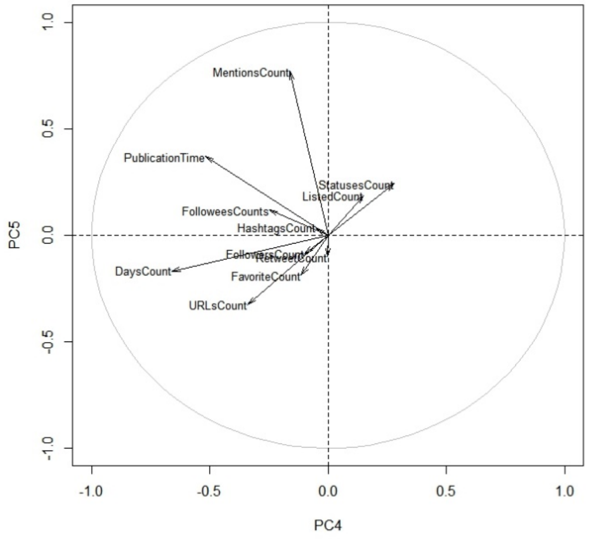


Figure 6.8: PCA factor map with factor 4 and factor 5.

the following form 6.1:

$$f(\mathbf{x}) = \mathbb{E}[y|\mathbf{x}] = \sum_{m=1}^{M} w_m \mathbb{I}(\boldsymbol{x} \in R_m) = \sum_{m=1}^{M} w_m \phi(\mathbf{x}, v_m) \tag{6.1}$$

where R_m is the m-th partition, w_m is the response in this partition, and v_m encodes the choice of variable to split on, together with the threshold value, on the path from the root to the m-th leaf. The best feature and the best value for that feature have been chosen by the split function 6.2:

$$(j^*, t^*) = \arg \min_{j\in\{1,..,D\}} \min_{t\in\mathcal{T}_j} cost\left(\{\mathbf{x}_i, y_i : x_{ij} \leq t\}\right) + cost\left(\{\mathbf{x}_i, y_i : x_{ij} > t\}\right) \tag{6.2}$$

In the classification setting, a multinoulli model has to be fitted to the data in the leaf satisfying the test $X_j < t$ by estimating the class-conditional probabilities $\hat{\pi}_c = dfrac1|\mathcal{D}| \sum_{k\in\mathcal{D}} \mathbb{I}(y_i = c)$, where $\mathcal{D}$ is the data in the leaf. Given the class-conditional probabilities, we have used the Gini index [81] to evaluate the partition: $\sum_{c=1}^{C} \hat{\pi}_c(1 - \hat{\pi}_c) = \sum_C \hat{\pi}_c - \sum_c \hat{\pi}_c^2 = 1 - \sum_C \hat{\pi}_c^2$ This index is the expected error rate $\hat{\pi}_c$ is the probability that a random entry in the leaf belongs to class c, and $(1 - \hat{\pi}_c)$ is the probability that it would be misclassified. To prevent overfitting, we have stopped the growth of the tree performing a pruning. This is performed by using a scheme that prunes the branches giving the least increase in the error [32]. A problem introduced by using recursive partitioning procedure is the fact that trees are unstable. One way to reduce the variance of an estimate is to average together many estimates using the bagging (bootstrap aggregating) technique. In the Random Forests approach [34] each tree is constructed using a different bootstrap sample from the original data. For each tree of the collection, a random subset of predictors is chosen to determine each split. In this way, the correlations between predictions of the individual trees are reduced. In other words, Random Forests try to decorrelate (each tree has the same expectation) the base learners by learning trees based on a randomly chosen subset of input variables, as well as a randomly chosen subset of data cases. In general, Random Forests procedure is better than bagging. Stochastic Gradient Boosting [74] is another way to reduce the variance. The algorithm for Boosting Trees evolved from the application of boosting methods. Boosting method [72] fits many large or small trees to reweighted versions of the training data, and performs classifications by weighted majority vote. In Stochastic Gradient Boosting, many small classification (or regression) trees are built sequentially from 'pseudo'-residuals (the gradient of the loss function of the previous tree). At each iteration, a tree is built from a random sub-sample of the dataset (selected without replacement) producing an incremental improvement in the model. An advantage of Stochastic Gradient Boosting is that it is not necessary to pre-select or transform predictor variables. It is also resistant to outliers. In general, boosting procedure outperforms the Random Forests. In the multinomial approach, trees are formulated as statistical models, alike generalized linear and additive models [51]. In this approach, splits are based on an explicit statistical model, the deviance of which defines the dissimilarity measure. For classification trees the use of a multinomial model is equivalent to the information index, with the deviance defined by the multinomial log-likelihood.

6.4.2 The probability to be retweeted

By following the line of [175] and [133], we have transformed the variable Retweet Count into a binary variable (0: no retweets, 1: one or more retweets). [175], fitted a Generalized Linear Model (GLM) to 10 K dataset, and used the results in a logistic equation to predict the probability of a retweet. [133], trained a prediction model to forecast the likelihood, for a given tweet, of being retweeted based on its contents. From the parameters learned by the model, they deduced which are the influential content features that contribute to the likelihood of a tweet to be retweeted. Our aim is to evaluate the relevant metrics associated to the action of retweeting in a predictive perspective: we used a learning approach to predict the probability for a tweet to be retweeted. The binary classification model provides us a general picture of the most important features (Table 6.1) related to retweeting. Given the finding that some features have strong relationship associated with the degree of retweeting, we have fitted the predictive models, presented in Section 6.4, on 500 K data set. In order to verify and validate the learned model parameters, we measure the accuracy of retweet prediction. Therefore, we split the set of tweets into a training and a test set. We have used about 80% of data for the training set, and 20% for the validation set. According to the results reported in Table 6.4, Random Forests is the best model in terms of accuracy (91.5%) and F_1score (90.61%). *Mentions Count* is the most relevant metric associated to retweeting in Random Forests, Recursive Partitioning and Gradient Boosting, while *Favorites Count* is the second one in all three models. In Multinomial (Logistic) Model, *Favorites Count* is the most important metric, followed by *Mentions Count.*

6.4.3 Predicting the degree of retweeting of a tweet

For the analysis of collected tweets, we conducted a 10-fold cross-validation evaluation on the complete **100 Million data set** and the features reported in Table 6.1. After the assessment of the above-mentioned approaches (as shown in the following), we have considered a CART model with Recursive Partitioning procedure (RPART model) as the best learning algorithm.

In the next section, a comparison of the above-mentioned methods is provided. In the considered predictive models the response variable Retweet Count has been transformed in a categorical variable, namely Retweet Class,

Classification methods	**Accuracy**	**Precision**	**Recall**	F_1**score**
Recursive partitioning	0.9071	0.9926	0.8157	0.8955
Random forests	**0.9150**	0.9826	0.8407	0.9061
Gradient boosting	0.9061	0.9936	0.8127	0.8941
Multinomial/Logistic model	0.9021	0.8115	0.9853	0.8899

Table 6.4: Retweet binary classification models comparison on 500 K data.

having classes: '0', '1−100', '101−1000', '1001−10,000', and 'Over 10,000', with the evident meaning of classifying the degree of retweeting, in 0 retweets, from 1 to 100 retweets, etc. Please note that the chosen classes are different from those of Fig. 6.3. Actually, classes '1−10' and '11−100', as depicted in Fig. 6.3, have been merged into a single size class '1−100'. In addition, we have created two new classes '1001−10,000' and 'Over 10,000', with the aim of understanding the degree of retweeting especially when the retweet count is high. As it will be described in the following, compacting classes '1−10' and '11−100' allowed us to obtain a higher accuracy (a better prediction model). Note that, the training set has been extracted as the 80% of 100 million data and the validation of the predictive capability has been performed on a test set of 20% of the total observations. According to the RPART approach, the CART models use a two-stage procedure. The resulting model can be represented as a binary tree. It should be noted that the resulting quality of most of the machine learning techniques is highly dependent on the calibration parameters. In our model, no optional classification parameters are specified, the Gini rule has been used for the splitting [165], according to which the prior probability is proportional to the observed data frequencies and the 0/1 losses are used. We used a cross-validation to choose the best value for the complexity parameter (CP). The 1-SE rule has been used to find the lowest cross-validation error as the sum between the smallest cross-validation error and the corresponding standard error. The results of RPART model statistics by class and the overall statistics are reported in Tables 6.5 and 6.6, respectively. The resulting accuracy of the predictive model is 68.15% and the precision is 85.64%, obtaining a satisfactory model for predicting the *degree of retweeting*. The kappa coefficient suggests that the level of agreement between the raters is discrete (see Table 6.6). The balanced accuracy (see Table 6.5) is very high for the first two classes, while it tends to decrease with the increasing degree of the retweeting classes.

The accuracy decrease is probably due to a lack of numerosity in the higher classes of retweet (Class: '1001−10,000', Class: 'Over 10,000') (see Fig. 6.3). Moreover, very high numbers of retweets are sporadic to be obtained, depending on many other factors, and less interesting for advertising and day by day activity of Twitter users. In fact, only the 6% over 100 Million of tweets obtain more than 1000 tweets. Typically, advertising campaigns are grounded on a large number of former tweets that collected less than 1000 retweets each. The classification performed also allows identifying when a tweet has low or null probability to be retweeted.

	Degree of Retweeting Classes				
Assessment Drivers	**0**	**0−100**	**101−1000**	**1001−10,000**	**Over 10,000**
Sensitivity	0.7737	0.8105	0.3142	0.0208	0.0136
Specificity	0.9132	0.6694	0.9199	0.9996	1.0000
Positive Predictive Value	0.8564	0.6256	0.3752	0.7345	0.8488
Negative Predictive Value	0.8579	0.8382	0.8975	0.9485	0.9915
Prevalence	0.4007	0.4053	0.1328	0.0526	0.0086
Detection Rate	0.3100	0.3285	0.0417	0.0011	0.0001
Detection Prevalence	0.3620	0.5251	0.1112	0.0015	0.0001
Balanced Accuracy	**0.8435**	**0.7399**	**0.6170**	0.5102	0.5068

Table 6.5: Predicting Class of degree of retweeting of the RPART procedure.

Figure 6.9 reports the features in order of importance in the prediction. The histogram suggests that the variable Mentions Count is the most correlated with the degree of retweeting. Furthermore, it has demonstrated to be the metric that better explains the volume of retweets. On the other hand, by eliminating the covariate Mentions Count from the model, the overall accuracy decreases to 0.5378, the precision to 0.5243, the recall equals to 0.6610

Assessment parameters	Values
Accuracy	0.6815
Accuracy 95% confidence interval (min, max)	(0.6813, 0.6817)
Recall	0.7737
Precision	0.8564
Kappa	0.4922

Table 6.6: Overall statistics in predicting class of degree of retweeting.

and Kappa index 0.2395. Table 6.7 reports the confusion matrix among the classes considered for the classification. From Table 6.7, it is also possible to understand how well the first two classes have been identified.

6.5 Comparison among different approaches

The choice of the RPART model has been justified by the fact that the accuracy obtained was higher than other ensemble learning techniques as Random Forests, Stochastic Gradient Boosting and Penalized Multinomial Regression. The comparisons have been performed by using the data sets of 100 K and 500 K tweets, due to the computational costs of some of the compared algorithms. Moreover, the recursive partitioning procedure is also the result of a compromise between goodness in terms of accuracy, simplicity in terms of interpretation (each tree derives from a series of logical rules [157]) and the ability to take into account of millions of data within a reasonable time frame.

Furthermore, RPART models can easily handle mixed discrete and continuous inputs, they are insensitive to monotone transformations of the inputs (because the split points are based on ranking the data points), they perform automatic variable selection, and they are relatively robust to outliers [132]. However, RPART model trees can produce models with high variance in the estimators. Two ways to reduce the variance of predictions could be adopted, for instance by using a bagging approach [33] or a boosting technique [162]: models like Random Forests often provide very good predictive accuracy. Actually, such an approach [34] aims at decorrelating the base learners by learning trees on the basis of a randomly chosen subset of input variables. Typically, the running time of classical Random Forests

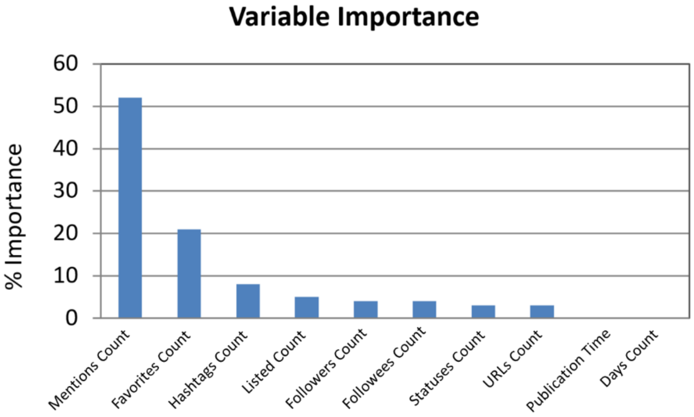


Figure 6.9: Variable Importance from the RPART model.

Degree of retweeting classes	Reference degree of retweeting classes				
	0	**0–100**	**101–1000**	**1001–10,000**	**Over 10,000**
0	31.0009	4.7219	0.3055	0.1487	0.0232
1-100	7.3885	32.8530	8.7785	2.9702	0.5240
101-1000	1.6765	2.9545	4.1732	2.0247	0.2941
1001-10,000	0.0005	0.0055	0.0258	0.1092	0.0077
Over 10,000	0.0000	0.0000	0.0000	0.0021	0.0117

Table 6.7: Confusion matrix of the RPART procedure.

technique is not viable for millions of observations. On the other hand, applying it on a 100 K tweet dataset does not provide relevant improvements in term of accuracy with respect to the recursive partitioning procedure.

The F_1score has been used to measure the models performance, and four approaches have been followed to build the model. Table 6.8 presents the results of the classification model with Recursive Partitioning procedure (RPART), the Random Forests techniques, the Stochastic Gradient Boosting model and the Multinomial Regression model on 100 K observations dataset. Also in these cases, we have used about 80% of data for the training set, and 20% for the validation set. In the fourth column, the F_1score is reported. This is a measure to evaluate the robustness of a model for making predictions, as a compromise between precision and recall:

$$F_1 score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{6.3}$$

$$\begin{aligned} Precision &= \frac{(\#\ tweets\ classified\ into\ class\ i)}{(\#\ tweets\ classified\ as\ class\ i)} \\ Recall &= \frac{(\#\ tweets\ classified\ into\ class\ i)}{(\#\ tweets\ belonging\ to\ theclass\ i)} \end{aligned} \tag{6.4}$$

According to results reported in Table 6.8, the differences among the first three methods in terms of F_1 score (6.3) are minimal. Moreover, we should remark that the Mentions Count is the most relevant metric in all the models. Then, the second more relevant metrics in the model are Favorites Count for Recursive Partitioning, Hashtag Count for Multinomial Model, Followers Count for Random Forests, and Favorites Count for Gradient Boosting (see Fig. 6.10). Please note that the only first two metrics are the same in the RPART model on 500 K and RPART model on 100 M.

Classification methods	**Accuracy**	**Precision**	**Recall**	F_1**score**
Recursive partitioning	**0.6827**	0.8436	0.7806	0.8108
Random forests	0.6812	0.8509	0.7761	**0.8117**
Gradient boosting	0.6764	0.8547	0.7715	0.8110
Multinomial model	0.6480	0.8423	0.7275	0.7807

Table 6.8: Models comparison on 100 K observations. The recursive partitioning resulted as the better ranked in terms of accuracy.

On the other hand, Table 6.9 shows the comparison among the models working on a 500 K data set in terms of processing time for training. The higher value of overall accuracy among the models, as well as the constraint of working with millions of observations (which, consequently, conveys fast execution times as a requirement), have led us to choose the recursive partitioning technique as the better ranked (see Table 6.9). The experiments

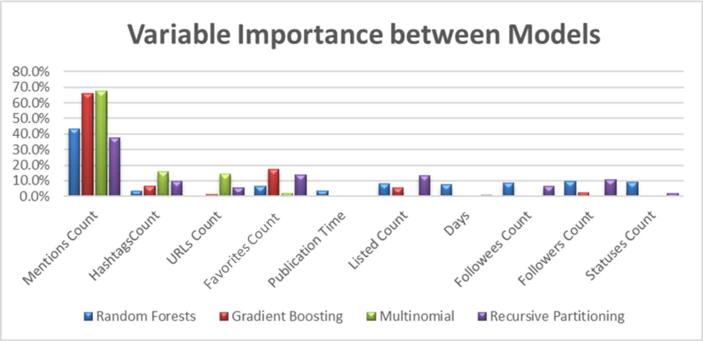


Figure 6.10: Variable Importance between models on 500 K data.

have been performed for the evaluation of the predictive models on a computational node with 98 GB Ram and 4 octa core CPUs (32 total cores, at 2.5 Ghz), using R which exploited only one core at time. Despite the lack of parallelization, the Recursive Partitioning approach resulted to be the most suitable to work on large data set, as 100 M or more.

6.6 Considerations

The work presented in this Chapter started with the aim of better understanding the correlation of features associated to tweets with respect to the action of retweeting. Most of the proposed papers in the literature proposed analysis without deriving models for predicting the *degree of retweeting*, in others they limited to identify the probability to be retweeted or not. The proposed analysis identified additional relevant metrics with respect to those proposed in the literature, namely, *Publication Time* and *Listed Count.* This approach resulted in obtaining a more effective principal component analysis and coverage of the phenomena. Therefore, on the basis of such an analysis, a method to predict the degree of retweeting through a classification trees model with recursive partitioning procedure applied on a data set of 100 Million of tweets has been proposed.

From the analysis results, the choice of the RPART model is justified by the fact that the accuracy is better with respect to Random Forests, Stochastic Gradient Boosting and Penalized Multinomial techniques, compared on a

Classification methods	**Accuracy**	**Precision**	**Recall**	F_1**score**	**Proc. time (sec)**
Recursive partitioning	0.6807	0.8512	0.7767	0.8122	180
Random forests	0.6884	0.8601	0.7866	0.8217	198,968
Gradient boosting	0.6796	0.8534	0.7731	0.8113	64,448
Multinomial model	0.6411	0.8367	0.7245	0.7765	31,576

Table 6.9: Retweet models comparison on 500 K data in terms of computation time in model estimation.

viable sample of 100 K observations. The Recursive Partitioning procedure is the result of a compromise between goodness in terms of accuracy, simplicity in terms of interpretation and the ability to take into account millions of observations within a reasonable time frame. By analyzing the results obtained with the Recursive Partitioning procedure, Mentions Count is the most correlated metric with the degree of retweeting, and the accuracy of the predictive model is about 68%.

The model produced can be used for assessing the degree of retweeting of each single tweet produced by some author or those prepared for advertising and/or for information campaign. Potential applications fields are many, including marketing and advertising, early monitoring, emergency response and, more generally, promoting and diffusing information; and the related raking and pricing of the actions performed in advertising.

Chapter 7

TV programme audience prediction

This chapter is focused on presenting the research results regarding the a set of metrics based on Twitter data have been identified and presented in order to predict the audience of scheduled television programmes, where the audience is highly involved such as it occurs with reality shows (i.e., X Factor and Pechino Express, in Italy). Identified suitable metrics are based on the volume of tweets, the distribution of linguistic elements, the volume of distinct users involved in tweeting, and the sentiment analysis of tweets. On this ground a number of predictive models have been identified and compared. The resulting method has been selected in the context of a validation and assessment by using real data, with the aim of building a flexible framework able to exploit the predicting capabilities of social media data.[1]

7.1 Introduction to TV programme audience predictions using Twitter

Social media analysis is becoming a very important instrument to monitor communities, users' preferences, and to make predictions. Among the social

[1] This chapter has been published as "Predicting TV programme audience by using twitter based metrics" in *Multimedia Tools and Applications Journal* [55].

media solutions, Twitter is one of the most widespread microblogs allowing users to have a personal news feed and followers attached to it. Followers receive some notification connected to the actions performed by the users they follow. Typical actions of users can be: posting a message (tweet), commenting, expressing like/favourite, retweeting (the echo of some tweet messages by some other users to the followers of the retweeting user). Therefore, tweets and retweets are shown (exposed) to other Twitter users, thus making more likely the chance of provoking their interests and reactions: retweets, comments, likes, etc. Some of these mechanisms can provoke viral processes that may lead to massive propagation of tweets in the user community. Twitter users are formally identified by '@' preceding their nickname. Any user may appeal to the attention of other users by including the @Twitter-username in the tweet. In the tweet text, every user can stress the attention to specific keywords called hashtags that are marked with '#' as first character. For example, hashtag: '#houseofcards' can be used to remark that the tweet is about the TV serial House of Cards (hashtags can be suggested to the audience by the TV producers, or spontaneously created by some users as well). Citations and hashtags are well indexed in Twitter.com and can be searched as main vehicles of involvement and remark, and thus are used by Twitter.com to propagate information to cited users and communities interested on following users or the hashtags, respectively.

Thanks to the above described social engagement mechanisms, a lot of users join and use Twitter every day; not only single users, but also news agencies, public institutions, producers, VIPs, teams, schools, municipalities, governments, etc., with the aim of sharing, promoting and communicating. On such grounds, Twitter is used as a source of information to deliver news, events, and innovations, and thus, it can be exploited as a tool for the prediction of different kinds of events and occurrences.

As described in the following, the research reported in this Chapter is about the usage of Twitter data to predict the attendance to TV shows by (i) computing metrics based on twitter data (volume of messages/posts including keywords (citation, hashtags) and/or mentions, volume of messages containing specific elements extracted from natural language processing (verbs, adjectives, words), and sentiment analysis by weighting each single text element on the basis of positive and/or negative moods), (ii) setting up and making in place predictive models also addressing feature selection. Thus, before passing to describe the solution proposed, the following subsection presents the related work.

7.1.1 State of the art

As previously stated, Twitter data have been used for setting up several kinds of predictive models in different domains according to the differences in the events and phenomena. In [170], a solution to predict football game results has been proposed by considering the volume of tweets. In more details the approach adopted defined a function for putting in correlation the delta changes in the volume of tweets with respect to a fixed number of categories, thus the obtained prediction rate was in the range of 68%. Opinion polls and predictions of political elections have been interrelated to the volume of tweets by using Sentiment Analysis techniques in [139]. In this case, the sentiment analysis has been performed by counting words and assigning to them negative or positive weights according to Opinion Finder lexicon based on only 2800 words, obtaining a highest correlation value of about 80% with respect to measures of public opinion derived from polls in the case of Obama elections. Voting results have been correlated with tweets in the 2009 German elections [183], addressing the counting of the tweets citing the different parties without providing a predictive model, another example can be found in [25]. In [77], sentiment analysis and volume approaches have been used for electoral prediction in the Senate competition which is 1:1, still obtaining correlations in the range of 40-60%.

Different models, based on both the volume of tweets and other means, have been also used for other predicting purposes: spread of contagious diseases [144] observing the inception over time of the adoption of terms which can be related to problems and symptoms that can be connected to specific illnesses. Other cases in the health domain have been studied for detecting the inception of public health seasonal flu [1], [110], [168], [35]. In economics, sentiment analysis has been adopted by employing Self-Organizing Fuzzy Neural Network, since long time series are present, predicting the direction of the stock market with a highest accuracy of over 86% [28]. Other cases in the market and business domains are described in [50], 43], for marketability of consumer goods in [166], and for book sales in [86].

With the aim of predicting box-office for movies, in [10] a model has been proposed adopting the average tweet rate, the presence of URLs in tweets and the volume of retweets as features. Also in this case, the time series are long (several days), and the model obtained an adjusted R squared of 0.94 via a linear model addressing sentiment analysis. Other cases in the same domain are: [111], [117], [127], [10], in which the combination of volume and sentiment analysis for long terms series has been proposed in a tool without proposing specific models. For example, in [97] the sentiment analysis is introduced by using the ratio from positive and negative score estimation of the tweets, obtaining accuracy of 64%. In [194], Twitter data have been used for predicting the performance of movies at the box office. To this end, a fuzzy inference system has been set up exploiting metrics such as the counting of tweets, followers, sentiment analysis metrics, and also additional information about the actors' rating according to the model proposed in [158]. The results presented on specific cases provide large mean square errors from 6% up to 27%.

Other applications highlighting Twitter data capabilities can be on: detecting crimes with the capability of identifying the inception of certain critical cases (such as micro discussions on crashes, fire, etc.) [193], places to be visited observing the most frequently attended places in a given location [47]. In addition, Twitter data has been used for assessing weather forecast information in [85], and in [84].

Twitter-based metrics have been used to estimate the number of people in specific locations like airports (the so called crowd size estimation) [29]. In this case, a simple linear model on the basis of volume metrics (i.e., number of tweets) has been proposed. In [80], the averaged value of past audience and Twitter data (contributions per minute) have been used for predicting audience (TV rating) on long series of political TV shows (from 14 to 280 shows), by using mainly volume metrics during broadcast time, and the rate of twitting people, obtaining an adjusted R squared of 0.95. On this regard, Nielsen Media Research discussed the capability of Twitter data to explain the variance of 2/3 of the difference in premiere audience sizes. TV rating is usually estimated sampling the audience with specific meters such as those installed by Auditel or more precise measures as those of Sky via set top box/decoders. In [94], a neural network approach has been used for predicting audience on the basis of Facebook data, obtaining a prediction accuracy in terms of Mean Absolute Percentage Error (MAPE) from about 6% to 24% on different TV shows. In [129], a number of TV shows have been analysed, clustering them for similarity, with the aim of identifying a predictive model for each cluster taking into account the Twitter data of previous days. The proposed predictive model is based on a linear regression (using volume and sentiment analysis metrics) that produced an R squared in range of 0.73-0.94 depending on the cluster. Typically, clusters with smaller amount of tweets in total per series are better ranked. A cross validation was not proposed to verify the robustness of the model. In those cases, very stable data and long series have been addressed. These series have a very different behaviour with respect to ‘reality TV shows’, in which there is a strong involvement of the audience in many phases of the show, and thus the number of tweets is much higher in the days before and massive in the day of the show. In [172], the authors discovered relevant correlations between the number of tweets passed 30 min before and after the show and in successive episodes without proposing a predictive model. In [191], a functional comparison of classical solutions for estimating TV show rating with respect to the TV data usage is proposed, together with an early solution for the estimation of TV rating based on textual, spatial, and temporal relevance, without proposing a predictive model.

According to the state of the art analysis, the predictive capabilities of Twitter data have been explained by using volume metrics on tweets (i.e., the total number of tweets and/or retweets associated with a Twitter user or having a given hashtag). However, in some cases a deeper semantic understanding of tweets has been required to create useful predictive capabilities. For these reasons, algorithms for sentiment analysis computation have been proposed to take into account the meaning of tweets via natural language processing algorithms (e.g., [139]). The adoption of techniques for segmenting, filtering or clustering by context (e.g., using natural language processing so as to avoid the misclassification of tweets related to the flu), or by users’ profiles (e.g., age, location, language, and genre) may help in getting more precise results in terms of predictability. Overviews of predictive methods exploiting tweets have been proposed in [169], and in [122]. Moreover, [122] have criticised the predictive capabilities of some proposed models based on Twitter data. In fact, some approaches proposed general models adopting specific filtering and/or classifications based on human assessors, thus reducing the replicability of the solution. Twitter data also present some problems due to the way they are ingested and collected. In particular, the access to the twitter API has some limitations such as: the maximum number of request calls in a period, the huge amount of tweets that can be produced for certain cases, the complexity of social relationships among users, the limited size of tweets (140 characters), and the fact that historical Twitter data are not accessible via the Twitter API, etc. These facts force the developers to set up specific architectures for collecting tweets, while attempting to get them with a sufficient reliability [140].

In [107], the trend of the dissemination information via Twitter has been analysed, observing the issues regarding the retweets cascade effect and the show count. Please note that the number of shows of a tweet is not easily accessible from Twitter data, but it is a well know observable metric exposed by internal Twitter analytic. The research work has demonstrated that the counting of retweets and the number of shows do not have a strong correlation. With the aim of predicting the number of shows, a number of predictive metrics have been proposed, and in particular: number of followers, friends, favourites; number of times the user has been listed; number of posts; number of active days, etc.

7.2 Framework for quantitative prediction by using Twitter Vigilance outcomes

As shown in Section 7.1.1 Twitter data have a relevant and flexible predictive power, and generally, they lead to quantitative statistical predictive capabilities of several social targets of interest. Relations among social media data and predictive variables are a priori unknown. An analysis of Twitter data related with media shows audience has been proposed in the literature. In [80], averaged value of audience in past events and Twitter data (contributions per minute) have been used for predicting audience on successive political TV shows having long series of events; thus demonstrating a correlation between the volume of tweets and the audience. In [94], a neural network approach has been used for predicting audience on the basis of Facebook data. In particular, the number of posts, the number of shares, number of comments, etc. without entering in the context of the posts; thus demonstrating the possibility of predicting the rating/share by using a neural network approach. In [115], a very high level analysis of the twitter data related to TV programme has been proposed, showing that the degree of interaction on Twitter was correlated with X Factor programme and its evolution. The approach of using Twitter for TV programme analysis is also used by Nielsen for analyzing if Twitter is helping the audience or viceversa, deducing that the fact is related, "*the volume of tweets caused significant changes in live TV ratings among 29 percent of the episodes*" [138].

This research work aimed at identifying suitable predictive models to predict media show audience (number of people following the programme) by exploiting social media info for reality shows. The research meant also to verify their validity in terms of prediction performance. The prediction of the number of attendees of the TV program is a more precise measure with respect to the estimation of the rating, as in [94]. The rating can be affected by the presence of other competing TV programmes in the same time slots. In addition, the prediction of audience in short term TV shows such as reality shows is very relevant for the present kind of television.

The framework proposed in this research work aims at defining a reliable statistical methodology to exploit Twitter data. Predictions with social data are generally based on conversational flow metrics concerning the volume of tweets, as well as tweet content/text in terms of keywords, hashtags, mentions; and/or users' activity.

Thus, our identified Twitter based metric predictors can be classified into a number of main classes and estimated for each single *TwitterVigilanceChannel* and/or for each single search per day or per hour, or in total per event, and in particular the:

(1) volume/number of tweets (TW) and retweets (RTW) versus time; (2) volume/number of tweets or retweets containing a certain keyword, verb, adjective, hashtags, citation, etc., versus time; (3) total sentiment analysis scores, taking into account positive and/or negative scores for elements in the tweets and/or retweets, versus time; (4) linear compositions of previous point tweets volumes statistics versus time (e.g., the ratio between number of retweets divided by the number of the corresponding tweets); (5) calendar variables calculated since the time tweets and/or retweets have been released; (6) volume of unique users tweeting and/or retweeting versus time. Please note that the metrics based on retweets have to be counted considering only the number of retweets at that time and not those in the future (for example up to the day before with respect to the predictive day value).

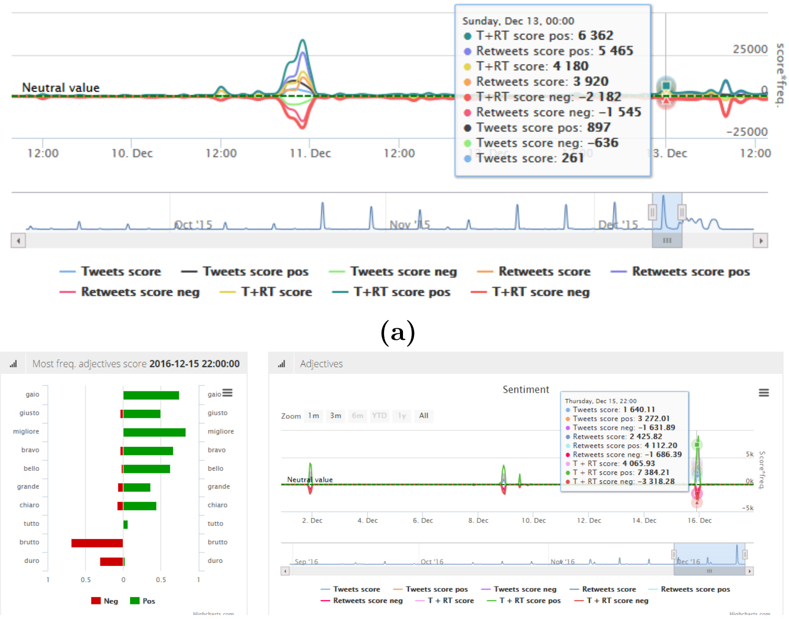
Moreover, it should be noted that the life cycle of retweet is limited in time. In the sense that according to the literature, almost all retweets are manifested in few minutes and sometimes few hours after the tweet, thus the number of those arriving after days can be neglected [206].

A detail of the Sentiment Analysis nd NLP manager interface is shown in Fig. Figure 7.2(a) reports the trends of the relevant sentiment analysis metrics over time. Please note that the most comprehensive metric 'R + RT score' (defined later), put together positive and negative trends highlighting the global positive/negative trend in time. In Fig. 7.2(a), an over-imposed arrow put in evidence the positive global value in that case. The reported metrics trends in Fig. 7.2(a) refer to a computation performed every hour on the basis of the last hour tweet and retweet collected on the channel, and in particular:

- (Tweet score pos) = Sentiment Analysis score for positive mood of Tweets;
- (Tweet score neg) = Sentiment Analysis score for negative mood of Tweets;
- (reTweet score pos) = Sentiment Analysis score for positive mood of reTweets;
- (reTweet score neg) = Sentiment Analysis score for negative mood of reTweets;
- (Tweet Score) = (Tweet score pos) + (Tweet score neg);
- (ReTweet Score) = (reTweet score pos) + (reTweet score neg);
- (T + RT score pos) = (Tweet score pos) + (reTweet score pos);
- (T + RT score neg) = (Tweet score neg) + (reTweet score neg);
- (T + RT Score) = (T + RT score pos) + (T + RT score neg).

The computation of the above presented sentiment analysis metrics is useful to detect the inception and position in time of relevant events as pikes. Once detected, the user can download the data table to estimate more complex and high level metrics (grounded on the above mentioned ones) which are more suitable for predicting the TV rating, as described in Section 7.2.1. In Fig. 7.2(b), the trends of the above listed sentiment analysis metrics, computed on the basis of the adjectives extracted in the tweets, are depicted.

The data acquisition is based on the Twitter Vigilance architecture presented in Section 1.1.2. The number of predictors that could be extracted depends on the Twitter data of the considered channel. Queries of popular keywords/searches on *TwitterVigilanceChannel* created for large events with many searches are very rich in information and complex to analyze. Many



(a)

(b)

Figure 7.1: Sentiment Analysis and NLP manager interface for an X Factor 9 event: **(a)** trend of the most relevant sentiment analysis metrics; **(b)** detail of Top-Sentiment rated Italian adjectives.

predictive models could be built; however, not all of them may have predictive capability, or the same effectiveness in predicting events, visitors and/or audience. The selection of predictors is crucial to build a reliable predictive model, on such grounds it is mandatory to identify predictors having a significant connection with the event which a prediction is needed for, with a reasonable temporal horizon. In order to build a reliable predictive model, the temporal dynamics explaining the predictive capability have to be identified. Predictive models and metrics show different behaviors when periodic or continuous events are considered. For example, the number of visitors during an event could show relevant or null relationship with calendar variables (as month, week day, year, etc.); while these variables are very important when the same attendee prediction is performed over uninterrupted and time-bound events of long term duration such as a long term event, a carnival, an exposition, etc.

7.2.1 Metrics definition and computation

A set of metrics that can be applied on XF9, XF10, and Pechino Express have been identified and reported below with their corresponding definition. The adopted metrics have been classified in: **volume metrics** when they are based on the volume of tweets or retweets; **NLP volume metrics** when the counting has been based on extraction of grammatical elements via natural language; **network metrics** when the counting is performed on the number of people involved in the community (the network); sentiment analysis metrics when the computation is based on the meaning and moods associated with words, verbs, adjectives, etc.; **high level metrics** are those that can be computed on the basis of other metrics with some non-linear function, such as the ratio between two metrics.

To take into account of the ratio from RTW e TW does not mean that for very high numbers of tweets the amount of retweets actually diminish the crowd/audience size, since the number of tweets and retweets are typically numerically balanced in absence of large viral events and audience as you can see from Twitter Vigilance platform. Furthermore, in presence of audience, the identified ratio (RTW/TW) is a measure of the reactivity, while as to measuring the volume metrics based on the total volume are more relevant. The ration of RTW/TW may lead to have large values if the event under monitoring becomes strongly viral, for example millions of retweets with only few tweets. This was not the case in general in all the three data sets tested.

On the other hand, this is not the case in our kind of events.

The computation of Sentiment Analysis metrics has been performed by exploiting SentiWordNet [62], a semantic knowledge base specifically designed for Sentiment Analysis. SentiWordnet assigns sentiment scores to each extracted keyword in order to estimate the general sentiment polarity of collected tweets. SentiWordNet is a sentiment-enriched implementation of WordNet [126], a widely used lexical database of English nouns, verbs, adjectives and adverbs grouped into sets of cognitive synonyms (synsets). In SentiWordNet independent positive, negative, and neutral sentiment values (i.e., real numbers varying in the interval from -1 to 1) are associated with about 117 thousands of synsets. In order to carry out the analysis in both English and Italian languages, the SentiWordNet lexicon (which has been originally designed in English) has been automatically ported to an Italian version, on the basis of MultiWordNet [149], a resource which aligns WordNet English synsets to Italian ones, which can therefore be used to transfer sentiment polarity information associated to English words to Italian corresponding ones. For each single tweet/retweet, its overall polarity score is given by the sum of all the sentiment weighted keywords extracted in it.

Most of the above mentioned metrics can be estimated every 5 min, every hour or day, or on more days according to the objective of the assessment (see Fig. 7.2 for example). The Twitter Vigilance platform (Section 1.1.2) allows estimating a number of them daily and other hourly. In any case, the user may re-compute them with different granularity from a specific interface requested an ad-hoc task.

Definition of metrics for assessing the stream of tweets per search and channel:

1. **Volume Metrics**:

   - Total number of tweets of the main hashtag collected over the 5 days preceding the event:

     $$TWWeek\_z = \sum_{d=D-5}^{D-1} TW_z^d$$

     where TW_z^d is the number of tweets collected at day d, varying from $D-5$ to $D-1$, being D the day of the event.

   - Total number of tweets plus retweets of the main hashtag over the

5 days preceding the event:

$$TWRTWeek_z = \sum_{d=D-5}^{D-1} TW_z^d + RTW_z^d$$

where TW_z^d is the number of tweets and RTW_z^d the number of retweets collected at day d, varying from $D-5$ to $D-1$, being D the day of the event.

2. **High Level Volume Metrics**:

   - Ratio from the number of retweets and tweets collected over the 5 days preceding the event, is a sort of measure of the reactivity of the audience of visitors with respect to the conversation based on single tweet inside a *TwitterVigilanceChannel*:

$$RTWWeekRatio_z = \sum_{d=D-5}^{D-1} \frac{TW_z^d + RTW_z^d}{TW_z^d}$$

   where TW_z^d is the number of tweets and RTW_z^d the number of retweets collected at day d, varying from $D-5$ to $D-1$, being D the day of the event.

3. **Network Metrics**:

   - Measures the number of unique users who retweeted in the 5 days preceding the event:

$$UnqUserRTW_z = \sum_{d=D-5}^{D-1} Uu_{RTW}^d$$

   where Uu_{RTW}^d is the number of unique users involved in retweeting estimated at day d, varying from $D-5$ to $D-1$, being D the day of the event.

   - Measures the number of unique users who tweeted in the 5 days preceding the event:

$$UnqUserTW_z = \sum_{d=D-5}^{D-1} Uu_{TW}^d$$

where Uu_{TW}^{d} is the number of unique users involved in tweeting estimated at day d, varying from $D-5$ to $D-1$, being D the day of the event.

- The whole set of unique users involved in tweeting and/or retweeting in the 5 days preceding the event:

$$FUnqUsers_z = \sum_{d=D-5}^{D-1} Uu^d$$

where Uu^d is the number of unique users involved in tweeting and/or retweeting estimated at day d, varying from $D-5$ to $D-1$, being D the day of the event.

4. **NLP Volume Metrics**:

- Score taking into account tweets in the 5 days preceding the event, counting the occurrence of distinct nouns, adjectives and verbs:

$$NLPTWWeek_z = \sum_{d=D-5}^{D-1} (\sum_{n=1}^{N_{nns}} TW_{nns_z^{(d,n)}} + \\ + \sum_{a=1}^{N_{adj}} TW_{adj_z^{d,a}} + \sum_{v=1}^{N_{ver}} TW_{ver_z^{d,v}})$$

where $TW_{nns_z^{(d,n)}}$, $TW_{adj_z^{d,a}}$ and $TW_{ver_z^{d,v}}$ are the total occurrence counts of, respectively, a generic noun n, a generic adjective a and a generic verb v extracted from collected tweets at day d, varying from $D-5$ to $D-1$, being D the day of the event. N_{nns}, N_{adj} and N_{ver} are the total number of distinct nouns, adjectives and verbs, respectively, extracted in tweets collected in the same temporal window.

5. **Sentiment Analysis Metrics**:

- Sentiment score taking into account all tweets in the 5 days preceding the event, adding the nouns, adjectives and verbs, each one weighted by its corresponding positive SA score:

$$SATWPosWeek_z = \sum_{d=D-5}^{D-1} (\sum_{n=1}^{N_{nns}} TW_{nns_z^{(d,n)}} \cdot ss_{pos^n} + \\ + \sum_{a=1}^{N_{adj}} TW_{adj_z^{d,a}} \cdot ss_{pos^a} + \sum_{v=1}^{N_{ver}} TW_{ver_z^{d,v}} \cdot ss_{pos^v})$$

where $TW_{nns_z^{(d,n)}}$ is the occurrence of a generic noun n with positive sentiment score ss_{pos^n} at day d; $TW_{adj_z^{d,a}}$ is the occurrence of a generic adjective a with positive sentiment score ss_{pos^n} at day d and $TW_{ver_z^{d,v}}$ is the occurrence of a generic verb v with positive sentiment score ss_{pos^v} at day d; these three metrics are computed for all the tweets collected in the 5 days preceding the event; N_{nns}, N_{adj} and N_{ver} are the total number of distinct nouns, adjectives and verbs, respectively, retrieved in tweets collected in the same temporal window.

- Sentiment score taking into account all tweets in the 5 days preceding the event, adding the nouns, adjectives and verbs, each one weighted by its corresponding negative SA score:

$$SATWNegWeek_z = \sum_{d=D-5}^{D-1} (\sum_{n=1}^{N_{nns}} TW_{nns_z^{(d,n)}} \cdot ss_{neg^n} + \\ + \sum_{a=1}^{N_{adj}} TW_{adj_z^{d,a}} \cdot ss_{neg^a} + \\ + \sum_{v=1}^{N_{ver}} TW_{ver_z^{d,v}} \cdot ss_{neg^v})$$

where $TW_{nns_z^{(d,n)}}$ is the occurrence of a generic noun n with negative sentiment score ss_{neg^n} at day d; $TW_{adj_z^{d,a}}$ is the occurrence of a generic adjective a with negative sentiment score ss_{neg^n} at day d and $TW_{ver_z^{d,v}}$ is the occurrence of a generic verb v with negative sentiment score ss_{pos^v} at day d; these three metrics are computed for all the tweets collected in the 5 days preceding the event; N_{nns}, N_{adj} and N_{ver} are the total number of distinct nouns, adjectives and verbs, respectively, retrieved in tweets collected in the same temporal window.

- Sentiment score taking into account all retweets in the 5 days preceding the event, adding the nouns, adjectives and verbs, each one weighted by its corresponding positive SA score:

$$\begin{aligned} SARTWPosWeek_z = & \sum_{d=D-5}^{D-1} (\sum_{n=1}^{N_{nns}} RTWnns_z^{(d,n)} \cdot ss_{pos^n} + \\ & + \sum_{a=1}^{N_{adj}} RTW_{adj_z^{d,a}} \cdot ss_{pos^a} + \\ & + \sum_{v=1}^{N_{ver}} RTW_{ver_z^{d,v}} \cdot ss_{pos^v}) \end{aligned}$$

  where $RTW_{nns_z^{(d,n)}}$ is the occurrence of a generic noun n with negative sentiment score ss_{neg^n} at day d; $RTW_{adj_z^{d,a}}$ is the occurrence of a generic adjective a with negative sentiment score ss_{neg^n} at day d and $RTW_{ver_z^{d,v}}$ is the occurrence of a generic verb v with negative sentiment score ss_{pos^v} at day d; these three metrics are computed for all the retweets collected in the 5 days preceding the event; N_{nns}, N_{adj} and N_{ver} are the total number of distinct nouns, adjectives and verbs, respectively, retrieved in tweets collected in the same temporal window.

- Sentiment score taking into account all retweets in the 5 days preceding the event, adding the nouns, adjectives and verbs, each one weighted by its corresponding negative SA score:

$$\begin{aligned} SARTWNegWeek_z = & \sum_{d=D-5}^{D-1} (\sum_{n=1}^{N_{nns}} RTW_{nns_z^{(d,n)}} \cdot ss_{neg^n} + \\ & + \sum_{a=1}^{N_{adj}} RTW_{adj_z^{d,a}} \cdot ss_{neg^a} + \\ & + \sum_{v=1}^{N_{ver}} RTW_{ver_z^{d,v}} \cdot ss_{neg^v}) \end{aligned}$$

  where $RTW_{nns_z^{(d,n)}}$ is the occurrence of a generic noun n with negative sentiment score ss_{neg^n} at day d; $RTW_{adj_z^{d,a}}$ is the occurrence of a generic adjective a with negative sentiment score

ss_{neg^n} at day d and $RTW_{ver_z^{d,v}}$ is the occurrence of a generic verb v with negative sentiment score ss_{pos^v} at day d; these three metrics are computed for all the retweets collected in the 5 days preceding the event; N_{nns}, N_{adj} and N_{ver} are the total number of distinct nouns, adjectives and verbs, respectively, retrieved in tweets collected in the same temporal window.

7.2.2 The overall process for model definition

The approach proposed to set up a predictive model includes the following steps:

1. Set up a *TwitterVigilanceChannel* semantically linked to the event in order to perform Twitter data harvesting. The creation of the channel is grounded on the official hashtags and Twitter users IDs, and relevant keywords. Other searchers to collect tweets can be added on the basis of the early analysis of the Twitter data, thus enlarging the set of searched queries on Twitter. This step is strongly dependent on the cases under analysis and described in Section 7.3.

2. Identify a first large set of possible metrics from early collected data, by using a coherent temporal basis of aggregation with respect to the real data values to be predicted (for example, volume of single channel query over time, unique users over time, calendar variables, natural language processing features, sentiment analysis features). In any case, the searches of the *TwitterVigilanceChannel* which collect a large number of tweets and retweets are typically significant and thus good potential predictors. Then, the time-series of metrics have to be merged to define a channel's *'guess metric matrix'*.

3. Select metrics: when metrics extracted from channel are too many, a statistical criterion may be applied to select the statistical significant metrics. For example, by using principal component analysis, PCA, which may give indication of the variance coverage and of complexity of data in terms of number of PCA to be considered. In addition, some early experiments adopting a multi-linear regressive schema may help with the support of the Akaike Information Criterion, AIC [2] in selecting/discharging the most/less significant metrics as predictors. The selection may be carried out by using stepwise process to build a

sharper model both discharging not reliable variables (by minimizing the AIC) and retaining the ones with a stronger linkage with variable to be predicted [189]. The statistical reliable predictors are defined as the ones having a *significant t-student test outcome* (*p-value* < 0.05). In alternative, machine learning approaches can be adopted, in any way the predictive capability, the adjusted R-squared and the AIC may help in deciding among the different methods. In most cases, the predictive model is produced by using the 70%-80% of data (e.g., estimating coefficient parameters, or learning parameters). Then the learned model is used to predict the remaining 30%-20% on which the MAPE (Mean Absolute Percentage Error), and or APE (Absolute Percentage Error) are estimated to perform the validation of the predictive mode against the actual values recorded by the auditing agencies.

In Fig. 7.2.2, the process for passing from data to prediction model is formalized. The process also presents a mapping of the above mentioned phases (from (1) to (3)), giving the evidence about what is performed by the TwitterVigilance tool, and what has to be performed by means of data analytics approaches described in the following. See for example Fig.7.2 which represents some trend for sentiment analysis metrics.

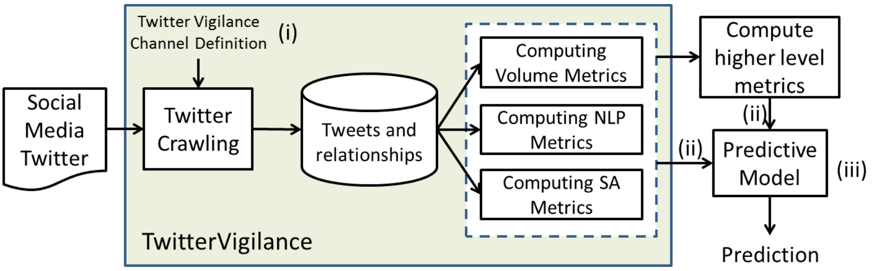


Figure 7.2: Overall process from Twitter data crawling to the computation of the prediction model.

7.2.3 Predictive models

TV programmes as reality show are in some sense short time events occurring with week periodicity and not for several weeks, thus concentrating the audience in few hours per week. Good examples of this kind of events are the so-called *reality shows*, such as: XF9, XF10 and Pechino Express, which are broadcasted live typically once per week (for a few hours), few weeks per year condensed in specific part of the year. Thus, a number of methods for creating predictive model for guessing the number of people following the show in the next week show has been considered and tested. The first method is a multi-linear regression model, that attempts to model relationships among explanatory variables/metrics $(z_1, z_2, ..., z_k)$ and a response variable x, all of them depending on t:

$$x_t = \beta_1 z_{1,t} + \beta_2 z_{2,t} + \beta_3 z_{3,t} + ... + \beta_{n1} z_{k,t} + n \tag{7.1}$$

The aim is to invert the model 7.1 by estimating $\beta_1, \beta_2, \beta_3, ..., \beta_k, n$, which represents the coefficients and the intercept of the best fitting line, respectively, obtained by a least squares model. In this process, the estimated model can be more or less significant and statistical significance can be estimated for each coefficient and for the whole fitting. Weights are estimated by means of a learning period, thus allowing targeting the model construction. Basically, several different models have been tested by estimating weights, and assessing predicting capabilities. In order to set up a predictive model, the value of x_t is estimated on the basis of explanatory variables/metrics $(z_1, z_2, ..., z_k)$ computed at $t - 1$ or before.

With many predictors and few observations in the data set, fitting the full model without penalization could result in large prediction intervals, and sometimes the model can over-fits the data: when there are issues with collinearity, the linear regression parameter estimates may become inflated. One consequence of large correlations between the predictor variances is that the variance can become very large. For this reason, a shrinkage/regularization model (i.e., ridge regression) has been tested [92], where it adds a penalty on the sum of the squared regression parameters. The effect of the penalty consists in the fact that the estimated parameters are allowed to become large only if there is a proportional reduction in sum of the squared errors (SSE). Thus, by adding the penalty, we are making a trade-off between the model variance and bias by sacrificing some bias, we can often reduce the variance enough to make the overall MSE (Mean Square Error) lower than unbiased models. In the selection of the best predictive model also other techniques have been tested such as lasso [178] and Elastic Net [212].

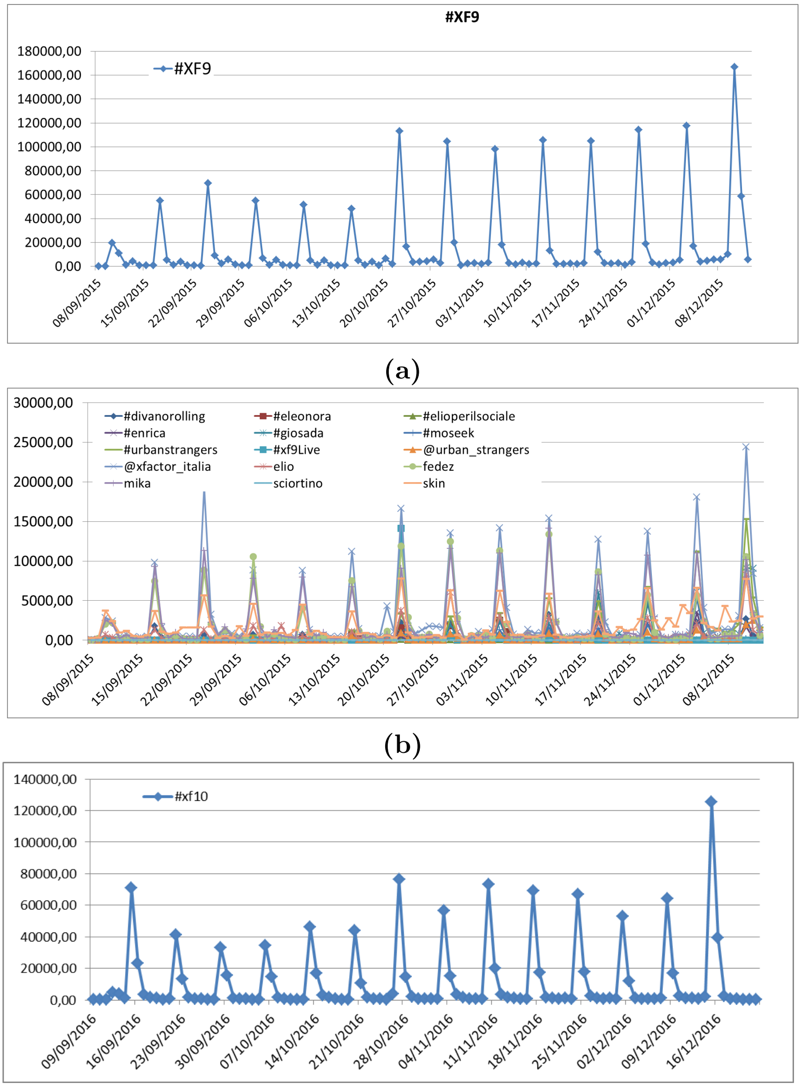
The following section refers to the prediction of the audience on TV programmes: X Factor 9, X Factor 10 and Pechino Express. For such reasons a suitable prediction model has been obtained by exploiting data from previous days using multi-regressive and ridge models. According to the above considerations, the reliable covariates we used have been individuated on the basis of their statistical relevance with respect to the variable to be predicted and by using a minimal AIC criterion [2]. The assessment quality of the models in terms of predictive capability has been performed against the validation period on the basis of the root mean square error (RMSE) and Mean Absolute Error (MAE) metrics that have been applied on the predicted values, as well as the correspondent ones that were observed during the validation/test period. The metric selection process has been carried out by approaching their incidence in exploiting the variable to be predicted in the multilinear regression model.

7.3 Predicting TV audience via twitter data

The adopted data refer to the last year seasons in the second part of the 2015 and 2016. About these events, the official actual data regarding the audience following those TV programmes have been published on Wikipedia and on the related official web sites. For example, for:

- XF9 description and actual audience data are accessible on: `https://it.wikipedia.org/wiki/X_Factor_%28nona_edizione%29`, while TwitterVigilance data can be accessed from: `http://www.disit.org/tv/index.php?p=chart_singlechannel&canale=Xfactor9`
- XF10 description and actual audience data are accessible on: `https://it.wikipedia.org/wiki/X_Factor_(decima_edizione)`, while TwitterVigilance data from: `http://www.disit.org/tv/index.php?p=chart_singlechannel&canale=xf10`
- Pechino Express description and actual audience data are accessible on: `https://it.wikipedia.org/wiki/Pechino_Express_%28quarta_edizione%29`, while TwitterVigilance data from: `http://www.disit.org/tv/index.php?p=chart_singlechannel&canale=ads`

In more details, X Factor is a television music competition format born in UK and then exported abroad, becoming the biggest television talent competition in Europe. In Italy the 9th season was televised (identified as XF9), from September to December 2015 and a season 10 in the 2016 with the first episodes devoted to auditions and singers' selections. The initial transmissions were followed by six weeks of weekly live shows where less appreciated singers have been progressively eliminated, thus, the best four talents could reach the final event where the winner was voted by the public. XF9 and XF10 have been broadcasted by pay-tv channel Sky1, while first phases and the final ones have been also transmitted on free of charge channels, i.e., national public television. The show began at prime time and closed after mid night with a shorter transmission called "Xtra Factor" to talk about the main show while always attracting the same audience. The audience of XF9 is typically based on young people, who are also engaged in voting singers and groups, so as to eliminate or push them ahead in the competition. As it occurs for every talent competition, the participation of the public is critical for the success of the show; social media play a relevant role in promoting singers, stimulating discussions and comments, while pushing audience to follow the show, voting their favourite singers and so on. Votes from the audience during the final broadcast of XF9 reached 7 million, and the official hashtag #xf9 was the most widely used of the day (on the 10th December, final show date) both in Italy and in the worldwide trending topic on Twitter. The competition has led to four finalists in December 2015: *Giosada*, *Urban Strangers*, *Davide Sciortino*, *Enrica Tara*, and the final selected *Giosada* has been the winner. A similar analysis could be performed for XF10.



(a)

(b)

(c)

Figure 7.3: Trends of twitters on Twitter Vigilance channels for XF9 and XF10: (a) trend of the whole #XF9 channel; (b) trends of the some of the channel XF9 less relevant searches, hashtags, mentions, keywords, etc. (among them also searches affected by relevant noise since connected to other meanings and not only to the TV show); (c) trend of the #XF10.

XF9/XF10 organization have prepared a wide and effective dissemination and marketing campaign also including social media, and thus Twitter accounts and hashtags reminding of the names of singers, and judges. Some of them have been proposed by the producers, while others have been spontaneously proposed by the audience and the social community. The initial Twitter hashtags were #XF9, #XTRA9 and later #xf9Live. Later on, some additional hashtags have been added to the above-mentioned keys, concerning the singers, such as: #UrbanStrangers, #eleonora, #giosada, #Enrica, etc. But concerning the judges/tutors, as well, and for specific cases as: @DivanoRolling, #divanorolling, #GiosadaAlBallottaggio, #Elio, #elioperilsociale, etc. XF9 cannel on Twitter Vigilance collected about 1.6 million of tweets in Italian language with hashtag #XF9. They have been mainly concentrated to the prime time; while smaller numbers have been collected over the days before the event (see Figure 7.3). The general volume of Twitter data in the XF9 channel resulted to be comprised of 43% of tweets and 57% of retweets. Similarly for XF10, we can see from Figure 7.3 that smaller volume of tweets have been detected, only the final event of the serial reached an audience comparable to those of season 9. According to Figure 7.2(b), the X Factor 9 channel presents a large number of other keywords, hashtags citations, etc. that presents a similar trend with respect to the main hashtag #XF9, and thus in some case add also a lot of noise. For example, those related to the judges that also provoke some tweet for their own activity not related to XF9. Metrics specifically related to those searches/keywords where discharge since less statistically relevant with respect to #XF9. The knowledge about the audience volume, and thus its prediction, can be very important when it comes to ads sale, which is delivered in the context of television programmes. Today, the ads value is only guessed since the measure of audience is obtained the day after, by Smart Panel Sky and/or Auditel in some cases (Auditel is the national metering of TV audience, could not provide measures of XF9 over 15 days in the period and on such basis it was not used as reference value). A similar case can be described for Pechino Express TV show. Figure 7.3 reports the trend Twitter data (TW+RTW) collected by *TwitterVigilanceChannel* and regarding Pechino Express. In this case, the trend is quite different: the number of attendees does not tend to increase aver the season, the last event does not attract a massive number of users.

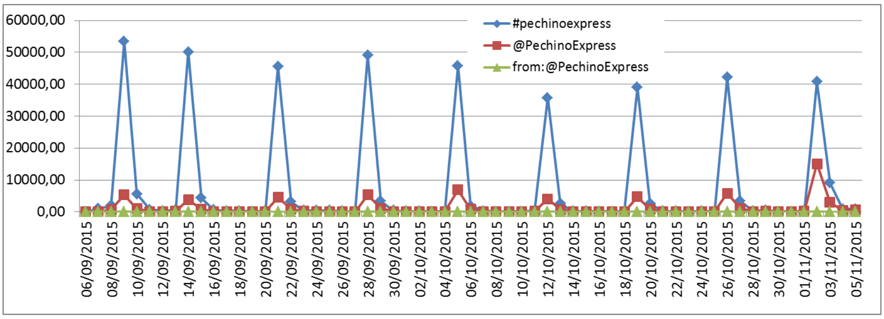


Figure 7.4: Trends of twitters on Twitter Vigilance channels for Pechino Express 2015.

7.3.1 Descriptive statistics

The Principal Component Analysis (PCA) is applied as exploratory technique for multivariate data and is used to reveal the underlying structure that maximally accounts for the variance in the data set. The basic goal of principal components analysis is to describe variation in a set of correlated variables, $x^T = (x_1, ..., x_q)$, in terms of a new set of uncorrelated variables, $y^T = (y_1, ..., y_q)$, each of which is a linear combination of the x variables. The new variables are derived in decreasing order of importance in the sense that y_1 accounts for as much as possible of the variation in the original data amongst all linear combinations of x. Then y_2 is chosen to account for as much as possible of the remaining variation, subject to being uncorrelated with y_1, and so on. The new variables defined by this process, $y_1, ..., y_q$, are the principal components [63]. The first few components take into account for a substantial proportion of the variation in the original variables and they be used to provide a lower-dimensional summary of these variables. Table 7.1 reports the importance of factors extracted by PCA in descending order of variance.

In the second column of Table 7.1 are reported the eigenvalues that represent the variance for each factor for XF9. The corresponding percentage of the variance is reported in the third column of the table. According to the Kaiser Criterion [103] that recommends to retain only factors with eigenvalues greater than one, the right number of principal components is three.

Table 7.1: Importance of components for XF9 data.

Factors	**Eigenvalue**	**% Variance**	*% Cumulative Variance*
1	2.63	53.26	53.26
2	1.92	28.44	81.71
3	1.15	10.15	91.85
4	0.86	5.72	97.57
5	0.46	1.61	99.18
6	0.23	0.41	99.59
7	0.18	0.25	99.83
8	0.12	0.11	99.94
9	0.07	0.04	99.98
10	0.05	0.02	100.00
11	0.01	0.00	100.00
12	0.01	0.00	100.00

The first five factors account for the 91.85% of the total variance. In Table 7.2, the principal component loadings for the X factor 9 features in Table 7.1 are reported. Factor 1 carries more than 53% of the total variability of the dataset (see metrics definition in Section) and this variability is mainly explained by the majority of covariates. The variability of Factor 2 (28.4%) is carried by the positive correlation of $RTWWeekRatio_z$ (0.8058) and the negative correlation of $SARTWPosWeek_z$ (-0.6526), while Factor 3 explains about 10.15% of the total variability. PCA allowed to sort the features according to the impact on total variability and understand the correlations among the metrics and the XF9 Sky audience.

7.3.2 Validation models

According to the above described data and cases, the first challenge was to identify a fitting model for XF9/XF10 and Pechino Express to validate the model consistency. The volume of data is characterized by a max of a dozen of sporadic events plus all days in the middle, and an explosive single event every week with a relevantly large audience for XF9 and XF10. Therefore, as a first step for XF9/XF10 and Pechino Express a fitting model has been identified by selecting the best metrics on the basis of PCA approach. As first approach, the multilinear regression model has been adopted in some cases to

Table 7.2: Principal Component loadings for XF9 data with respect to identified metrics.

Metrics and Data	**PC1**	**PC2**	**PC3**
Sky Audience	−0.1913	−0.4001	−0.7099
$TWRTWWeek_z$	−0.8745	0.4396	−0.1848
$TWWeek_z$	−0.8572	0.4846	−0.1485
$RTWWeekRatio_z$	−0.3462	0.8058	0.3857
$UnqUserTW_z$	−0.9241	0.2170	−0.1115
$UnqUserRTW_z$	−0.7276	0.6518	0.0693
$FUnqUsers_z$	−0.7607	0.6225	0.0707
$SATWPosWeek_z$	−0.8562	−0.3978	−0.1180
$SATWNegWeek_z$	−0.8439	−0.4174	−0.2269
$SARTWPosWeek_z$	−0.6261	−0.6526	0.2860
$SARTWNegWeek_z$	−0.5478	−0.5900	0.5665
$NLPTWWeek_z$	−0.8680	−0.4149	−0.1310
$NLPRTWWeek_z$	−0.6449	−0.5671	0.3206

estimate a model exploiting volume metrics of Twitter such as in [29]. [29] proposes a multilinear model to guess the number of people attending an event at the time of measure. Such approach does not lead to a predictive model, but rather to a model able to guess the volume of people in a given area at the current instant of measure on the basis of the Twitter data volume. Thus, in Table 7.1, an early model estimated by the multi-linear regression approach for XF9 and XF10 based by using volume and network metrics has been derived to confirm its validity in terms of structure with the aim of using the same metric set for both cases. From Table 3, the models present a satisfactory AIC and R-squared, while a less satisfactory adjusted R squared has been obtained for XF10a. In the case of XF9a, according to the p-value, some of the metrics are not significant (such as $FUnqUsers_z$ and $FUnqUsers_z$). If removed, slightly better results have been obtained producing the XF9b model as depicted in Table 7.5. In that case, an adjuster R squared of 0.768, with an AIC of 302, have been obtained. Please note that RMSE remain comparable among the three models. Therefore, the resulting models for XF9 and XF10 may be in principle very similar, obtaining similar results in terms of fitting. Note that in the case of XF9b, on the basis of the p-value, the metrics considered ($TWRTWWeek_z$, $UnqUserTW_z$ and $RTWWeekRatio_z$) are statistically significant.

Table 7.3: Parameters of the validation models using ridge approach with mixed metrics (volume, NLP and SA) estimated for XF9.

	XF9a Validation Model			
Metrics & Statistics	Coeff	Std Err	t-val	p-val
$TWRTWWeek_z$ (β_1)	161.2	144.1	1.119	0.314
$TWWeek_z$ (β_2)	-220.4	240.1	-0.918	0.401
$RTWWeekRatio_z$ (β_3)	-2190936	1308957	-1.674	0.155
$UnqUserTW_z$ (β_4)	-327.8	490.8	-0.668	0.534
$UnqUserRTW_z$ (β_5)	-99.16	670.1	-0.148	0.888
$FUnqUsers_z$ (β_6)	-5.461	617.1	-0.009	0.993
$Intercept$ (n)	5387852	2306725	2.336	0.067
R squared	0.867			
Adjusted R squared	0.707			
AIC	306			
RMSE	42159			
MAE	34244			
Weeks	12			
Millions of Tweets + Retweets on Twitter Vigilance	1.625			

According to results of AIC, R^2 and *p-values* of XF9a and XF9b validation models, volume and network based metrics alone seems to well explain the X factor audience. Please note that $RTWWeekRatio_z$ metric could lead to produce very large values depending on the kind of performance event. In the considered events, a total of 1.6 million of tweets have been collected, therefore the risk is not present and the metric is linearly dependent [145]. Thus the application of some solution for controlling the metrics is not needed.

For Pechino Express, the same set of metrics produced a similar model as reported in Table 7.6, in which a very similar Adjusted R squared and

Table 7.4: Parameters of the validation models using ridge approach with mixed metrics (volume, NLP and SA) estimated for XF10.

	XF10a Validation Model			
Metrics & Statistics	Coeff	Std Err	t-val	p-val
$TWRTWWeek_z$ (β_1)	999.6	788.1	1.268	0.260
$TWWeek_z$ (β_2)	-1489	1412	-1.054	0.340
$RTWWeekRatio_z$ (β_3)	-11342148	4477279	-2.533	0.052
$UnqUserTW_z$ (β_4)	2761	-2.323	0.068	
$UnqUserRTW_z$ (β_5)	-6655	2821	-2.359	0.065
$FUnqUsers_z$ (β_6)	6208	2726	2.277	0.072
$Intercept$ (n)	21546552	8072832	2.669	0.044
R^2	0.781			
Adjusted R^2	0.517			
AIC	310			
RMSE	50800			
MAE	42288			
Weeks	12			
Millions of Tweets + Retweets on Twitter Vigilance	1.383			

RMSE have been obtained, with a more satisfactory AIC. The data trend is in this case very linear as we can see from Fig. 6. As a result, the identified set of metrics for volume, unique users, and ratio are suitable for creating fitting models. Moreover, starting from the whole set of metrics reported in Section 3a, a mixed model taking into account also sentiment and NLP metrics has been obtained as reported in Table 7.7.

The model has been produced after testing several combinations of the metrics according to systematic approaches which allowed us to derive the best model in terms of AIC produced exploiting volume, NLP and sentiment analysis metrics (using both multilinear and ridge). Also in this case, according to the p-value, we could identify some less satisfactory metrics for XF9 data that may be good for XF10. Thus a compromise model fitting

Table 7.5: Parameters of the validation models according to the 7.1 using only volume and network based metrics for XF9 with a multilinear regression approach.

	XF9b Validation Model			
Metrics & Statistics	Coeff	Std Err	t-val	p-val
$TWRTWWeek_z$ (β_1)	15.19	5551	2.736	0.0256
$UnqUserTW_z$ (β_2)	-346.2	81.7	-4.237	0.0028
$RTWWeekRatio_z$ (β_3)	-1,505,184	382610	-3.934	0.0043
$Intercept$ (n)	4092413	612821	6.678	0.00015
R squared	0.832			
Adjuster R squared	0.768			
AIC	302			
RMSE	47408			
MAE	40745			
Weeks	12			
Millions of Tweets + Retweets on Twitter Vigilance	1.625			

Table 7.6: Parameters of the validation models according to the 7.1 using only volume and network based metrics for Pechino Express with a multilinear regression approach.

	PEb Validation Model			
Metrics & Statistics	Coeff	Std Err	t-val	p-val
$TWWeek_z$ (β_1)	-136.5	53,07	-2.573	0.062
$UnqUserRTW_z$ (β_2)	3175	1491	2.130	0.100
$FUnqUsers_z$ (β_3)	-1392	1082	-1.286	0.268
$Intercept$ (n)	2235653	112963	19.790	3.85E-05
R squared	0.877			
Adjuster R squared	0.785			
AIC	203			
RMSE	42747			
MAE	36453			
Weeks	8			
Millions of Tweets + Retweets on Twitter Vigilance	0.455			

Table 7.7: Parameters of the validation models using ridge approach with mixed metrics (volume, NLP and SA) estimated for XF9.

	XF9c mixed Validation Model			
Metrics & Statistics	Coeff	Std Err	t-val	p-val
$RTWWeekRatio_z$ (β_1)	-969524	354103	-2.738	0.041
$SATWNegWeek_z$ (β_2)	253.4	327.8	0.773	0.474
$SARTWPosWeek_z$ (β_3)	7.541	2.563	2.943	0.032
$SARTWNegWeek_z$ (β_4)	-4.489	7.064	-0.635	0.553
$NLPTWWeek_z$ (β_5)	-13.73	10.62	-1.293	0.252
$NLPRTWWeek_z$ ($beta_6$)	0.03587	0.2756	0.130	0.901
$Intercept$ (n)	3193367	647930	4.929	0.004
R squared	0.859			
Adjuster R squared	0.690			
AIC	306			
RMSE	43370			
MAE	33374			
Weeks	12			
Millions of Tweets on Twitter Vigilance	1.625			

satisfactory for both cases has been reported. The final model has been obtained with ridge approach, and the obtained adjusted R squared is of 0.69, and an R squared of about 0.86, having a suitable AIC of about 305 in both cases. Please note that, comparing Tables 7.3, 7.4, 7.6, 7.7 and 7.8, both multilinear and ridge approaches produced similar results. In some cases, the model based on volume metrics may be better ranked with respect to the mixed models in terms of adjusted R squared, and worst in terms of RMSE. In the next section, a wider comparison with other approaches is reported in the context of predictive models. For Pechino Express, the identical mixed model is not viable since the number of metrics (and thus the number of coefficients β_i to be estimated) is too high with respect the number of samples, thus producing an unstable model.

Table 7.8: Parameters of the validation models using ridge approach with mixed metrics (volume, NLP and SA) estimated for XF10.

	XF10c mixed Validation Model			
Metrics & Statistics	Coeff	Std Err	t-val	p-val
$RTWWeekRatio_z$ (β_1)	-2288390	899333	-2.545	0.051
$SATWNegWeek_z$ (β_2)	809.8	3.081	0.027	
$SARTWPosWeek_z$ (β_3)	73.66	-1.699	0.150	
$SARTWNegWeek_z$ (β_4)	310.6	98.05	3.168	0.025
$NLPTWWeek_z$ (β_5)	-73.77	19.37	-3.809	0.012
$NLPRTWWeek_z$ ($beta_6$)	3.97	2.378	1.669	0.156
$Intercept$ (n)	3193367	1646706	3.266	0.022
R squared	0.861			
Adjuster R squared	0.695			
AIC	305			
RMSE	40358			
MAE	31982			
Weeks	12			
Millions of Tweets on Twitter Vigilance	1.383			

7.3.3 Predictive models

According to the above described data the final challenge was to predict the audience attending the TV event in the prime time once a week. In the general framework, with the aim of creating a predictive model from a machine learning perspective, the last three weeks of the data have been used as test set and the remaining weeks have been used as training set. Four different approaches were tested, i.e., multi-linear regression (LM), ridge regression [92], lasso [178] and Elastic Net [212] applied on metrics adopted for XF9c, XF10c and PEb. The resulting comparison among such models for XF9c metrics is reported in Table 7.9.

According to these results the ridge regression approach has been proved to be the most accurate in prediction with respect to the above mentioned approaches. Therefore, models XF9c, XF10c and PEb (Pechino Express b model) produced by using the ridge approach, have been adopted as predictive models estimating coefficients on the basis of initial weeks data with the

Table 7.9: Comparison among predictive models considered in the case of XF9 data, APE and MAPE are estimated on the test prediction period on the basis of the model defined on the training data set.

	XF9 comparison of different pred. Models			
Prediction Errors and parameters	Lasso	Elastic net	Ridge reg.	LM
APE-week 11/6	0.2425	0.1173	0.0853	6 0.3456
APE-week 12/7	0.0907	0.1044	0.0429	0.1234
APE-week 13/8	0.3879	0.1837	0.2457	0.4257
MAPE	0.2403	0.1352	0.1246	0.2983
Training set	Weeks 1-10			
Test/prediction	Weeks 11-13			

aim of predicting the audience of the last 3 weeks major events in advance. The results are reported in Figs. 7.5 (a), (b) and (c), where the actual values are compared with the predicting values with confidence values. As a summary of the predictive models comparison, in Table 7.10, the mean absolute percentage error (MAPE) over the last 3 predicted weeks and the specific absolute percentage error (APE) have been reported for each of the predicted prime time audience (using the two approaches that provided best performance). Actually, MAPE is one the most widely used metrics for the assessment of prediction accuracy [131]. It should be noted that the precision in guessing the audience at the next and successive prime-time events on the basis of the model computed on data of weeks 1-10 is very high: for all cases in the range of 92%-95% of accuracy. On the other hand, the model is not capable to perform highly reliable predictions for the last event of season in which a strong non linearity occur. The general precision is in the range of 80%-94%. In the case of XF9 and XF10 the prediction on the last major event is less accurate with respect to Pechino Express, since XF9 and XF10 last live shows presented a quite explosive final event regarding the TV audience with respect to the Pechino Express. In fact, for the PE, the prediction of the 3rd week is still in the range of 95% since the last event is not massive as in the X Factor. As a general consideration, the prediction models identified are suitable to predict reality show audience in most of the cases. And thus the identified limitations of the state of the art algorithms and solutions have been overcome.

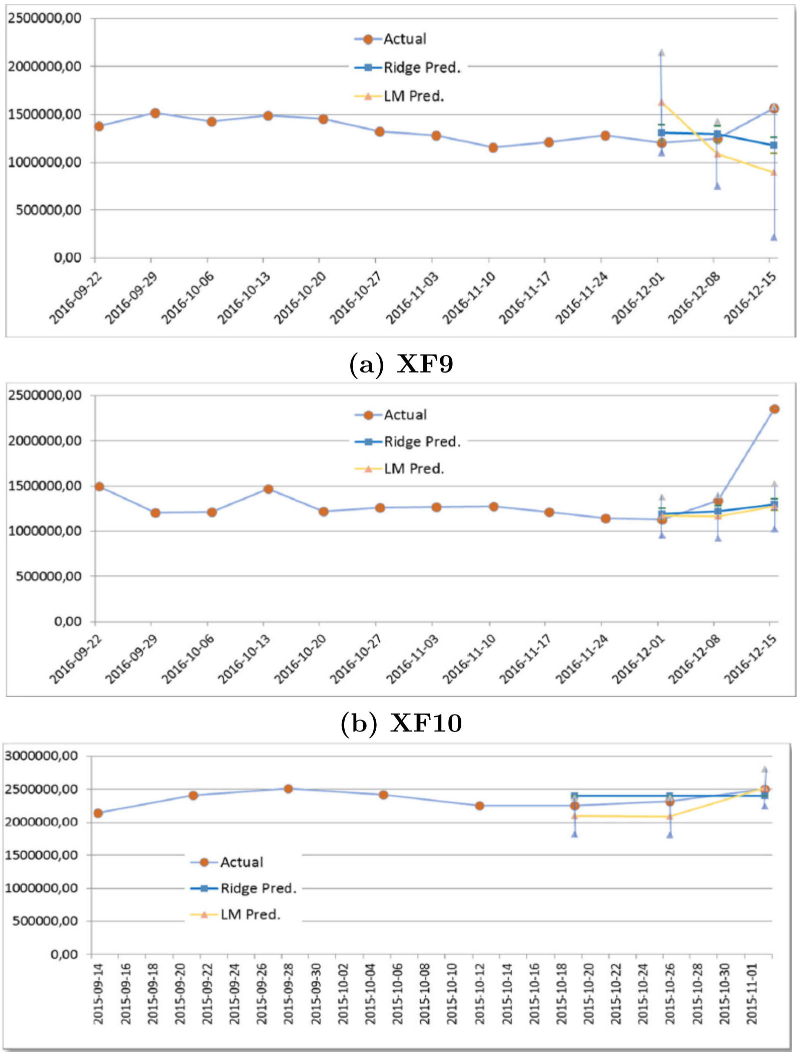
7.4 Considerations

This Chapter proposed an approach for creating Twitter-based models and metrics in order to predict the expected audience on television programmes. The proposed solution has been tuned by using reality shows, which are specific kinds of TV shows not addressed in the literature, and which present high volume of Twitter data due to the high involvement of audience in the trend of the programme by voting and interacting. Metrics identified have been: volume of tweets and retweets versus time; ratio between number of retweets divided by the number of the corresponding tweets; number of users involved in tweeting; natural language processing features extracted by Twitter data, and sentiment analysis assessment of tweets.

Table 7.10: Consumptive results about the prediction of attendees at TV programmes XF9, XF10 and Pechino Express on the basis of the predictive models XF9c, XF10c and PEb, using both multi-linear regression and ridge regression.

	XF9		XF10		Pechino Express	
Prediction Errors and Parameters	Ridge reg.	LM	Ridge reg.	LM	Ridge reg.	LM
APE-week 11/6	0.0853	0.3456	0.0511	0.0323	0.0670	0.0696
APE-week 12/7	0.0429	0.1234	0.0896	0.1327	0.0341	0.0998
APE-week 13/8	0.2457	0.4257	0.4479	0.4580	0.0412	0.0093
MAPE (11-13)/ (6-8)	0.1246	0.2983	0.1962	0.2077	0.0474	0.0596
Training set	Weeks 1-10				Weeks 1-5	
Test/prediction	Weeks 11-13				Weeks 6-8	

These metrics have been computed on the basis of data collected in the previous days and weeks, and they are capable to help predicting the TV rating of the prime-time show on the basis of the previously described predictive model. The Chapter reported full details about the method adopted to achieve the identification of the models and framework, and their validation by using real data. The produced predictive models have been validated and assessed in terms of quality, while highlighting the predicting capabilities for the analyzed cases, namely X Factor 9, X Factor 10, and Pechino Express. In all such cases, the predictive capability of the produced models according to the identified metrics has been proved. Moreover, a comparison among



(a) XF9

(b) XF10

(c) PE

Figure 7.5: Trend of actual and predicted values (for the last three events) for a XF9, b XF10 and c Pechino Express applied on basis of the predictive models XF9c, XF10c and PEb using training period as described in Table 7.9.

four different approaches has been presented: multilinear regression, ridge regression, lasso and elastic net. The ridge approach has been demonstrated to the better ranked approach. In almost all predictive models, metrics have been defined as the ratio between the number of retweets and tweets collected and related to the major hashtags of events and they have demonstrated high predictive capabilities in explaining visitors/audience volumes. Also the volume of tweets and the sum of tweets and retweets have confirmed their predictive capabilities. Another interesting predictor can be the number of unique users involved, as well as opinion mining features, such as natural language processing and sentiment analysis related metrics earlier described. As a result, the resulting models are based on ridge and/or multiregressive for short term prediction. On the other hand, other models and approaches have been tested without success, as reported in the Chapter. Most of the metrics based on Twitter data have been computed by Twitter Vigilance tool and provided directly to the users, while high level metrics have been computed for the model. Future work on this topic is related to the identification of other predictive and/or early detection models for different kinds of events, with the aim of producing better results with respect to those proposed in the literature. The specific topics would be: predicting politics election results, city comparison for tourism attraction, early detection of disasters, early detection of new drugs and/or critical situation in the city, etc. On the tool development aspects, we are addressing the development for improving the usability and the flexibility in computing metrics directly on the tool.

Chapter 8

Conclusions

This thesis summarizes the research work carried out at DISIT Lab (Distributed Data Intelligence and Technology Laboratory) of the Department of Information Engineering at the University of Florence as part of PhD research activity.

The connection, integration and analysis of the information produced by the various forms of data from smart cities, has led the necessity to the study of models capable to predict urban processes and simulate the likely outcomes of future urban development. One of the main aim of public administrations is become to provide a multitude of final applications based on the kind of user who requires a certain service, in particular through the use of real-time analytics to manage aspects of how a city functions and is regulated.

At the same time, social media have become an important communication tool and instrument for monitoring preferences of users. Many social media platforms, e.g., Twitter, allow rapid multimedia information diffusion, and thus they may be used as a source of information for viral advertising and marketing, early warning, emergency response and, more generally, for promoting and/or informing many users. The massive use of social media among the population has made possible to collect and analyze data in order to make predictions in many contexts.

To meet these needs, the research work is divided into two main parts: smart city solutions and social media data analysis.

The first part presents Smart City solutions to develop intelligent services for citizens and improve the lifestyle through the large amount of data coming from the city. Understanding of city users' behavior and studying the urban environment with the aim of offering citizens optimal solutions for their daily needs is one of the most challenging activities in a Smart City context.

On one hand, WiFi access points can be used to understand and predict the users' behavior in the city as presented in Chapter 2. A comparative analysis has shown that is possible to have a reasonable precision in assessing city behavior by access points positioning and collecting data from Wi-Fi. In the city of Florence a set of significant APs has been identified, and collected data from Wi-Fi have been analyzed though clustering techniques for grouping city users' usage trends in the day for each city area. The results have shown that about 12 different major clusters/patterns have been identified and each AP has been classified with respect to a cluster trend. The corresponding AP data, trend, and cluster have been used for predicting number of accesses and thus city usage, as a well as for detecting unexpected trends incepting in the different places of the city or to detect anomalies as early warning tool.

On the other hand, traffic and garages sensors can be used to manage the mobility needs as the prediction of the number of free parking lots in the city presented in Chapter 3. Looking for available parking slots is a serious issue in today urban sustainable mobility. The solution has been to provide suggestions to drivers about the parking availability 30 minutes and 1 hour in advance (thus producing a precise time stamp of which time they refer to) to allow their conscious decision-making process. To this end, the Bayesian Regularized Neural Network prediction model exploiting historical data, traffic flow sensors and weather data has demonstrated high predictive capabilities in explaining the number of free parking slots. With respect to the state of the art, the strength of the approach has been the advantage to be robust with respect to critical cases such as when the number of free slots reaches zero, or when some data are missing in the stream.

Another important aspect of smart cities is related to sustainable mobility and pollution. To make the city more enjoyable, a solution can be reducing the number vehicles in certain areas to increase the usage of public transportation in specific time slots, incentivizing ecological transportation choices and suggesting alternative public mean of transport (bus/tram) instead of the private car/motorbike. With this purpose, the real-time identification of a private transportation mode (car or motorbike) has a central role in personalized assistance messages delivery for sustainable mobility. In Chapter 4 has been presented a solution to create a classification system that uses mobile devices' sensor values and GIS data (user contextual information) to identify the transportation mean of users: stationary, walking, on a motorized private transport (car or motorbike) or in a public transport (tram, bus or train). The proposed solution overcome those of the literature since it presents a solution that is capable to produce reliable results in real conditions (i.e., real-time applications and background modality of operations) with a real set of devices. At the same time, the real-time monitoring of environmental and weather parameter is crucial in order to understand how much pollution affects the quality of the air that citizens breath. A system to carry out automatic real-time statistical data analysis from environmental sensors positioned in Smart Cities has been presented in Chapter 5. The environmental data collected from devices hosted by city users and from data providers, have been used to provide informative view to city users regarding environmental data thanks to the automatic creation of interpolation heat-maps.

The second part presents social media analysis and related tools. Twitter has revealed to be one of the most widespread micro-blogging services for instantly publishing and sharing opinions, feedbacks, ratings etc., contributing in the development of the emerging role of users as sensors. In Chapter 6, features extracted from Twitter data have been analyzed to create predictive models in order to predict the degree of retweeting of tweets (i.e., the number of retweets a given tweet may get), obtaining indications about the probable number of retweets a tweet may obtain from the social network. The solution of better understanding the correlation of features associated to tweets with respect to the action of retweeting had not yet been addressed in literature. In fact, most of the papers propose analysis without deriving models for predicting the *degree of retweeting* or they limited to identify the probability to be retweeted or not. Furthermore, Twitter is increasingly used as a source of real-time information about entertainment. In Chapter 7, suitable metrics based on the volume of tweets, the distribution of linguistic elements, the volume of distinct users involved in tweeting, and the sentiment analysis of tweets have been identified in order to create a prediction model of the scheduled television programmes' audience, where the audience is highly involved such as it occurs with reality shows (i.e., X Factor and Pechino Express, in Italy). In detail, these metrics have been computed on the basis of data collected in the previous days and weeks, and they are capable to help predicting the TV rating of the prime-time show on the basis of a predictive model.

Bibliography

[1] H. Achrekar, A. Gandhe, R. Lazarus, S.-H. Yu, and B. Liu, "Twitter improves seasonal influenza prediction." in *Healthinf*, 2012, pp. 61–70.

[2] H. Akaike, "Factor analysis and aic," in *Selected papers of hirotugu akaike*. Springer, 1987, pp. 371–386.

[3] H. Akima, *A method of bivariate interpolation and smooth surface fitting for values given at irregularly distributed points.* US Department of Commerce, Office of Telecommunications, 1975, vol. 75, no. 70.

[4] W. Alajali, S. Wen, and W. Zhou, "On-street car parking prediction in smart city: A multi-source data analysis in sensor-cloud environment," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage.* Springer, 2017, pp. 641–652.

[5] A. Alessandrini, C. Gioia, F. Sermi, I. Sofos, D. Tarchi, and M. Vespe, "Wifi positioning and big data to monitor flows of people on a wide scale," in *2017 European Navigation Conference (ENC).* IEEE, 2017, pp. 322–328.

[6] S. Allwinkle and P. Cruickshank, "Creating smart-er cities: An overview," *Journal of urban technology*, vol. 18, no. 2, pp. 1–16, 2011.

[7] S. An, B. Han, and J. Wang, "Study of the mode of real-time and dynamic parking guidance and information systems based on fuzzy clustering analysis," in *Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826)*, vol. 5. IEEE, 2004, pp. 2790–2794.

[8] K. Ashok and M. E. Ben-Akiva, "Alternative approaches for real-time estimation and prediction of time-dependent origin–destination flows," *Transportation Science*, vol. 34, no. 1, pp. 21–36, 2000.

[9] H. I. Ashqar, M. H. Almannaa, M. Elhenawy, H. A. Rakha, and L. House, "Smartphone transportation mode recognition using a hierarchical machine learning classifier and pooled features from time and frequency domains," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 244–252, 2018.

[10] S. Asur and B. A. Huberman, “Predicting the future with social media,” in *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01.* IEEE Computer Society, 2010, pp. 492–499.

[11] C. Badii, P. Bellini, D. Cenni, A. Difino, P. Nesi, and M. Paolucci, “Analysis and assessment of a knowledge based smart city architecture providing service apis,” *Future Generation Computer Systems*, vol. 75, pp. 14–29, 2017.

[12] C. Badii, P. Bellini, D. Cenni, A. Difino, M. Paolucci, and P. Nesi, “User engagement engine for smart city strategies,” in *2017 IEEE International Conference on Smart Computing (SMARTCOMP).* IEEE, 2017, pp. 1–7.

[13] C. Badii, P. Nesi, and I. Paoli, “Predicting available parking slots on critical and regular services by exploiting a range of open data,” *IEEE Access*, vol. 6, pp. 44 059–44 071, 2018.

[14] X. Ban, L. Chu, R. Herring, and J. Margulici, “Sequential modeling framework for optimal sensor placement for multiple intelligent transportation system applications,” *Journal of Transportation Engineering*, vol. 137, no. 2, pp. 112–120, 2011.

[15] X. J. Ban, R. Herring, J. Margulici, and A. M. Bayen, “Optimal sensor placement for freeway travel time estimation,” in *Transportation and traffic theory 2009: Golden jubilee.* Springer, 2009, pp. 697–721.

[16] A. Banerjee and R. N. Dave, “Validating clusters using the hopkins statistic,” in *2004 IEEE International conference on fuzzy systems (IEEE Cat. No. 04CH37542)*, vol. 1. IEEE, 2004, pp. 149–153.

[17] T. Bantis and J. Haworth, “Who you are is how you travel: A framework for transportation mode detection using individual and environmental characteristics,” *Transportation Research Part C: Emerging Technologies*, vol. 80, pp. 286–309, 2017.

[18] X. Bao, H. Li, L. Qin, D. Xu, B. Ran, and J. Rong, “Sensor location problem optimization for traffic network with different spatial distributions of traffic information,” *Sensors*, vol. 16, no. 11, p. 1790, 2016.

[19] M. Batty, K. W. Axhausen, F. Giannotti, A. Pozdnoukhov, A. Bazzani, M. Wachowicz, G. Ouzounis, and Y. Portugali, “Smart cities of the future,” *The European Physical Journal Special Topics*, vol. 214, no. 1, pp. 481–518, 2012.

[20] P. Bellini, M. Benigni, R. Billero, P. Nesi, and N. Rauch, “Km4city ontology building vs data harvesting and cleaning for smart-city services,” *Journal of Visual Languages & Computing*, vol. 25, no. 6, pp. 827–839, 2014.

[21] P. Bellini, I. Bruno, P. Nesi, and N. Rauch, “Graph databases methodology and tool supporting index/store versioning,” *Journal of Visual Languages & Computing*, vol. 31, pp. 222–229, 2015.

[22] P. Bellini, D. Cenni, and P. Nesi, “Optimization of information retrieval for cross media contents in a best practice network,” *International Journal of Multimedia Information Retrieval*, vol. 3, no. 3, pp. 147–159, 2014.

[23] P. Bellini, D. Cenni, P. Nesi, and I. Paoli, “Wi-fi based city users’ behaviour analysis for smart city,” *Journal of Visual Languages & Computing*, vol. 42, pp. 31–45, 2017.

[24] P. Bellini, P. Nesi, M. Paolucci, and I. Zaza, “Smart city architecture for data ingestion and analytics: Processes and solutions,” in *2018 IEEE Fourth International Conference on Big Data Computing Service and Applications (BigDataService)*. IEEE, 2018, pp. 137–144.

[25] A. Bermingham and A. Smeaton, “On using twitter to monitor political sentiment and predict election results,” in *Proceedings of the Workshop on Sentiment Analysis where AI meets Psychology (SAAIP 2011)*, 2011, pp. 2–10.

[26] F. Biljecki, H. Ledoux, and P. Van Oosterom, “Transportation mode-based segmentation and classification of movement trajectories,” *International Journal of Geographical Information Science*, vol. 27, no. 2, pp. 385–407, 2013.

[27] C. M. Bishop *et al.*, *Neural networks for pattern recognition.* Oxford university press, 1995.

[28] J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” *Journal of computational science*, vol. 2, no. 1, pp. 1–8, 2011.

[29] F. Botta, H. S. Moat, and T. Preis, “Quantifying crowd size with mobile phone and twitter data,” *Royal Society open science*, vol. 2, no. 5, p. 150162, 2015.

[30] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control.* John Wiley & Sons, 2015.

[31] D. Boyd, S. Golder, and G. Lotan, “Tweet, tweet, retweet: Conversational aspects of retweeting on twitter,” in *2010 43rd Hawaii International Conference on System Sciences.* IEEE, 2010, pp. 1–10.

[32] L. Breiman, J. Friedman, R. Olshen, and C. Stone, “Classification and regression trees (wadsworth, belmont, ca),” *ISBN-13*, pp. 978–0 412 048 418, 1984.

[33] L. Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[34] L. Breiman *et al.*, “Statistical modeling: The two cultures (with comments and a rejoinder by the author),” *Statistical science*, vol. 16, no. 3, pp. 199–231, 2001.

[35] D. A. Broniatowski, M. Dredze, M. J. Paul, and A. Dugas, “Using social media to perform local influenza surveillance in an inner-city hospital: a retrospective observational study,” *JMIR public health and surveillance*, vol. 1, no. 1, p. e5, 2015.

[36] H. Bunyamin and T. Tunys, “A comparison of retweet prediction approaches: the superiority of random forest learning method,” *Telkonika (Telecommun Comput Electron Control)*, vol. 14, no. 3, pp. 1052–1058, 2016.

[37] F. R. Burden and D. A. Winkler, “Robust qsar models using bayesian regularized neural networks,” *Journal of medicinal chemistry*, vol. 42, no. 16, pp. 3183–3187, 1999.

[38] F. Caicedo, “The use of space availability information in ’parc’ systems to reduce search times in parking facilities,” *Transportation Research Part C: Emerging Technologies*, vol. 17, no. 1, pp. 56–68, 2009.

[39] F. Caicedo, C. Blazquez, and P. Miranda, “Prediction of parking space availability in real time,” *Expert Systems with Applications*, vol. 39, no. 8, pp. 7281–7290, 2012.

[40] M. Caliskan, A. Barthels, B. Scheuermann, and M. Mauve, “Predicting parking lot occupancy in vehicular ad hoc networks,” in *2007 IEEE 65th Vehicular Technology Conference-VTC2007-Spring.* IEEE, 2007, pp. 277–281.

[41] E. F. Can, H. Oktay, and R. Manmatha, “Predicting retweet count using visual cues,” in *Proceedings of the 22nd ACM international conference on Information & Knowledge Management.* ACM, 2013, pp. 1481–1484.

[42] A. Caragliu and C. Del Bo, “Nijkamp, smart cities in europe,” in *Proceedings of the 3rd Central European Conference in Regional Science. Košice, Slovak Republic*, 2009, pp. 7–9.

[43] A. Caragliu, C. Del Bo, and P. Nijkamp, “Smart cities in europe,” *Journal of urban technology*, vol. 18, no. 2, pp. 65–82, 2011.

[44] E. Cascetta and M. N. Postorino, “Fixed point approaches to the estimation of o/d matrices using traffic counts on congested networks,” *Transportation science*, vol. 35, no. 2, pp. 134–147, 2001.

[45] R. B. Cattell, “The scree test for the number of factors,” *Multivariate behavioral research*, vol. 1, no. 2, pp. 245–276, 1966.

[46] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, “Measuring user influence in twitter: The million follower fallacy,” in *fourth international AAAI conference on weblogs and social media*, 2010.

[47] A. Chauhan, K. Kummamuru, and D. Toshniwal, “Prediction of places of visit using tweets,” *Knowledge and Information Systems*, vol. 50, no. 1, pp. 145–166, 2017.

[48] T. Chen, T. He, M. Benesty, V. Khotilovich, and Y. Tang, “Xgboost: extreme gradient boosting,” *R package version 0.4-2*, pp. 1–4, 2015.

[49] X. Chen, “Parking occupancy prediction and pattern analysis,” *Dept. Comput. Sci., Stanford Univ., Stanford, CA, USA, Tech. Rep. CS229-2014*, 2014.

[50] H. Choi and H. Varian, “Predicting the present with google trends,” *Economic Record*, vol. 88, pp. 2–9, 2012.

[51] L. Clark and D. Pregibon, “Tree-based models. in. chambers, jm and hastie, tj eds. statistical models in s, california: Wadsworth & brooks,” 1992.

[52] W. S. Cleveland, S. J. Devlin, and E. Grosse, “Regression by local fitting: methods, properties, and computational algorithms,” *Journal of econometrics*, vol. 37, no. 1, pp. 87–114, 1988.

[53] J. Connor and L. Atlas, “Recurrent neural networks and time series prediction,” in *IJCNN-91-Seattle international joint conference on neural networks*, vol. 1. IEEE, 1991, pp. 301–306.

[54] S. Contreras, P. Kachroo, and S. Agarwal, “Observability and sensor placement problem on highway segments: A traffic dynamics-based approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 3, pp. 848–858, 2015.

[55] A. Crisci, V. Grasso, P. Nesi, G. Pantaleo, I. Paoli, and I. Zaza, “Predicting tv programme audience by using twitter based metrics,” *Multimedia Tools and Applications*, vol. 77, no. 10, pp. 12 203–12 232, 2018.

[56] J. Dai, B. Yang, C. Guo, C. S. Jensen, and J. Hu, “Path cost distribution estimation using trajectory data,” *Proceedings of the VLDB Endowment*, vol. 10, no. 3, pp. 85–96, 2016.

[57] A. Danalet, M. Bierlaire, and B. Farooq, “Estimating pedestrian destinations using traces from wifi infrastructures,” in *Pedestrian and Evacuation Dynamics 2012*. Springer, 2014, pp. 1341–1352.

[58] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[59] J. Doblas and F. G. Benitez, “An approach to estimating and updating origin–destination matrices based upon traffic counts preserving the prior structure of a survey matrix,” *Transportation Research Part B: Methodological*, vol. 39, no. 7, pp. 565–591, 2005.

[60] H. Drucker, C. J. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in neural information processing systems*, 1997, pp. 155–161.

[61] J. C. Dunn, "A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters," 1973.

[62] A. Esuli and F. Sebastiani, "Sentiwordnet: A publicly available lexical resource for opinion mining." in *LREC*, vol. 6. Citeseer, 2006, pp. 417–422.

[63] B. Everitt and T. Hothorn, *An introduction to applied multivariate analysis with R.* Springer Science & Business Media, 2011.

[64] T. Fabusuyi, R. C. Hampshire, V. Hill, and K. Sasanuma, "A predictive model and evaluation framework for smart parking: The case of parkpgh," in *Proceedings of the 18th ITS World Congress, Orlando, FL, USA*, 2011, pp. 16–20.

[65] T. Fang and X. Hong, "Discovering meaningful mobility behaviors of campus life from user-centric wifi traces," in *Proceedings of the SouthEast Conference.* ACM, 2017, pp. 76–80.

[66] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[67] X. Fei, H. S. Mahmassani, and P. Murray-Tuite, "Vehicular network sensor placement optimization under uncertainty," *Transportation Research Part C: Emerging Technologies*, vol. 29, pp. 14–31, 2013.

[68] S. N. Firdaus, C. Ding, and A. Sadeghian, "Retweet prediction considering user's difference as an author and retweeter," in *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining.* IEEE Press, 2016, pp. 852–859.

[69] F. D. Foresee and M. T. Hagan, "Gauss-newton approximation to bayesian learning," in *Proceedings of International Conference on Neural Networks (ICNN'97)*, vol. 3. IEEE, 1997, pp. 1930–1935.

[70] M. S. Fox, "The role of ontologies in publishing and analyzing city indicators," *Computers, Environment and Urban Systems*, vol. 54, pp. 266–279, 2015.

[71] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American statistical Association*, vol. 97, no. 458, pp. 611–631, 2002.

[72] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *icml*, vol. 96. Citeseer, 1996, pp. 148–156.

[73] J. Friedman, T. Hastie, R. Tibshirani *et al.*, "Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors)," *The annals of statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[74] J. H. Friedman, "Stochastic gradient boosting," *Computational statistics & data analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[75] Y. Fukuzaki, M. Mochizuki, K. Murao, and N. Nishio, "A pedestrian flow analysis system using wi-fi packet sensors to a real environment," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication.* ACM, 2014, pp. 721–730.

[76] G. D. Garson, "Interpreting neural-network connection weights," *AI expert*, vol. 6, no. 4, pp. 46–51, 1991.

[77] D. Gayo-Avello, "A meta-analysis of state-of-the-art electoral prediction from twitter data," *Social Science Computer Review*, vol. 31, no. 6, pp. 649–679, 2013.

[78] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[79] R. Giffinger, C. Fertner, H. Kramar, R. Kalasek, N. Pichler-Milanović, and E. Meijers, "Smart cities: Ranking of european medium-sized cities. vienna, austria: Centre of regional science (srf), vienna university of technology," *www. smart-cities. eu/download/smart_cities_final_report. pdf*, 2007.

[80] F. Giglietto, "Exploring correlations between tv viewership and twitter conversations in italian political talk shows," *Available at SSRN 2306512*, 2013.

[81] C. Gini, "Measurement of inequality of incomes," *The Economic Journal*, vol. 31, no. 121, pp. 124–126, 1921.

[82] A. T. Goh, "Back-propagation neural networks for modeling complex systems," *Artificial Intelligence in Engineering*, vol. 9, no. 3, pp. 143–151, 1995.

[83] Z. Gong, "Estimating the urban od matrix: A neural network approach," *European Journal of operational research*, vol. 106, no. 1, pp. 108–115, 1998.

[84] V. Grasso, A. Crisci, M. Morabito, P. Nesi, and G. Pantaleo, "Public crowdsensing of heat waves by social media data," *Advances in Science and Research*, vol. 14, pp. 217–226, 2017.

[85] V. Grasso, I. Zaza, F. Zabini, G. Pantaleo, P. Nesi, and A. Crisci, "Weather events identification in social media streams: tools to detect their evidence in twitter," *PeerJ preprints*, vol. 4, p. e2241v1, 2016.

[86] D. Gruhl, R. Guha, R. Kumar, J. Novak, and A. Tomkins, "The predictive power of online chatter," in *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining.* ACM, 2005, pp. 78–87.

[87] A. Gyrard, A. Zimmermann, and A. Sheth, "Building iot-based applications for smart cities: How can ontology catalogs help?" *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 3978–3990, 2018.

[88] G. Hancke, B. Silva, G. Hancke Jr *et al.*, "The role of advanced sensing in smart cities," *Sensors*, vol. 13, no. 1, pp. 393–425, 2013.

[89] L. K. Hansen, A. Arvidsson, F. Å. Nielsen, E. Colleoni, and M. Etter, "Good friends, bad news-affect and virality in twitter," in *Future information technology.* Springer, 2011, pp. 34–43.

[90] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100–108, 1979.

[91] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *Proceedings of the 11th ACM conference on embedded networked sensor systems.* ACM, 2013, p. 13.

[92] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[93] L. Hong, O. Dan, and B. D. Davison, "Predicting popular messages in twitter," in *Proceedings of the 20th international conference companion on World wide web.* ACM, 2011, pp. 57–58.

[94] W.-T. Hsieh, T. C. Seng-Cho, Y.-H. Cheng, and C.-M. Wu, "Predicting tv audience rating with social media," in *Proceedings of the IJCNLP 2013 Workshop on Natural Language Processing for Social Media (SocialNLP)*, 2013, pp. 1–5.

[95] R. J. Hyndman and A. B. Koehler, "Another look at measures of forecast accuracy," *International journal of forecasting*, vol. 22, no. 4, pp. 679–688, 2006.

[96] J. Ivanchev, H. Aydt, and A. Knoll, "Routing choice information maximising robust optimal sensor placement against variations of traffic demand based on importance of nodes," 2015.

[97] V. Jain, "Prediction of movie success using sentiment analysis of tweets," *The International Journal of Soft Computing and Software Engineering*, vol. 3, no. 3, pp. 308–313, 2013.

[98] B. J. Jansen, M. Zhang, K. Sobel, and A. Chowdury, "Twitter power: Tweets as electronic word of mouth," *Journal of the American society for information science and technology*, vol. 60, no. 11, pp. 2169–2188, 2009.

[99] Y. Ji, D. Tang, P. Blythe, W. Guo, and W. Wang, "Short-term forecasting of available parking space using wavelet neural network model," *IET Intelligent Transport Systems*, vol. 9, no. 2, pp. 202–209, 2014.

[100] B. Jiang, J. Liang, Y. Sha, R. Li, W. Liu, H. Ma, and L. Wang, "Retweeting behavior prediction based on one-class collaborative filtering in social networks," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval.* ACM, 2016, pp. 977–980.

[101] S. Jiang, J. Ferreira, and M. C. González, "Clustering daily patterns of human activities in the city," *Data Mining and Knowledge Discovery*, vol. 25, no. 3, pp. 478–510, 2012.

[102] I. Jolliffe, *Principal component analysis.* Springer, 2011.

[103] H. F. Kaiser, "The application of electronic computers to factor analysis," *Educational and psychological measurement*, vol. 20, no. 1, pp. 141–151, 1960.

[104] S. Kaivonen and E. Ngai, "Real-time air pollution monitoring with sensors on city bus," *Digital Communications and Networks*, 2019.

[105] L. Kaufman and P. J. Rousseeuw, *Finding groups in data: an introduction to cluster analysis.* John Wiley & Sons, 2009, vol. 344.

[106] H. Kim, S. Baek, and Y. Lim, "Origin-destination matrices estimated with a genetic algorithm from link traffic counts," *Transportation Research Record*, vol. 1771, no. 1, pp. 156–163, 2001.

[107] A. Kupavskii, A. Umnov, G. Gusev, and P. Serdyukov, "Predicting the audience size of a tweet," in *Seventh International AAAI Conference on Weblogs and Social Media*, 2013.

[108] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th international conference on World wide web.* AcM, 2010, pp. 591–600.

[109] W. H. Lam, Z.-C. Li, H.-J. Huang, and S. Wong, "Modeling time-dependent travel choice problems in road networks with multiple user classes and multiple parking facilities," *Transportation Research Part B: Methodological*, vol. 40, no. 5, pp. 368–395, 2006.

[110] V. Lampos, T. De Bie, and N. Cristianini, "Flu detector-tracking epidemics on twitter," in *Joint European conference on machine learning and knowledge discovery in databases.* Springer, 2010, pp. 599–602.

[111] J. Leskovec, "Social media analytics: tracking, modeling and predicting the flow of information through networks," in *Proceedings of the 20th international conference companion on World wide web.* ACM, 2011, pp. 277–278.

[112] A. Lewis and P. Edwards, "Validate personal air-pollution sensors," *Nature News*, vol. 535, no. 7610, p. 29, 2016.

[113] X. Li, M. Ge, X. Dai, X. Ren, M. Fritsche, J. Wickert, and H. Schuh, "Accuracy and reliability of multi-gnss real-time precise positioning: Gps, glonass, beidou, and galileo," *Journal of Geodesy*, vol. 89, no. 6, pp. 607–635, 2015.

[114] G. Liu, C. Shi, Q. Chen, B. Wu, and J. Qi, "A two-phase model for retweet number prediction," in *International Conference on Web-Age Information Management*. Springer, 2014, pp. 781–792.

[115] M. Lochrie and P. Coulton, "Tweeting with the telly on!" in *2012 IEEE Consumer Communications and Networking Conference (CCNC)*. IEEE, 2012, pp. 729–731.

[116] E. Lovisari, C. C. de Wit, and A. Y. Kibangou, "Optimal sensor placement in road transportation networks using virtual variances," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 2786–2791.

[117] Y. Lu, R. Krüger, D. Thom, F. Wang, S. Koch, T. Ertl, and R. Maciejewski, "Integrating predictive analytics and social media," in *2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, 2014, pp. 193–202.

[118] J. T. Lundgren and A. Peterson, "A heuristic for the bilevel origin–destination-matrix estimation problem," *Transportation Research Part B: Methodological*, vol. 42, no. 4, pp. 339–354, 2008.

[119] W. Ma, X. Zhu, J. Huang, and G. Shou, "Detecting pedestrians behavior in building based on wi-fi signals," in *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*. IEEE, 2015, pp. 1–8.

[120] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural computation*, vol. 4, no. 3, pp. 448–472, 1992.

[121] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[122] L. Madlberger and A. Almansour, "Predictions based on twitter-a critical view on the research process," in *2014 International Conference on Data and Software Engineering (ICODSE)*. IEEE, 2014, pp. 1–6.

[123] V. Manzoni, D. Maniloff, K. Kloeckl, and C. Ratti, "Transportation mode identification and real-time co2 emission estimation using smartphones," *SENSEable City Lab, Massachusetts Institute of Technology, nd*, 2010.

[124] T. M. Martinetz, S. G. Berkovich, and K. J. Schulten, "'neural-gas' network for vector quantization and its application to time-series prediction," *IEEE transactions on neural networks*, vol. 4, no. 4, pp. 558–569, 1993.

[125] G. McLachlan, "Peel., d," *Finite Mixture Models*, 2000.

[126] G. A. Miller, *WordNet: An electronic lexical database.* MIT press, 1998.

[127] G. Mishne, N. S. Glance *et al.*, "Predicting movie sales from blogger sentiment." in *AAAI spring symposium: computational approaches to analyzing weblogs*, 2006, pp. 155–158.

[128] P. Misra and P. Enge, "Global positioning system: signals, measurements and performance second edition," *Global Positioning System: Signals, Measurements And Performance Second Editions,*, 2006.

[129] L. Molteni and J. P. De Leon, "Forecasting with twitter data: an application to usa tv series audience," *International Journal of Design & Nature and Ecodynamics*, vol. 11, no. 3, pp. 220–229, 2016.

[130] M. Morchid, R. Dufour, P.-M. Bousquet, G. Linares, and J.-M. Torres-Moreno, "Feature selection using principal component analysis for massive retweet detection," *Pattern Recognition Letters*, vol. 49, pp. 33–39, 2014.

[131] J. J. M. Moreno, A. P. Pol, A. S. Abad, and B. C. Blasco, "Using the r-mape index as a resistant measure of forecast accuracy," *Psicothema*, vol. 25, no. 4, pp. 500–506, 2013.

[132] K. P. Murphy, *Machine learning: a probabilistic perspective.* MIT press, 2012.

[133] N. Naveed, T. Gottron, J. Kunegis, and A. C. Alhadi, "Bad news travel fast: A content-based analysis of interestingness on twitter," in *Proceedings of the 3rd international web science conference.* ACM, 2011, p. 8.

[134] P. Nesi, C. Badii, P. Bellini, D. Cenni, G. Martelli, and M. Paolucci, "Km4city smart city api: an integrated support for mobility services," in *2016 IEEE International Conference on Smart Computing (SMARTCOMP).* IEEE, 2016, pp. 1–8.

[135] P. Nesi, G. Pantaleo, I. Paoli, and I. Zaza, "Assessing the retweet proneness of tweets: predictive models for retweeting," *Multimedia Tools and Applications*, vol. 77, no. 20, pp. 26 371–26 396, 2018.

[136] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights," in *1990 IJCNN International Joint Conference on Neural Networks.* IEEE, 1990, pp. 21–26.

[137] Y. Nie, H. Zhang, and W. Recker, "Inferring origin–destination trip matrices with a decoupled gls path flow estimator," *Transportation Research Part B: Methodological*, vol. 39, no. 6, pp. 497–518, 2005.

[138] Nielsen, "The follow-back: Understanding the two-way causal influence between twitter activity and tv viewership," 2013.

[139] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith, "From tweets to polls: Linking text sentiment to public opinion time series," in *Fourth International AAAI Conference on Weblogs and Social Media*, 2010.

[140] M. Oussalah, F. Bhat, K. Challis, and T. Schnier, "A software architecture for twitter collection, search and geolocation services," *Knowledge-Based Systems*, vol. 37, pp. 105–120, 2013.

[141] R. Pálovics, B. Daróczy, and A. A. Benczúr, "Temporal prediction of retweet count," in *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom)*. IEEE, 2013, pp. 267–270.

[142] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, 2013, pp. 1310–1318.

[143] P. Patil and A. Kokil, "Wifipi-tracking at mass events," in *2015 International Conference on Pervasive Computing (ICPC)*. IEEE, 2015, pp. 1–4.

[144] M. J. Paul and M. Dredze, "You are what you tweet: Analyzing twitter for public health," in *Fifth International AAAI Conference on Weblogs and Social Media*, 2011.

[145] E. A. Pena and E. H. Slate, "gvlma: Global validation of linear models assumptions," *R package version*, vol. 1, no. 0.2, 2014.

[146] H.-K. Peng, J. Zhu, D. Piao, R. Yan, and Y. Zhang, "Retweet modeling using conditional random fields," in *2011 IEEE 11th International Conference on Data Mining Workshops*. IEEE, 2011, pp. 336–343.

[147] F. Pezzoni, J. An, A. Passarella, J. Crowcroft, and M. Conti, "Why do i retweet it? an information propagation model for microblogs," in *International Conference on Social Informatics*. Springer, 2013, pp. 360–369.

[148] C. Pflügler, T. Köhn, M. Schreieck, M. Wiesche, and H. Krcmar, "Predicting the availability of parking spaces with publicly available data," *Informatik 2016*, 2016.

[149] E. Pianta, L. Bentivogli, and C. Girardi, "Multiwordnet: developing an aligned multilingual database," in *First international conference on global WordNet*, 2002, pp. 293–302.

[150] R. Piedrahita, Y. Xiang, N. Masson, J. Ortega, A. Collier, Y. Jiang, K. Li, R. P. Dick, Q. Lv, M. Hannigan *et al.*, "The next generation of low-cost personal air quality sensors for quantitative exposure monitoring," *Atmospheric Measurement Techniques*, vol. 7, no. 10, pp. 3325–3336, 2014.

[151] M. Piorkowski, N. Sarafijanovic-Djukic, and M. Grossglauser, "A parsimonious model of mobile partitioned networks with clustering," in *2009 First*

International Communication Systems and Networks and Workshops. IEEE, 2009, pp. 1–10.

[152] E. C. Polley and M. J. Van Der Laan, "Super learner in prediction," 2010.

[153] O. A. Popoola, D. Carruthers, C. Lad, V. B. Bright, M. I. Mead, M. E. Stettler, J. R. Saffell, and R. L. Jones, "Use of networks of low cost air quality sensors to quantify air quality in urban settings," *Atmospheric environment*, vol. 194, pp. 58–70, 2018.

[154] A. C. Prelipcean, G. Gidófalvi, and Y. O. Susilo, "Mobility collector," *Journal of Location Based Services*, vol. 8, no. 4, pp. 229–255, 2014.

[155] ——, "Transportation mode detection–an in-depth review of applicability and reliability," *Transport reviews*, vol. 37, no. 4, pp. 442–464, 2017.

[156] Z. S. Qian and R. Rajagopal, "Optimal parking pricing in general networks with provision of occupancy information," *Procedia-Social and Behavioral Sciences*, vol. 80, pp. 779–805, 2013.

[157] J. R. Quinlan, "Learning logical definitions from relations," *Machine learning*, vol. 5, no. 3, pp. 239–266, 1990.

[158] A. S. S. Reddy, P. Kasat, and A. Jain, "Box-office opening prediction of movies based on hype analysis through data mining," *International Journal of Computer Applications*, vol. 56, no. 1, pp. 1–5, 2012.

[159] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using mobile phones to determine transportation modes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, p. 13, 2010.

[160] A. Sauerländer-Biebl, E. Brockfeld, D. Suske, and E. Melde, "Evaluation of a transport mode detection using fuzzy rules," *Transportation research procedia*, vol. 25, pp. 591–602, 2017.

[161] 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.* Springer, 2011, pp. 431–446.

[162] R. E. Schapire and Y. Freund, "Boosting: Foundations and algorithms," *Kybernetes*, 2013.

[163] L. Schauer, M. Werner, and P. Marcus, "Estimating crowd densities and pedestrian flows using wi-fi and bluetooth," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services.* ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2014, pp. 171–177.

[164] R. C. Shah, C.-y. Wan, H. Lu, and L. Nachman, “Classifying the mode of transportation on mobile phones using gis information,” in *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*. ACM, 2014, pp. 225–229.

[165] Y.-S. Shih, “Families of splitting criteria for classification trees,” *Statistics and Computing*, vol. 9, no. 4, pp. 309–315, 1999.

[166] Y. Shimshoni, N. Efron, and Y. Matias, “On the predictability of search trends,” 2009.

[167] D. C. Shoup, “Cruising for parking,” *Transport Policy*, vol. 13, no. 6, pp. 479–486, 2006.

[168] A. Signorini, A. M. Segre, and P. M. Polgreen, “The use of twitter to track levels of disease activity and public concern in the us during the influenza a h1n1 pandemic,” *PloS one*, vol. 6, no. 5, p. e19467, 2011.

[169] S. Sikdar, S. Adali, M. Amin, T. Abdelzaher, K. Chan, J.-H. Cho, B. Kang, and J. O’Donovan, “Finding true and credible information on twitter,” in *17th International Conference on Information Fusion (FUSION)*. IEEE, 2014, pp. 1–8.

[170] S. Sinha, C. Dyer, K. Gimpel, and N. A. Smith, “Predicting the nfl using twitter,” *arXiv preprint arXiv:1310.6998*, 2013.

[171] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[172] B. Sommerdijk, E. Sanders, and A. van den Bosch, “Can tweets predict tv ratings?” in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, 2016, pp. 2965–2970.

[173] L. Stenneth, O. Wolfson, P. S. Yu, and B. Xu, “Transportation mode detection using mobile phones and gis information,” in *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in geographic information systems*. ACM, 2011, pp. 54–63.

[174] P. Stopher, C. FitzGerald, and J. Zhang, “Search for a global positioning system device to measure person travel,” *Transportation Research Part C: Emerging Technologies*, vol. 16, no. 3, pp. 350–369, 2008.

[175] B. Suh, L. Hong, P. Pirolli, and E. H. Chi, “Want to be retweeted? large scale analytics on factors impacting retweet in twitter network,” in *2010 IEEE Second International Conference on Social Computing*. IEEE, 2010, pp. 177–184.

[176] R. Suzuki and H. Shimodaira, “pvclust: Hierarchical clustering with p-values via multiscale bootstrap resampling, 2009,” *R package version*, pp. 2–0.

[177] D. Teodorović and P. Lučić, "Intelligent parking systems," *European Journal of Operational Research*, vol. 175, no. 3, pp. 1666–1681, 2006.

[178] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[179] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 411–423, 2001.

[180] T. Tiedemann, T. Vögele, M. M. Krell, J. H. Metzen, and F. Kirchner, "Concept of a data thread based parking space occupancy prediction in a berlin pilot region," in *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[181] C. Tiexin, T. Miaomiao, and M. Ze, "The model of parking demand forecast for the urban ccd," *Energy Procedia*, vol. 16, pp. 1393–1400, 2012.

[182] A. M. Townsend, *Smart cities: Big data, civic hackers, and the quest for a new utopia.* WW Norton & Company, 2013.

[183] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe, "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Fourth international AAAI conference on weblogs and social media*, 2010.

[184] I. Uysal and W. B. Croft, "User oriented tweet ranking: a filtering approach to microblogs," in *Proceedings of the 20th ACM international conference on Information and knowledge management.* ACM, 2011, pp. 2261–2264.

[185] M. J. Van Der Laan and S. Dudoit, "Unified cross-validation methodology for selection among estimators and a general cross-validated adaptive epsilon-net estimator: Finite sample oracle inequalities and examples," 2003.

[186] M. J. van der Laan, S. Dudoit, and A. W. van der Vaart, "The cross-validated adaptive epsilon-net estimator," *Statistics & Decisions*, vol. 24, no. 3, pp. 373–395, 2006.

[187] M. J. Van der Laan, E. C. Polley, and A. E. Hubbard, "Super learner," *Statistical applications in genetics and molecular biology*, vol. 6, no. 1, 2007.

[188] V. Vapnik, "The support vector method of function estimation," in *Nonlinear Modeling.* Springer, 1998, pp. 55–85.

[189] W. N. Venables and B. D. Ripley, *Modern applied statistics with S-PLUS.* Springer Science & Business Media, 2013.

[190] E. I. Vlahogianni, K. Kepaptsoglou, V. Tsetsos, and M. G. Karlaftis, "A real-time parking prediction system for smart cities," *Journal of Intelligent Transportation Systems*, vol. 20, no. 2, pp. 192–204, 2016.

[191] S. Wakamiya, R. Lee, and K. Sumiya, "Towards better tv viewing rates: exploiting crowd's media life logs over twitter for tv rating," in *Proceedings of the 5th International Conference on Ubiquitous Information Management and Communication.* ACM, 2011, p. 39.

[192] S. Wang, C. Chen, and J. Ma, "Accelerometer based transportation mode recognition on mobile phones," in *2010 Asia-Pacific Conference on Wearable Computing Systems.* IEEE, 2010, pp. 44–46.

[193] X. Wang, M. S. Gerber, and D. E. Brown, "Automatic crime prediction using events extracted from twitter posts," in *International conference on social computing, behavioral-cultural modeling, and prediction.* Springer, 2012, pp. 231–238.

[194] X. Wang, L. White, X. Chen, D. D. Gaikar, B. Marakarkandy, and C. Dasgupta, "Using twitter data to predict the performance of bollywood movies," *Industrial Management & Data Systems*, 2015.

[195] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013.

[196] G. Welch, G. Bishop *et al.*, "An introduction to the kalman filter," 1995.

[197] J. Xiao, Y. Lou, and J. Frisby, "How likely am i to find parking?–a practical model-based framework for predicting parking availability," *Transportation Research Part B: Methodological*, vol. 112, pp. 19–39, 2018.

[198] G. Yan, W. Yang, D. B. Rawat, and S. Olariu, "Smartparking: A secure and intelligent parking system," *IEEE Intelligent Transportation Systems Magazine*, vol. 3, no. 1, pp. 18–30, 2011.

[199] B. Yang, J. Dai, C. Guo, C. S. Jensen, and J. Hu, "Pace: a path-centric paradigm for stochastic path finding," *The VLDB Journal-The International Journal on Very Large Data Bases*, vol. 27, no. 2, pp. 153–178, 2018.

[200] B. Yang, C. Guo, C. S. Jensen, M. Kaul, and S. Shang, "Stochastic skyline route planning under time-varying uncertainty," in *2014 IEEE 30th International Conference on Data Engineering.* IEEE, 2014, pp. 136–147.

[201] F. Yang and Jensen, "ipark," in *16th Int. Conf. Extending Database Technol.* EDTB, 2013, pp. 705–708.

[202] J. Yang and S. Counts, "Predicting the speed, scale, and range of information diffusion in twitter," in *Fourth International AAAI Conference on Weblogs and Social Media*, 2010.

[203] Z. Yang, H. Liu, and X. Wang, "The research on the key technologies for improving efficiency of parking guidance system," in *Proceedings of the 2003*

IEEE International Conference on Intelligent Transportation Systems, vol. 2. IEEE, 2003, pp. 1177–1182.

[204] G. Yanyun, Z. Fang, C. Shaomeng, and L. Haiyong, "A convolutional neural networks based transportation mode identification algorithm," in *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2017, pp. 1–7.

[205] M.-C. Yu, T. Yu, S.-C. Wang, C.-J. Lin, and E. Y. Chang, "Big data small footprint: the design of a low-power classifier for detecting transportation modes," *Proceedings of the VLDB Endowment*, vol. 7, no. 13, pp. 1429–1440, 2014.

[206] T. Zaman, E. B. Fox, E. T. Bradlow *et al.*, "A bayesian approach for predicting the popularity of tweets," *The Annals of Applied Statistics*, vol. 8, no. 3, pp. 1583–1611, 2014.

[207] T. R. Zaman, R. Herbrich, J. Van Gael, and D. Stern, "Predicting information spreading in twitter," in *Workshop on computational social science and the wisdom of crowds, nips*, vol. 104, no. 45. Citeseer, 2010, pp. 17 599–601.

[208] G. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks:: The state of the art," *International journal of forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[209] Q. Zhang, Y. Gong, J. Wu, H. Huang, and X. Huang, "Retweet prediction with attention-based deep neural network," in *Proceedings of the 25th ACM international on conference on information and knowledge management*. ACM, 2016, pp. 75–84.

[210] Y. Zhao, H. Zhang, L. An, and Q. Liu, "Improving the approaches of traffic demand forecasting in the big data era," *Cities*, vol. 82, pp. 19–26, 2018.

[211] W.-X. Zhu and E.-X. Chi, "Analysis of generalized optimal current lattice model for traffic flow," *International Journal of Modern Physics C*, vol. 19, no. 05, pp. 727–739, 2008.

[212] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the royal statistical society: series B (statistical methodology)*, vol. 67, no. 2, pp. 301–320, 2005.