ScuDo
Scuola di Dottorato ~ Doctoral School
WHAT YOU ARE, TAKES YOU FAR

Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (30.th cycle)

Mining Heterogeneous Urban Data at Multiple Granularity Layers

Antonio Attanasio

* * * * * *

Supervisors
Prof. Silvia Anna Chiusano, Supervisor
Prof. Tania Cerquitelli, Co-supervisor

Doctoral Examination Committee:
Prof. Robert Wrembel, Referee, Poznan University of Technology
Prof. Rosa Meo, Referee, Università degli Studi di Torino
Prof. Enrico Masala, Politecnico di Torino
Prof. Elisa Quintarelli, Politecnico di Milano
Prof. Antonio Servetti, Politecnico di Torino

Politecnico di Torino
June 19, 2018

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

. . . .

Antonio Attanasio
Turin, June 19, 2018

Summary

The recent development of urban areas and of the new advanced services supported by digital technologies has generated big challenges for people and city administrators, like air pollution, high energy consumption, traffic congestion, management of public events. Moreover, understanding the perception of citizens about the provided services and other relevant topics can help devising targeted actions in the management. With the large diffusion of sensing technologies and user devices, the capability to generate data of public interest within the urban area has rapidly grown. For instance, different sensors networks deployed in the urban area allow collecting a variety of data useful to characterize several aspects of the urban environment.

The huge amount of data produced by different types of devices and applications brings a rich knowledge about the urban context. Mining big urban data can provide decision makers with knowledge useful to tackle the aforementioned challenges for a smart and sustainable administration of urban spaces.

However, the high volume and heterogeneity of data increase the complexity of the analysis. Moreover, different sources provide data with different spatial and temporal references. The extraction of significant information from such diverse kinds of data depends also on how they are integrated, hence alternative data representations and efficient processing technologies are required.

The PhD research activity presented in this thesis was aimed at tackling these issues. Indeed, the thesis deals with the analysis of big heterogeneous data in smart city scenarios, by means of new data mining techniques and algorithms, to study the nature of urban related processes. The problem is addressed focusing on both infrastructural and algorithmic layers. In the first layer, the thesis proposes the enhancement of the current leading techniques for the storage and elaboration of Big Data. The integration with novel computing platforms is also considered to support parallelization of tasks, tackling the issue of automatic scaling of resources. At algorithmic layer, the research activity aimed at innovating current data mining algorithms, by adapting them to novel Big Data architectures and to Cloud computing environments. Such algorithms have been applied to various classes of urban data, in order to discover hidden but important information to support the optimization of the related processes. This research activity focused on the development of a distributed framework to automatically aggregate heterogeneous data at multiple temporal and spatial granularities and to apply different data mining techniques. Parallel computations are performed according to the MapReduce paradigm and exploiting in-memory computing to reach near-linear computational scalability. By exploring manifold data resolutions in a relatively short time, several additional patterns of data can be discovered, allowing to further enrich the description of urban processes. Such framework is suitably applied to different use cases, where many types of data are used to provide insightful descriptive and predictive analyses.

In particular, the PhD activity addressed two main issues in the context of urban data mining: the evaluation of buildings energy efficiency from different energy-related data and the characterization of people's perception and interest about different topics from user-generated content on social networks. For each use case within the considered applications, a specific architectural solution was designed to obtain meaningful and actionable results and to optimize the computational performance and scalability of algorithms, which were extensively validated through experimental tests.

Acknowledgements

I would like to thank my tutors, Prof. Silvia Chiusano and Prof. Tania Cerquitelli, for their fundamental advices and for all the time they spent to help me during the PhD activity.

In addition, I would like to thank all the other researchers who worked with me during these years and in particular those of the Database and Data Mining Group of the Politecnico.

I want to express my gratitude to my colleagues at ISMB for letting me take this opportunity and for their support.

Finally, I want to thank my family and especially Marta, who always encouraged me and helped me to get over the hardest periods.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the last few years, the capability to both generate and collect data of public interest within urban areas has increased at an unprecedented rate, to such an extent that data rapidly scale towards *big urban data*.

A large variety of data can be collected in the urban context, ranging from data generated by citizens to those collected through sensors deployed in the city and monitoring environmental variables. *Air quality* measures, *weather* conditions, geo-referenced *people's activities*, contents from *social networks*, electric and thermal *energy consumption*, *traffic flows* and use of transport systems are just some examples of information that can be retrieved from a smart city.

The abundance and variety of data describing the urban context provide a remarkable opportunity to tackle interesting challenges and to add intelligence in several urban scenarios. Various types of analysis can be executed for many applications, like *environmental monitoring* to control pollution and reduce its effects over people; optimization of *buildings energy consumption*; detection of *similar interests* and activities among citizens; *road traffic management*; enhancement of *transportation systems*, etc. [1, 2].

The integrated analysis of all such types of data yields a more thorough outlook on the factors that characterize urban scenarios, useful to support a smarter administration of cities. When *huge amounts of heterogeneous data* are available, devising efficient *data mining* techniques that leverage on their highly informative power can effectively boost the evolution of urban areas into smart cities.

However, the huge *volume* and the *heterogeneity* of urban data collected from *manifold sources* and expressed with different *space and time granularities* increase the complexity of the analysis. To deal with such issues and to extract meaningful results, innovative *data management and processing techniques* should be devised.

1.1 Research topics description

This PhD activity mainly focuses on the characterization (i) of *buildings energy efficiency* from energy-related data and (ii) of *popular topics among citizens* from user-generated content on social networks. The proposed analyses are aimed at supporting the enhancement of urban services, like the distribution of heating energy for residential buildings, and at understanding the perception of citizens about such services and other related topics. The two application domains are introduced below, while the detailed research activities are described in Section 1.2.

Energy efficiency is a growing policy priority for many countries worldwide. According to the International Energy Agency (IEA), *buildings* represent roughly 40% of total final energy consumption in most countries. The amount of energy used for heating and cooling systems is about 60% in the residential sector [3], thus particular attention has been devoted to carry out innovative strategies for both monitoring and improving energy efficiency of *building heating systems.* To achieve this aim, many energy firms have begun exploiting Internet of Things (IoT) technologies to monitor the *Heating Distribution Networks* (HDN) in urban environments. Thanks to the pervasive proliferation of *sensors* and *smart meters*, the data generation capability of energy-related applications has rapidly increased, providing energy managers with tons of different fine-grained measures to be managed and analyzed. In addition, the integration of variables related with energy consumption (e.g., indoor and outdoor temperatures) makes possible to analyze a richer data collection and to obtain more significant results.

The analysis of energy-related data collections has received increasing attention from the research community. Indeed, they hold a great potential in terms of interesting knowledge that can be discovered to support the efficient management of heating systems. For instance, a critical challenge is the prediction of future buildings *energy/power demand* and of their daily peak values. An accurate estimation of these values makes possible the implementation of more efficient strategies to satisfy the aggregate energy demand.

Nevertheless, efficient *data integration* and *data analytics* methods should be devised to extract meaningful results from such *heterogeneous data* collected from *manifold sources* and expressed with different *space and time granularities.*

Social networks are often used by people to report information related to a variety of urban aspects. The analysis of such data can provide useful information to discover popular topics among people in a city. Understanding the collective dynamics of people's interests and needs can be a powerful advantage to devise effective targeted actions in the management of a smart city. Policy makers can exploit information from social networks to better understand people's opinions regarding highly debated topics such as transport networks, health-care systems, public safety [4], taxes, services, etc.

Location and time information associated with user-generated contents on social networks can enable a more complete characterization of frequent patterns of user interests across different cities and of their evolution over time. However, the ever *increasing volume* and the rather *heterogeneous dimensions* characterizing such data (*space*, *time* and *text content*) increase the complexity of the mining process. A further issue is represented by the *sparsity* that often characterize collections of data from social networks. Moreover, text content must be elaborated with appropriate algorithms to accurately quantify its *relevancy to a given topic*.

For both the aforementioned application domains, an additional challenge is represented by the computing *performance* and *scalability* of algorithms and underlying platforms, that often require *parallelized computations* and *distributed databases*. To properly deal with such challenges, advanced data management platforms and efficient processing algorithms are required.

1.2 Research activity overview

The overall PhD activity described in this thesis dealt with the *collection*, *aggregation* and *analysis* of different kinds of urban data, with a specific focus on the management of *heterogeneity* in *spatio-temporal data granularities*. Big Data challenges were addressed at both architectural and algorithmic layers. In the *architectural layer*, novel techniques for the storage and elaboration of big data, like *NoSQL distributed databases* and *in-memory cluster computing* platforms, were studied and employed. In the *algorithmic layer*, the thesis aimed at innovating current data mining algorithms, by operating with *parallel programming models* and *real-time stream processing* architectures, to fully exploit the underlying platforms. More specifically, the challenges addressed by the research activities are described below.

One of the main challenges of the research activity is to find efficient and effective integration strategies to extract relevant patterns for the objectives of the analysis. For this purpose, a distributed business intelligence framework, called *Multiple Spatio-Temporal Layers Explorator* (MuSTLE), was developed to support data mining algorithms by exploring heterogeneous data at *multiple layers of space and time aggregation*, using a scalable approach. The MuSTLE framework relies on the MapReduce paradigm to aggregate data from different sources and to build multiple layers of space-time granularity for the analysis, keeping in the database only data with the original granularity. The application of MuSTLE enables the exploration of multiple representations of data in a relatively short time (on-the-fly) and the extraction of patterns that can be detected only when data are expressed at given spatial and/or temporal resolutions. This approach increases the possibility to extract more significant patterns among data and to highlight phenomena that wouldn't be obtained with a unique data representation.

This PhD activity addressed different issues in the context of urban data mining, basically within two kinds of *applications*: the evaluation of buildings energy efficiency (*energy application*) and the estimation of main interests among people (*social application*). For each use case within the considered applications, a specific architectural solution, based on a particular instance of MuSTLE, was designed to optimize the computational performance and scalability of algorithms, which were extensively validated through experimental tests.

For the *energy application*, a full assessment of buildings energy efficiency was carried out analyzing both real consumption data (*operational rating*) and buildings features (*asset rating*). The MuSTLE framework was used for space-time characterization of energy-related variables at multiple granularities and for correlation and regression analysis.

Within the context of *operational rating*, two different platforms have been designed and implemented, based on a common architecture and extended according to the characteristics of the analysis: DA-BOR for descriptive analytics and PA-BOR for predictive analytics. *Descriptive analytics* algorithms were used to compute different classes of Key Performance Indicators (KPIs) for building energy efficiency, also taking into account the relationship with other variables like external temperature. In this activity a *NoSQL distributed database* and a *parallel programming model* were used to speed-up the computation of KPIs from huge amounts of energy-related data.

On the other hand, *predictive analytics* algorithms were devised to forecast instantaneous *power demand* values of heating systems at fine-grained time granularity. A correct estimation of the power (and energy) demand of buildings heating systems is useful to devise strategies for improving the overall energy utilization. However, the estimation of the power required during a short time interval is a complicated task, as it is affected by several elements difficult to be modeled. Moreover, when estimations are based on data collected in real-time, a specific issue is represented by the sizing of computational resources needed to provide results for thousands of buildings in time. Therefore, in this activity *in-memory cluster computing* platforms and *real-time stream processing* algorithms were used to estimate future heating power demand in near-real-time and with a small prediction error.

Within the context of *asset rating*, a suitable data mining approach, called HEDEBAR, was devised to model the relationship between building features and heating energy demand, using data from Energy Performance Certificates (EPCs). Energy performance certification of buildings is considered as a cornerstone for improving energy efficiency, however, the proliferation of several certification methods does not facilitate a uniform evaluation of buildings located in different areas. Moreover, such methods are often based on a plethora of parameters and are hardly interpretable by experts. Therefore, the challenge of this activity was the definition of an asset rating methodology to generate accurate building energy demand models based on few relevant features and that are easily interpretable by domain experts. In this activity various data mining algorithms were suitably employed in all the steps of the methodology.

For the *social application*, the PhD activity focused on the analysis of data from social networks to provide useful information about the relationship, in several respects, between citizens and various popular topics relevant for the city. The proposed analysis has a specific focus on the *text and space-time characterization* of user posts on Twitter (tweets), to highlight the common interests among people from different cities.

Specifically, a data analytics methodology, called TCharM, based on *clustering analysis* and *association rules*, has been developed for the exploration of large collections of Twitter data along three dimensions, i.e., text content, posting time and place, to support context-aware topic trend analysis. The main obstacle to the extraction of significant patterns from large collections of tweets is represented by the inherent sparseness of tweets and the consequent low cohesion of clusters extracted using distance measures proposed by existing related works. Therefore, a new distance measure has been also defined and extensively validated through an analytical comparison with other measures. The methodology provides results describing the most discussed topics across space and over time, thus enabling to highlight the main differences of users' interests among multiple cities and their temporal evolution. In this activity clustering and association rules mining algorithms were implemented and executed over an *in-memory cluster computing* platform.

This thesis is organised as follows. The MuSTLE architecture is described in Chapter 2. Chapter 3 presents the research activities focused on the evaluation of thermal energy efficiency of buildings through operational rating, while Chapter 4 investigates the relationships between buildings features and thermal energy demand with asset rating. Chapter 5 presents the research activities focused on the characterization of people's interests from social networks. Chapter 6 summarizes the achieved results and discusses future developments for the proposed approaches.

Chapter 2

The MuSTLE framework for big urban data mining

Interest in *urban data mining* has rapidly grown during the last few years, both in the industrial and research domains, as well as in the Public Administration [1]. The joint analysis of data coming from different sources enables the discovery of meaningful relationships among various aspects of the urban environment and thus could increase the awareness of policy makers for city planning. For instance, discovering the correlation between traffic flow in a given area of the city and high pollution in the same area can help administrators in defining more targeted and effective environmental policies. Challenging issues come from the application of innovative *data management* and *data mining* techniques to new and more complex fields, as well as from the design of innovative systems able to continuously monitor and analyze a smart city environment.

Urban data mining is often characterized by high *data volume* and *data heterogeneity*, which increase the complexity of the analysis. Therefore, alternative efficient *data storage* and *data processing* techniques are required. Also *data integration* should be smart enough to produce suitable data sets from data generated by several sources that make use of different *space and time references.* To discover useful results, it is important to express features with appropriate *space and time granularities.* According to the type of targeted analysis, the exploration of data at multiple space-time granularities can bring out interesting knowledge at different levels.

The research activity described in this thesis, focused on the analysis of big heterogeneous urban data, led to the design and development of a *distributed business intelligence engine*, called *Multiple Spatio-Temporal Layers Explorator* (MuSTLE), that efficiently supports the *integration* and *analysis* of huge and heterogeneous data collections generated in the smart city context.

With the aim of supporting different data analyses with various spatial and temporal abstraction levels, MuSTLE stores fine-grained data collected in the urban area. To efficiently deal with big heterogeneous data sets, in MuSTLE data are stored in a *distributed NoSQL repository* based on MongoDB [5, 6].

Then, MuSTLE computes *space-time aggregation* to transpose the original data into the proper resolution for the analyses.

To gain useful insights from the stored collections, e.g., to predict future values of some parameters, MuSTLE runs *correlation and regression analysis* among different urban data, for multiple space-time abstraction level.

MuSTLE exploits the *MapReduce paradigm* [7] to quickly perform data aggregation and data analysis *on-the-fly*, i.e., it stores data only at the original space-time granularities and aggregates them once for each target granularity. The MapReduce operations make possible to distribute computation load over multiple nodes, reducing execution time and scaling up to bigger data collections.

In this chapter the overall architecture of MuSTLE is described. A representative use case is also presented to demonstrate the use of MuSTLE with different kinds of urban data. The work presented in this chapter has been published in [8].

This chapter is organized as follows. The overall context for the analysis of urban data is described in Section 2.1. Section 2.2 presents the related research work on urban data mining. Section 2.3 introduces the components of the MuSTLE framework. Section 2.4 illustrates some demonstrative examples of urban data analysis with the MuSTLE framework. Section 2.5 demonstrates the scalability of data aggregation with MuSTLE. Section 2.6 discusses in depth the experimental results.

2.1 Context for heterogeneous urban data mining

In a smart city context, many data sources are usually employed to monitor different urban processes. Monitoring devices may be deployed in different city areas and they may use a different timeline in sampling values.

To take into account the various facets of the urban environment, the MuSTLE system collects and analyses measures of different data types as *air pollutant concentrations*, *weather conditions*, *vehicle traffic*, and *building energy consumption* data. More specifically, the following categories of data are currently collected and analysed in MuSTLE.

Meteorological measures. Weather conditions are monitored by collecting the most common meteorological indicators (as air temperature, relative humidity, cumulated precipitations and precipitation rate, wind speed, atmospheric pressure) from weather stations distributed throughout the city. Data are collected with a sampling period of few minutes (usually 5 minutes), but different and variable resolutions can be used by some stations.

Pollutant concentration. Concentration measures for each air pollutant are periodically collected through dedicated sensors deployed in monitoring stations. Various pollutants are monitored, including particulate matters (PM_{10} and $PM_{2.5}$), benzene (C_6H_6), nitric oxide (NO), nitrogen dioxide (NO_2), sulfur dioxide (SO_2), carbon monoxide (CO).

Each station monitors the concentrations of various pollutants at a fixed time granularity. Data are usually collected daily or hourly, according to the specific pollutant.

Urban facilities as the *energy consumption and power level* measures registered by the heating systems of residential buildings. These buildings represent the nodes of a monitoring network. The volume of each building is also used to normalize energy and power values, to make comparisons in terms of consumption per volume unit. Data are collected with a variable sampling period (with a mean value of about 5 minutes).

Citizen mobility as vehicle traffic data. Road traffic flow is measured by roadside traffic recording stations that count the number of vehicle transits per minute. Data are usually collected every minute.

The research study presented in this chapter aims at discovering interesting relationships among different factors characterizing the urban environment, inspecting multiple layers of space and time granularity. Two different kinds of analysis are proposed: *correlation analysis*, through the computation of a correlation coefficient that quantifies the degree of connection of couples of variables; *regression analysis*, that tries to model the relationship between variables according to a (linear) equation. A proper regression equation allows also to estimate the unknown values of a variable through the measured values of the other (e.g., estimate the concentration of a pollutant when the energy consumption of buildings is known).

Data analyzed in this chapter are referred to the city of Torino (Italy), administratively organized in 10 districts, each one including one or more quarters. As a reference case study for data analysis, a *Space Frame* (SF) corresponding to a quite large district (about 7 km^2) located in the central part of Torino was considered. This district includes one station for air pollution monitoring, one traffic recording station, 5 weather stations, and around 100 monitored heating systems of residential buildings. As a *Time Frame* (TF) for data analysis, a 7-month time period from October 2014 to April 2015 was considered.

Pollutant concentration data were gathered by the ARPA Piemonte [9] through monitoring stations equipped with a set of sensors, each one measuring a different pollutant, and provided by the Sistema Piemonte open data portal [10]. Meteorological measures were collected through the Weather Underground web service [11], which gathers data from a geo-referenced network of Personal Weather Stations registered by users. *Vehicle traffic* and *buildings energy consumption* were provided by the Smartdatanet open data platform [12] managed by Regione Piemonte. Traffic data register the number of vehicles per minute that pass by the sensors placed in some fixed points of the city. Buildings energy data register the energy consumptions for space heating of residential buildings.

2.2 Related work

Analysis of data related to the urban context is often aimed at evaluating performance indicators [13], discovering relevant relationships [14, 15], detecting particular events [16, 17] or building predictive models [18]. Also data associated to citizens are often extracted from phone networks and social media to take into account the human perception of the urban environment [19].

Some works like [20, 21, 22], are based on the active involvement of people in data collection through feedbacks from mobile devices. Since multiple kinds of data from heterogeneous sources are often considered for such analyses, the integration issue has already emerged in literature. As an example, authors in [15] propose a platform for the integration of data gathered from independent organizations, to deliver an integrated research environment for analysis of urban data. Authors in [14] focus on the problem of data aggregation over spatial and temporal dimensions, before evaluating the correlations between a single dynamic record set of mobile phone calls and other contextual and static datasets like urban demographics and points-of-interest (POIs). Two spatial resolutions are considered: the smaller squared cells used in mobile phone networks and the larger administrative districts. Pearson's and Spearman's coefficients are used for correlation analysis and multiple linear regression for prediction.

In [16], integration is achieved through the use of data fusion techniques like *CumSum* algorithm [23] for outlier detection and the *linear opinion pool* method [24] for deriving the final value. [25] presents another space-time model for the integration of vector data based on ontology classification and geocoding techniques. In [26] the fusion of data from heterogeneous sources relies on existing open source ETL tools, like Pentaho, CloverETL, Talend. Other web and cloud based services for heterogeneous urban data integration and analysis are proposed in [27], where semantic enrichment is included and in [17] as a monitoring tool for energy management of the whole city infrastructure. In [16] fusion techniques, regression and fuzzy logic are employed to derive a decision making tool for the identification of city environmental events.

Other works aim at characterizing the impact of different spatial [28], [29] and temporal [30] aggregation techniques over the analysis.

On the technological side, other works have exploited the flexibility of NoSQL databases for the storage of heterogeneous urban data. For instance, in [31] a NoSQL database schema is defined to enhance scalability and facilitate searches for the analysis of some urban metrics. However, the final results of the analyses are stored in a relational database. The adoption of MapReduce paradigm to deal with data integration is experimented in [32], where it is used for heterogeneous query execution on large datasets, based on the concept of Virtual Database (VDB), a container for components used to integrate data from multiple data sources, so that they can be accessed through a uniform API. [33] addresses the problem of large scale data integration with a service oriented data integration architecture based on Hadoop and MapReduce, to exploit distributed processing and data replication. [34] presents a tool for heterogeneous data integration

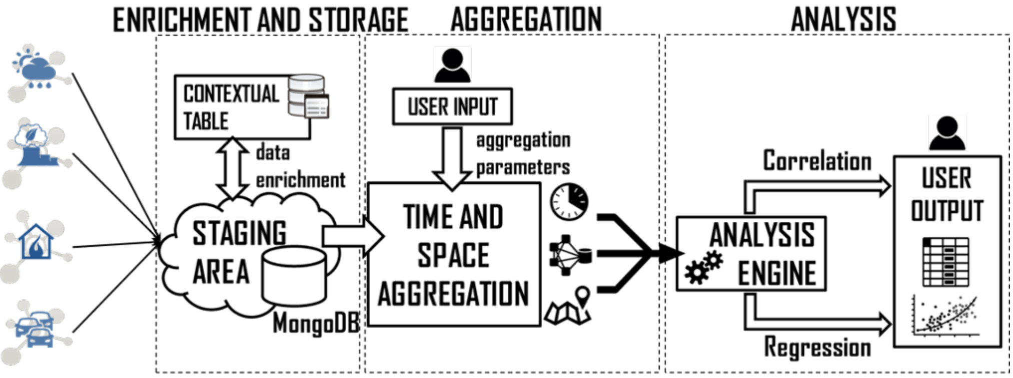


Figure 2.1: Data analysis steps of the proposed MuSTLE framework

based on Hadoop, that integrates also visual analysis. In this case, MapReduce is used just to compare individual records in order to reduce redundancy.

Contribution of the research activity. This research activity proposes the MuSTLE framework that executes data aggregation on-the-fly at different spatial and temporal resolutions, using the MapReduce computational paradigm, to facilitate the discovery of relevant relationships among data. None of the cited works has proposed a similar solution for the integration of different kinds of heterogeneous data.

2.3 MuSTLE architecture

Figure 2.1 represents the data analysis steps of MuSTLE framework, which is fed with data collected from devices distributed all over the city. Data elaboration is executed in three stages: *data ingestion and storage*, *temporal and spatial data aggregation* and *data analysis*.

In the first stage, collected data are stored in a *document-oriented distributed database*. Since data analysis can be based on different space and time resolutions, the intermediate stage performs the *temporal and spatial aggregation* for each target value of *Space Granularity* (SG) and *Time Granularity* (TG). Therefore, data of different types are aggregated into documents, which represent the input for the *data analysis* stage. The MuSTLE components are detailed in the following subsections.

2.3.1 Data ingestion and storage

Raw data received from various sources include *contextual metadata* to characterize the spatial and the temporal contexts in which measures have been acquired. More specifically, *spatial metadata* describe the geographical position of each data source in the urban area, while *temporal metadata* describe the time when data values were measured or generated by the data source.

Data are then formatted as JSON documents and stored in a distributed repository based on *MongoDB*. This choice is motivated by multiple factors. (i) First of all, the *horizontal scalability* is enhanced by the *sharded cluster* architecture, that stores data in a distributed fashion. As the size of the data increases, adding more servers allows to scale and satisfy the demand of a higher number of read and write operations. (ii) The overall *computational time* is also reduced by processing data in parallel, through the built-in MapReduce engine, used in MuSTLE for both data aggregation and data analysis. (iii) Moreover, the sharded cluster architecture provides *high redundancy and availability*, since data are replicated on different shards, thus reducing the risk of data loss from a single server failure. (iv) Finally, the document-oriented data model of MongoDB eases the storage of *heterogeneous data* as well as the integration of new data types, as it doesn't require to define a schema before storing data. Even if special purpose databases can provide advanced features for the management of time series, MongoDB has been preferred due to other important features that make it well suited for the requirements of the analysis. First, MongoDB has an integrated *MapReduce* engine that can be directly executed on the storage nodes. Other databases do not directly support MapReduce operations, but they need a connector to other platforms like Apache Hadoop. Other popular databases, specifically designed for time series management, do not support MapReduce operations at all. Second, MongoDB provides a configurable sharding strategy, that allows the user to optimally distribute data across storage nodes and to improve the performance of queries. This feature is available in few other databases. Finally, even if time series databases provide an easy management of time series synchronization, MuSTLE entails also the aggregation of data along the space dimension, which needs to be addressed separately.

Horizontal scalability is obtained by exploiting *data sharding*, i.e., by dividing the collection and storing its data documents across distributed servers (*shards*). Data are distributed across shards using the hash value of the *document ID* as *sharding key*. This random policy provides uniform data distribution independently of temporal and spatial attributes and also computational effort for aggregation is evenly distributed across all shards, whatever the target space-time granularity.

Replication is obtained by exploiting *replica sets* of MongoDB. Each replica set consists of a primary node and a secondary node. All write requests go to the primary node, while the secondary node can be exploited to increase the read capacity, even if with possible data inconsistencies which are easily tolerated at the application layer.

2.3.2 Distributed space-time data aggregation

In the previous stage, collected data are stored in the data repository with their original spatial and temporal resolutions. To analyse different types of data, they must be expressed according to a unique combination of space and time resolutions.

The data aggregation process is driven by the following two parameters: *Spatial Granularity* (SG) and *Temporal Granularity* (TG). They represent the common values of *resolution*, respectively in space and time, used to express the input data of the analysis. The concept of *granularity* identifies elementary units of time and space for data representation. For each pair of time and space unit, every feature takes a single value in the input data set. The elementary unit can have either fixed or variable dimension. In the former case, SG can have, e.g., a length of $1km$, a square surface of $1m^2$, etc., while TG can last, e.g., 1 hour, 1 day, 1 week, etc. In the latter case, SG can take the dimension of, e.g., a building (in a city of buildings with variable dimensions), a quarter, a district, etc., while TG can last, e.g., 1 month (28, 29, 30 or 31 days), 1 year (365 or 366 days), or the duration of a given process. The target values of SG and TG, used for the analysis, should be equal to or higher than all the granularities of the original data.

The exploration of multiple granularities in both dimensions increases the possibility to extract more significant patterns among data and to highlight phenomena that wouldn't be obtained with a single pair of SG and TG. For this reason, MuSTLE explores data at multiple space-time granularities by means of *on-the-fly aggregations*. For each data collection, the data aggregation task is organized as a two-phase process.

The first phase computes the *temporal data aggregation* according to the target TG value. For each data source, measures collected during the same time slot are combined together to obtain a single *temporal aggregated value* for the time slot. The aggregation operator used to combine measures may vary based on the considered feature and on the scope of the analysis. For instance, mean and median values can be suitable operators for this phase.

The second phase computes the *spatial data aggregation*, i.e., data aggregation over subsets of sources grouped according to the SG value. For each area defined by SG, temporal aggregated values referred to the same time slot are combined in turn to obtain a single *space-time aggregated value*. The aggregation operator used to combine measures may vary based on the considered feature and on the scope of the analysis. For instance, when the number of measures that contribute to the *temporal aggregated values* varies from one source to another, the *weighted mean value* can be a suitable operator for this phase. In this case, the weight associated to a source depends on the number of measures that contribute to the temporal aggregation for a given slot.

The *aggregation process* relies on operations based on the MapReduce paradigm, which is well suited to compute summations and also (weighted) mean values. For each processed document (containing a measure), (i) the *map* function emits its *value* and the *key* that identifies its time window or space area and (ii) the *reduce* function both counts and sums the values of all the measures emitted for the same key. For spatial aggregation, the values are first multiplied by their weights and the sums of weights are computed too. Finally, (iii) a *finalize* function computes the aggregated values dividing each sum by the respective count (or sum of weights).

2.3.3 Distributed data analysis

Data aggregated at the desired time and space granularities are ready to be eventually processed, using the MapReduce engine integrated in MuSTLE. As an example, (i) *correlation analysis* and (ii) *regression analysis* are presented in this section.

Correlation analysis in MapReduce

Let $X = \{x_i, i = 1, ..., n\}$ and $Y = \{y_i, i = 1, ..., n\}$ be two time series monitored in a same space area in the city, where i value determines the time frame (e.g., a pollutant concentration and the temperature value at different time instants). MuSTLE analyses the pairwise correlation between time series X and Y through the *Pearson Product Moment Correlation* (PPMC) coefficient $\rho_{X,Y}$ [35], which is a measure of linear dependence not influenced by the unit of measure of X and Y. The higher the $\rho_{X,Y}$ value, the stronger the correlation (either negative or positive, according to the sign of the coefficient). In MuSTLE, the $\rho_{X,Y}$ coefficient between time series X and Y has been computed on MapReduce using the following equation [36]

$$\rho_{X,Y} = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}}. \tag{2.1}$$

Based on the MapReduce paradigm, the *map* function emits key-value pairs, where *key* identifies the dataset for the current analysis and *value* is the set of linear and quadratic terms that appear in Equation 2.1. The *reduce* function collects and condenses the values with the same key to calculate the summations used in Equation 2.1, that is eventually computed by the *finalize* function.

Regression analysis in MapReduce

Regression analysis allows estimating the unknown values of a dependent variable (Y) through the measured values of an explanatory one (X).

Simple linear regression analysis [37] is provided in MuSTLE to model the relationships between variables Y and X based on equation $Y = aX + b$. Parameters a and b are respectively the slope and the intercept of the linear function; they are computed in MapReduce using the following equations [38]

$$a = \frac{(\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} x_i y_i)}{n(\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)^2}, \tag{2.2}$$

$$b = \frac{n(\sum_{i=1}^{n} x_i y_i) - (\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i)}{n(\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)^2}. \tag{2.3}$$

Since the linear and quadratic terms are the same of Equation 2.1, regression analysis employs the same *map* and *reduce* functions of correlation analysis.

2.4 Examples of data analysis

This section reports the results of correlation analysis and regression analysis performed on a collection of urban data, described in Section 2.1, using MuSTLE.

Experiments address the analysis of the correlation between different factors characterizing the urban environment, i.e., meteorological data, concentration of air pollutants, buildings energy consumption and traffic flows. For instance, air pollution is usually assumed to be affected, to different extents, by all the other factors, while energy consumption of buildings can be influenced by climate condition. The proposed analysis are aimed at assessing such hypotheses and, possibly at discovering new unexpected relationships.

Both data aggregation and analysis have been implemented as described in Section 2.3.2 and Section 2.3.3 respectively.

Correlation analysis

To analyse the pairwise correlation between urban variables, the PPMC coefficient $\rho_{X,Y}$ has been computed in MapReduce (see Section 2.3.3) at different time granularity (TG) values. The most relevant results, i.e., those with overall highest values of $\rho_{X,Y}$, were obtained with *TG = 1 day* and are reported in Table 2.1 and Table 2.2.

Although there are no hard rules for describing the correlation strength based on the $\rho_{X,Y}$ value, after an extensive survey, the following rule-of-thumb has been used to evaluate the results: the $\rho_{X,Y}$ value can reveal a *weak* ($0 < |\rho_{X,Y}| \leq 0.3$), *moderate* ($0.3 < |\rho_{X,Y}| \leq 0.7$), or *strong* ($0.7 < |\rho_{X,Y}| \leq 1$) correlation between variables X and Y.

Table 2.1 shows the value of $\rho_{X,Y}$ between *building energy consumption* and the concentration of various *air pollutants*. Results point out a *moderate correlation* with Carbon Monoxide (CO) ($\rho_{X,Y} = 0.63$), Benzene (C_6H_6) (0.56), Nitric Oxide (NO) (0.55), Nitrogen Dioxide (NO_2) (0.49); and a *weak correlation* with Particulate Matters PM_{10} (0.26).

The scatter plot in Figure 2.2 further details the correlation between values of building energy consumption and CO pollutant concentration for different days.

Table 2.2 reports the correlation between various *weather variables* and both *air pollutants concentration* and *building energy consumption*. Both temperature and wind

Table 2.1: PPMC coefficient $\rho_{X,Y}$ between building energy consumption and pollutants concentration

	CO	**NO$_2$**	**NO**	**C$_6$H$_6$**	**PM$_{10}$**
Energy cons.	0.63	0.49	0.55	0.56	0.26

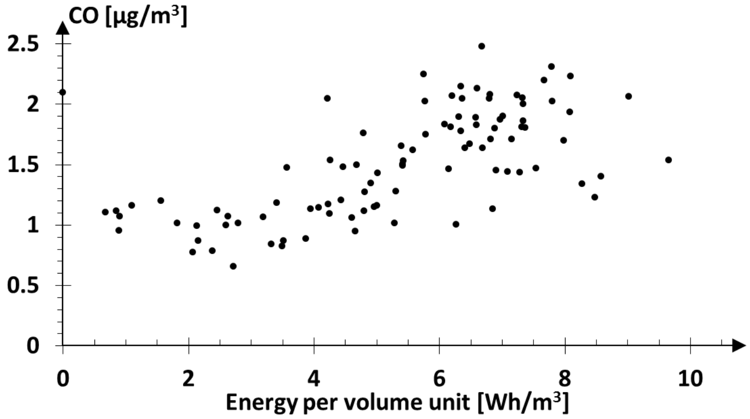


Figure 2.2: Daily average buildings energy consumption per volume unit vs average CO concentration

speed have a *moderate (negative) correlation* influence over CO (-0.56 for both), NO$_2$ (-0.4 and -0.53 respectively), C$_6$H$_6$ (-0.47 and -0.49) and NO (-0.42 and -0.41). Pollutants have a *low (negative) correlation* with precipitation rate (values in the range from -0.31 to -0.18). The negative correlation values indicate an inverse dependency between the two variables. For instance, the CO concentration is higher for lower values of temperature and wind speed.

As an example, the scatter plot in Figure 2.3 details the correspondence between wind speed and CO values. Humidity shows a weak correlation with all pollutants.

Concerning the *building energy consumption*, it shows a *strong correlation* with temperature (-0.82) and a *moderate correlation* with humidity (0.42) and wind speed (-0.59). Thus, the energy consumption is higher for lower values of temperature and wind speed and for higher values of humidity.

The scatter plot in Figure 2.4 shows the distribution of temperature and building energy consumption values.

Regression analysis

Simple linear regression analysis has been applied to couples of variables with high and moderate correlation. In order to assess the predictive power of the model, tests have been performed on the available data using a 10-fold cross validation. The *Mean Absolute Percentage Error* (*MAPE*) and the *Symmetric Mean Absolute Percentage Error*

Table 2.2: Pearson's correlation coefficient between weather parameters and building energy consumption and pollutants

	Temperature	**Humidity**	**Precipit.**	**Wind speed**
Energy cons.	-0.82	0.42	0.22	-0.59
CO	-0.56	0.14	-0.18	-0.56
NO_2	-0.40	-0.04	-0.23	-0.53
NO	-0.42	0.04	-0.22	-0.41
C_6H_6	-0.47	0.19	-0.23	-0.49
PM_{10}	-0.22	0.11	-0.31	-0.36

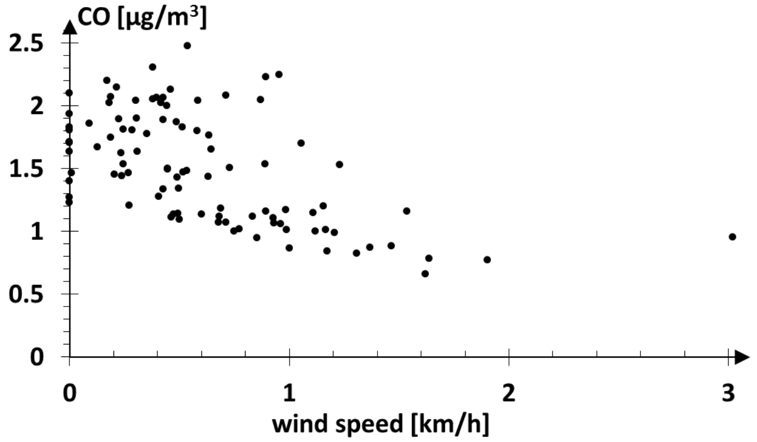


Figure 2.3: Daily average wind speed vs average CO concentration

(*SMAPE*) metrics are used to measure the predictive performance of the model:

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{A_i - P_i}{A_i}\right| \tag{2.4}$$

$$SMAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|A_i - P_i|}{|A_i| + |P_i|} \tag{2.5}$$

In both equations, A_i is the actual value of the target variable for sample $s^{(i)}$, while P_i is the corresponding predicted value.

MAPE is not a well suited metric when many actual power levels close to zero are also included. In this case, MAPE may significantly increase as it poses no upper bound to the error rate of overestimated predictions (while MAPE → 100% when $P_i \rightarrow 0 \wedge A_i \geq$

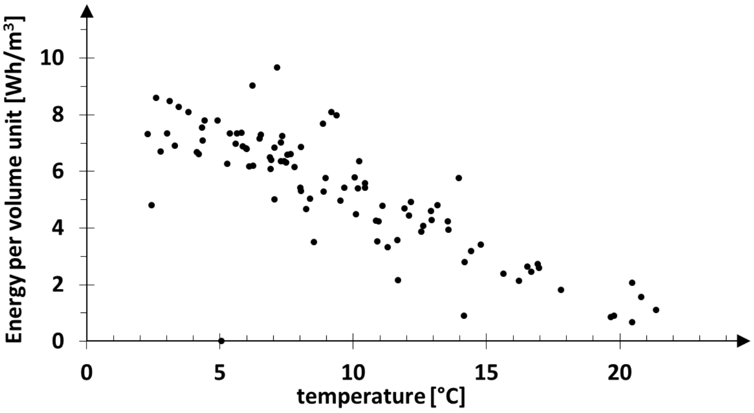


Figure 2.4: Daily average temperature vs average building energy consumption per volume unit

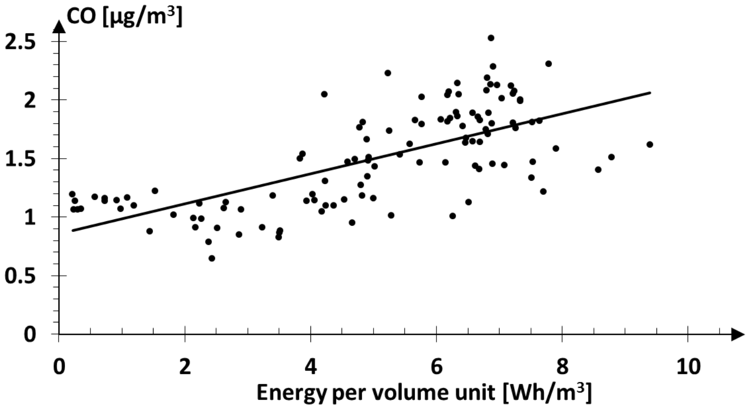


Figure 2.5: Linear regression plot between daily building energy consumption per volume unit and CO concentration

0). On the other hand, SMAPE is always in the range [0%, 100%], thus limiting the error rate on the predictions of lower values and reducing their influence on the overall error. The only drawback of SMAPE is that it is not symmetric between overestimated and underestimated forecasts of the same actual values. Specifically, for a same value of absolute prediction error, the underestimated forecast has a greater impact on the overall SMAPE value.

As an example, Figure 2.5 shows the regression equation two moderately correlated parameters: building energy consumption against CO pollutant concentration. The resulting model is quite accurate in predicting the CO concentration knowing the building energy consumption, since it exhibits a MAPE equal to 18.8%.

2.5 Computing performance of algorithm

The algorithm for time and space data aggregation has been tested on a MongoDB cluster of 7 nodes, measuring the variation of the execution time with respect to the number of employed nodes (from 1 to 7). The plot in Figure 2.6 refers to daily aggregation (TG = 1 day) of about 68 thousands records related to a 7-months period. The execution time goes down but with a decreasing rate. The highest *efficiency* (speedup divided by the number of nodes) is reached with 2 nodes (0.83) and significantly decreases from 5 (0.82) to 6 nodes (0.76). Therefore, 5 nodes can be considered a good balance between the speed-up and the need to minimize the employed nodes.

Despite the small size of the data set, the plot allows to evaluate the speedup and the efficiency for different sizes of the cluster. A more challenging use case, where the use of parallel processing is adequate, is the one of Section 3.4.4, where the same algorithm is applied to data about 2000 buildings (more than 50 million records).

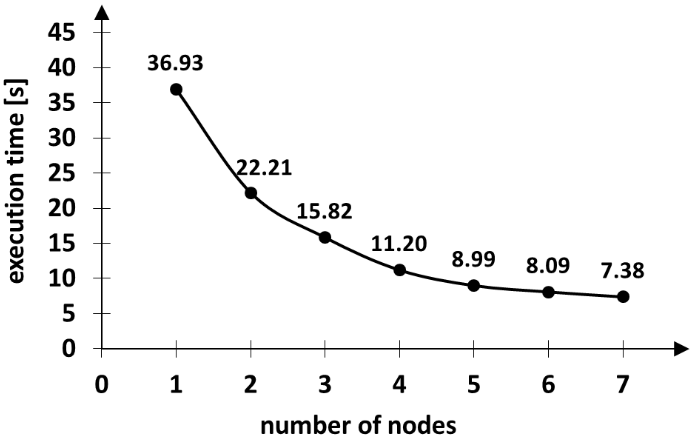


Figure 2.6: Execution time of the MapReduce aggregation algorithm on about 68k records using up to 7 computing nodes

2.6 Discussion

The experimental Section 2.4 has introduced some examples of information provided by MuSTLE, through the exploration of different kinds of data at multiple space and time layers. MuSTLE does it by processing data in a distributed way, ensuring a good scalability of algorithms, as demonstrated in Section 2.5.

Some significant information about the relationship among variables of different type was discovered. For instance, we discovered a strong correlation between air temperature and buildings energy consumption and a moderate negative correlation between wind speed and pollutants concentration. The PPMC coefficient has been used to quantify the correlation between variables of different type. The use of distance measures for time series, like *Dynamic Time Warping* (DTW) [39], has not been taken into account as they are not really suitable for the considered experiments, where the distance between time series related to the same variable is not considered. Moreover, DTW expresses a measure of distance between time series. The correlation analysis proposed in this study is aimed at discovering variables that are correlated, but not necessarily because they have similar values (either at the same time or at shifted times).

It is important to note that these high *statistical correlations* do not necessarily imply a physical dependence among the variables (e.g., cause and effect). Nevertheless, no apparent statistical correlation was found into the analysed data between the observed traffic flow and the other variables. These results may be due to the fact that the selected district is located in the central city area. Thus, the average daily traffic shows a low variability, loosely dependent from the other factors. With a view to improving the results of this study, Bayesian networks [40] could be a good tool to infer and quantify the dependencies among the analysed variables and will be taken into account for a future extension of the MuSTLE engine.

In next Chapters 3 and 5, MuSTLE has been enriched with advanced data mining algorithms, for different kinds of analysis: (i) by assessing the data abstraction level that highlights the most interesting correlations among available data, in particular for energy-related data and (ii) to discover the most relevant topics discussed by users in different urban areas, for user-generated data on social networks. Additional urban data types can be collected and integrated in MuSTLE.

Chapter 3

Evaluation of buildings energy efficiency using real energy-related data

Energy efficiency is a growing policy priority in many countries. A notable portion of total final energy consumption is ascribed to the heating and cooling systems of residential buildings. Thus, innovative strategies for improving energy efficiency are needed. The recent availability of different energy-related data collected through IoT devices makes possible a more in depth assessment of buildings energy consumption. Indeed, interesting knowledge can be discovered from those data to support a more targeted management of heating systems, reducing inefficiencies and energy waste.

In this chapter, the characterization of the urban context is focused on the analysis of real data related to buildings energy consumption for space heating. These values have been measured by devices placed inside and around buildings. The research activity presented in this chapter has been published in [13, 41].

The overall purpose of this study, in the context of *urban data mining*, is to provide data analytics services useful to improve the energy efficiency of buildings. The *energy efficiency* of a building is evaluated by comparing its energy consumption per unit of volume (or floor area) with the average values of energy consumption of buildings of the same type, under similar environmental conditions.

The research activity addressed two different kinds of analysis to thoroughly investigate the *energy consumption* of residential buildings for space heating, using real data (*operational rating*). The two analyses have different purposes: *descriptive* and *predictive*. In Section 3.4 descriptive analytics algorithms are used to compute different classes of Key Performance Indicators (KPIs) about thermal *energy efficiency* of buildings. Section 3.5 is still focused on operational rating but with a predictive purpose, as the aim is to forecast instantaneous *power demand* values of heating systems at fine-grained time granularity.

Data employed in the research activity presented in this chapter represent measures of urban parameters, collected through *geo-referenced devices* and expressed as *time series*. In particular, we combined *meteorological data* with *energy-related measures*, due to the well-assessed strong correlation between climate conditions and thermal energy consumption in buildings. Such data are analysed at different space and time granularities, adopting the aggregation strategy of the MuSTLE framework described in Section 2.3.2.

This chapter is organized as follows. The description of the urban context considered for the analysis of buildings energy data is provided in Section 3.1. Section 3.2 presents the research work related to the management and analysis of real building energy consumption data. Section 3.3 describes all the layers of the general architecture for operational rating. Section 3.4 and Section 3.5 present and comment the experimental results, respectively for descriptive and predictive analytics. Section 3.6 discusses in depth the achieved results.

3.1 Urban context for the analysis of buildings energy data

The PhD research activity presented in this chapter focuses on the analysis of heating energy data collected from *urban buildings*. The reference city context is organised in various districts. A *district* includes several buildings which are provided with energy coming from a *District Heating System* (DHS) and transferred through a *Heating Distribution Network* (HDN). Each building has one (or more) *heat exchanger* connected to the HDN to receive energy for space heating. *Smart meters* and *sensors* placed inside buildings monitor the status of the heat-exchangers by measuring several variables, like energy and power levels and heat temperatures. Other *sensor networks* are deployed throughout the city and monitor the environmental conditions surrounding the buildings, like temperature and humidity.

For the analyses described in this chapter, two types of data have been used to investigate the energy efficiency of buildings: *dynamic data*, measured roughly every few minutes and potentially exhibiting huge volume and highly variable values; and *static data*, describing some time invariant properties of the data source.

Dynamic data include energy-related data, continuously acquired from each building, and other variables that characterize the environmental conditions, both inside and outside buildings. Smart meters installed in buildings provide fine-grained data related to building thermal energy, like instantaneous *power demand*, cumulative *thermal energy consumption*, *water flow* and corresponding *temperature* inside the heat exchanger. Sensors placed inside and outside buildings collect measure about *climate conditions* for different points of the city and at different time instants. Meteorological web services (e.g., Weather Underground [42]) can be queried to access such data, which include different kinds of meteorological variables as temperature, relative humidity, precipitation, wind direction, UV index, solar radiation and atmospheric pressure.

Static data report features characterizing the data source as its *geographical location* (longitude and latitude). Static data also include information characterizing buildings as the *volume* and *floor area* of each building where smart meters are located. These values are used to normalize energy and power values to compare different buildings in terms of consumption per volume/surface unit. Also the indication of building type (residential, schools, etc.) is useful to compare buildings of the same type.

The PhD research activity addressed the design and development of a platform to monitor thermal energy consumption in a DHS, to efficiently compute the energy performance and forecast future consumption of every building in a city district, and to eventually improve the energy efficiency and the overall management of the DHS. The following two different kinds of analysis are proposed: descriptive analytics and predictive analytics.

Descriptive analytics is used in Section 3.4 to characterize the energy efficiency of buildings by computing different classes of KPIs based on real energy-related data. KPIs are based either on a single energy-related variable or on the joint analysis of manifold energy-related variables. In both cases, their purpose is to make possible a proper comparative analysis of the energy efficiency of a building. KPIs are computed using the MuSTLE framework.

Predictive analytics is used in Section 3.5 to forecast the instantaneous *power demand* and the *heating energy consumption* of a building, based on the relationship between these parameters and the climate conditions surrounding the building. The proposed algorithms estimate the power level of heat exchange during various time slots, using a fine-grained time resolution. The correct estimation of peak values for each building allows energy manager to properly size the HDN and manage the whole DHS for the next days.

The research studies presented in this chapter are based on energy-related data collected in a real world system in the city of Turin (Italy), where several residential buildings are served by the HDN. Energy-related variables are measured from about 4,000 monitored buildings, each one generating about 2,000 data frames per day, thus resulting in a growing database of at least 8 million data frames per day. Meteorological data are collected from the Weather Underground web service [42]. Data from many weather stations are collected to estimate the weather conditions nearby each building.

3.2 Related Work

In Smart City and Smart Grid scenarios, different solutions have been proposed to enable a pervasive monitoring and management of energy data and to provide general purpose services [43, 44, 45, 46, 47, 48, 49]. These solutions are mainly focused on the system architecture and device interoperability, without really innovative systems that continuously collect useful data and provide advanced analytics services to eventually improve energy efficiency.

Significant research activities have been carried out using database management systems, exploratory data mining techniques, and statistical tools in the field of storage and analysis of energy data, to evaluate the energy efficiency of buildings [50, 51]. The proliferation of sensor networks for monitoring indoor and outdoor environmental parameters has brought to facility managers huge archives of measures with temporal and spatial references. Research contributions on these large data volumes have been carried out for: (i) supporting data visualization and notification of anomalies [52]; (ii) efficient storage and retrieval of sensors data for energy data inspection [53, 54]; (iii) characterizing consumption profiles among different users [55, 56]; (iv) identifying the main factors that increase energy consumption (e.g., floors and room orientation [57], location [56]).

A parallel effort has been devoted to designing and implementing systems based on Big Data technologies to provide different kinds of *descriptive analytics* for buildings energy consumption. Proposed solutions are general purpose [58, 59] or tailored to a given application domain, such as thermal energy consumption for space heating [60], residential energy use [61, 62], renewable energy [63], air pollution levels [64]. Authors in [58] highlight the key components that should be included in an analytics cloud service: service management, workflow management and data management. The work in [60] tries to point out the key features of an Energy Management System (EMS), to support frequent pattern discovery on event streams. A Data Stream Management System (DSMS) is used, to better suit the typical queries of real-time EMSs on time-varying data streams. Belussi and Danza [65] presents a research project aimed at establishing an analytical methodology to analyse energy performance of buildings through *energy signature* and to highlight malfunctions. Ghiaus [66] proposes a robust analysis of ES with linear regression, to calculate the *total heat loss coefficient* of the building and the associated external temperature. The data used are referred to the daytime in the range $[10am - 6pm]$, in order to reduce dynamic effects.

Different research efforts have been carried out also in the development of novel *predictive analytics* algorithms to forecast buildings energy consumption and performace indexes (e.g., *power demand*, *energy consumption*, *heat loss coefficient*). Some works have been devoted to characterizing energy consumption, using data driven approaches [67] also with the support of machine learning algorithms like Neural Networks [68, 69], Gaussian Mixture Models [70], and Support Vector Machines [71] as well as energy efficiency by extracting relevant features of heating systems [72] and through Energy Signature (ES) analysis. Sjögren, Andersson, and Olofsson [73] studied the sensitivity of the heat loss coefficient and internal temperature estimated with the ES method to different time periods and gained energy. The data used in the analysis consists of monthly energy used and water use referred to 9 multifamily buildings in Stockholm for the period 2003-2006. Authors in [74] investigate the possibility to predict the electrical gain factor and the heat loss factor in order to describe the building performance. The case study are two buildings in Sweden, where hourly data of internal and external temperature and heating Energy and Power are available. The data were averaged to daily values in order to reduce the dynamic component. Results show a high precision on the estimation of the heat loss coefficient value. Danov et al. [75] show a method for evaluating the heat loss coefficient from daily measurements, taking into account also the dynamic and solar effects. It calculates the dynamic component with respect data from previous days. The results show that the dynamic and the solar gain correction improves the precision of the estimation. Mangematin, Pandraud, and Roux [76] propose a quick methodology to estimates thermal parameters in buildings. The method consist in setting up measurements of energy consumption, internal and external temperatures, in absence of people inside the building and without solar gains. On the other hand, Bogomolov et al. [77] propose an approach to predict energy consumption based on human presence in the building, derived from GSM network call data records.

Contribution of the research activity. The PhD activity described in this chapter brings several innovative elements in the field of data mining for the analysis of buildings energy efficiency. First of all, new building energy KPIs are computed at different space-time granularities using the MapReduce paradigm with the MuSTLE framework (see Section 2.3). Moreover, a platform and a novel algorithm have been devised for the prediction of instantaneous *power demand* values - rather than cumulative energy - which is subject to a higher variability over time. The algorithm, based on the concept of *Energy Signature* and *heat loss coefficient*, works at fine-grained resolutions and allows to detect and quantify peak values. Finally, the proposed platform works in real time and provides updated forecasts as soon as new data are received.

3.3 Proposed data analytics architecture for operational energy rating

The proposed system architecture for operational rating is represented in Figure 3.1. It consists of three main blocks: (i) *Data Collection*, (ii) *Data Management* and (iii) *Application*. The Data Collection block, in common between descriptive and predictive analytics tasks, includes the *Source Layer* and the *Middleware Layer*, which are briefly described below. The Data Management and the Application blocks are specific for each of the analytics task, so their characteristics are described in the respective sections.

Source Layer

The *Source Layer* is the lowest layer in Figure 3.1. It includes different kinds of hardware and software entities that continuously provide various data types of interest.

Hardware entities correspond to *smart meters* and *sensors* measuring physical quantities.

Software entities are software services exposing collected physical measures to external clients. They allow acquiring data values complementary to those collected through

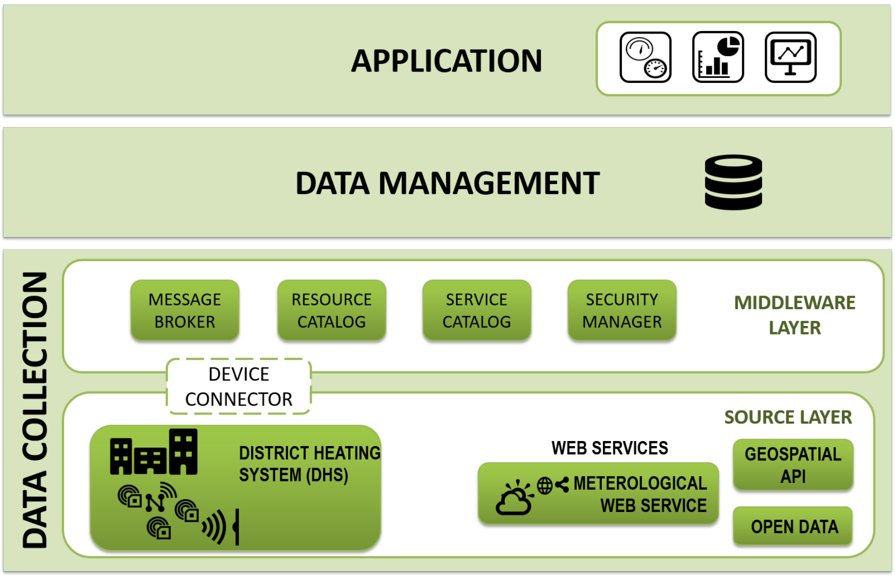


Figure 3.1: The general system architecture for operational energy rating of buildings.

hardware entities, that contribute in the overall characterization of the context under analysis. Web services are an example of software entities that expose interfaces over the Internet allowing clients to send requests and get data using HTTP as transport protocol.

Measurements collected from the hardware and software entities are enriched with additional *spatio-temporal information* useful to describe the spatial and temporal distribution of the acquired values (e.g, the spatio-temporal distribution of thermal energy consumption). To this aim, the Data-source layer includes additional *contextual data sources* such as web services exposing topological data (e.g., municipality Open Data portals [78]) or calendar data. More specifically, the geo-coordinates (longitude and latitude) of each monitoring node are mapped to the corresponding neighborhood and city district. While the geo-referenced location of nodes is given in the hardware/-software entities, both the neighborhood and district names corresponding to the geo-referenced location have been added as additional contextual features. They have been retrieved from contextual data sources. Moreover, each measurement time is associated with different blends of time spans as daily time slot (e.g., morning, afternoon, evening, or night), week day, holiday or working day, month, 2-months, or 6-months time periods.

When a new data source is registered in the Source Layer, all related static data are acquired and stored in the data repository. To effectively support the interoperability across heterogeneous devices, the *Source Layer* includes the *Device Connector*, a middleware-based component that abstracts a given technology and translates its functionalities into Web Services. The Device Connector enables the communication among heterogeneous devices and works as a bridge between the entities in the Source Layer and the overlying Middleware Layer.

The collected data are sent to the *Data Management* block, through the Middleware Layer (described below), where they are preprocessed in order to clean and synchronize data gathered from the different sources. The preprocessing tasks are described later in Sections 3.4.1 and 3.5.1, when the specific application-specific architectures are introduced.

Middleware Layer

The *Middleware Layer* in Figure 3.1 is in charge of providing features to discover available resources and services in the Data-source Layer. It creates a network among different entities that can exchange information exploiting two communication paradigms: (i) request/response based on REST [79] and (ii) publish/subscribe [80] based on MQTT protocol. Such features are key characteristics of a software infrastructure dealing with IoT devices. The Middleware Layer includes four software components described below.

The *Message Broker* allows the communication among different entities (both hardware and software) through the publish/subscribe paradigm. This approach supports the development of loosely-coupled event-based systems. Indeed, it removes explicit dependencies between interacting entities (i.e., producer and consumer of the information), thus each entity in the middleware network can publish data and other subscribers can receive it independently. This increases the scalability of the whole system [81]. PABOR adopts the MQTT communication protocol and delivers data to subscribers as soon as they are measured and published (the delay is negligible).

The *Resource Catalog* registers and provides a list of IoT devices and resources available into the middleware network. It exposes JSON-based REST API to automatically access and manage such information. For instance, Device Connectors register their devices and resources, while other middleware-based entities discover such devices and their access protocols.

The *Service Catalog* provides information about available services in the middleware network exposing a JSON-based REST API. It is used by middleware-based entities to discover available services in the network. For instance, it provides the end-points of services such as Resource Catalog and Message Broker.

The *Security Manager* provides features to enable a secure communication among entities in the middleware network. Indeed, it is in charge of authenticating and granting accesses to applications and other middleware-based components. Hence, malicious actors cannot call services in the middleware network and cannot receive any kind of data.

In the following sections, two different instances of this architecture are presented with the aim of addressing and implementing descriptive analyitics (Section 3.4) and predictive analytics (Section 3.5) methods.

3.4 Descriptive analytics for operational energy rating

As a first step, the research activity has focused on the characterization of the energy efficiency of buildings in a HDN. The activity led to the design and development of an IoT system and of a platform called *Descriptive Analytics for Buildings Operational Rating* (DA-BOR). It uses *descriptive analytics* algorithms to compute different classes of KPIs about thermal energy efficiency of buildings, based on real data on building energy consumption (operational rating). Such KPIs can be used to (i) evaluate the efficient use of a heating system over time and (ii) compare the performances of nearby or similar buildings.

To characterize the building thermal energy consumption and efficiency, different methods have been proposed in the literature by energy scientists and professionals. The definition of concise Key Performance Indicators (KPIs) is one of the main approaches [82]. In this context, we defined KPIs as quantitative indicators of thermal energy efficiency of buildings across different spatial and temporal aggregation layers.

3.4.1 The DA-BOR platform architecture

The DA-BOR platform for the analysis of buildings energy KPIs is represented in Figure 3.2. It consists of four layers: (i) *Source Layer* and (ii) *Middleware Layer* within the Data Collection block; (iii) *Storage Layer* within the Data Management block; and (iv) *Descriptive Analytics Layer* within the Application block.

The DA-BOR platform is based on the general architecture presented in Section 3.3. In particular, the Data Collection block is the same as described before, while the Data Management and the Application blocks are tailored to the analysis presented in this section.

The DA-BOR platform performs distributed and scalable computations of KPIs with the MuSTLE framework, using data aggregation and correlation analysis defined respectively in Section 2.3.2 and Section 2.3.3.

The *Storage Layer* and the *Descriptive Analytics Layer* are described below.

Storage Layer

Data records sent by the gateways, through the Middleware Layer, are managed by the *Storage Layer* component, based on a two-level database architecture.

The first level, named *sensor data storage*, collects the raw data continuously received from smart meters. Due to the fixed and constant nature of those measurements a relational Oracle database is used for their storage (*Sensor Data SQL DB*). The Oracle database has been chosen by the administrator of the DHS to store the time series for each variable measured by smart sensors (hundreds of variables are available for each building heating system).

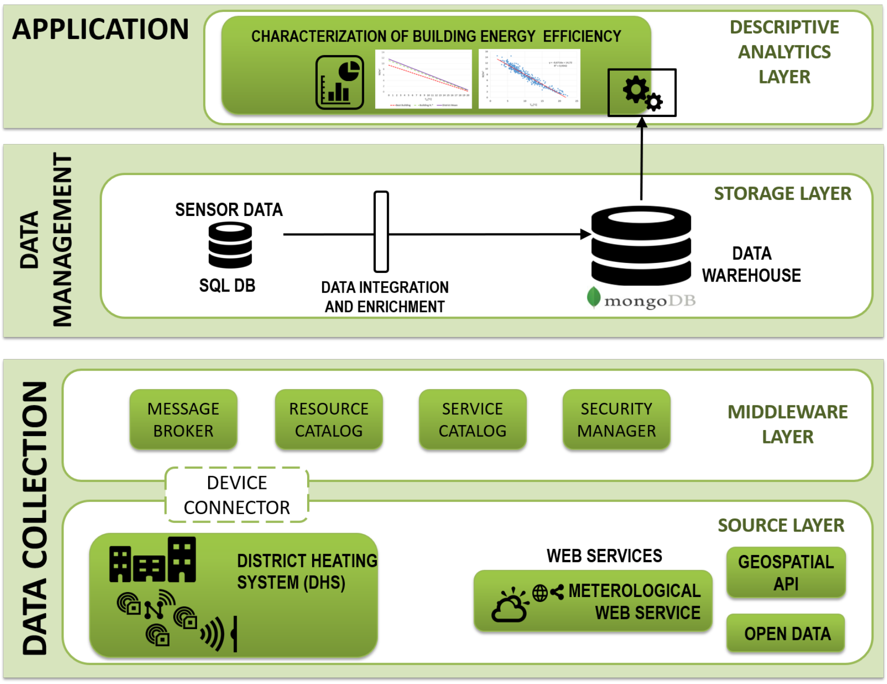


Figure 3.2: The DA-BOR architecture for descriptive analytics of buildings energy data.

Data inside this database are not yet suitable for the analysis. Indeed, preprocessing tasks are necessary to *clean* and *synchronize* data from different sources.

First, missing (or erroneous) values are inferred (or replaced) through the interpolation between the previous and the following measure from the same sensor.

Then, duplicated records (same value and same timestamp) measured by the same sensor are discarded.

Sensor data are then integrated with meteorological data and enriched with topological and contextual (space and time) information at different granularity levels. Data collected through the smart meters are integrated with the following meteorological measures: air temperature (expressed in $°C$), relative humidity (percentage), precipitation level (mm), wind speed (km/h) and sea level atmospheric pressure (hPa). The date and time of each measurement is also included.

As a last step of data preprocessing, the integration of different sources entails the *synchronization* between energy-related data and weather data as follows. For each weather sensor, a specific energy record is associated with the meteorological measures with the closest timestamp. Weather data associated with the energy records of a given building are computed as a distance-based weighted mean of the values provided by the three nearest weather sensors. The weight is inversely proportional to the distance from the sensor to the building address. While the address is an information recorded for the monitored building, the geographical coordinates and both the neighborhood and district names corresponding to the address are added as additional contextual features (*data enrichment*).

The enriched data are eventually stored in the second level of the Storage Layer, the *Data Warehouse*. Being enriched data significantly more variable and heterogeneous than original raw data, their analysis requires a different technological solution. Therefore, for enriched data we exploit the NoSQL distributed database MongoDB.

Documents are sharded across nodes using a hash-based partition on the building ID. This choice makes possible to evenly distribute data across shards when new buildings are included and improves the computation of KPIs for buildings comparison.

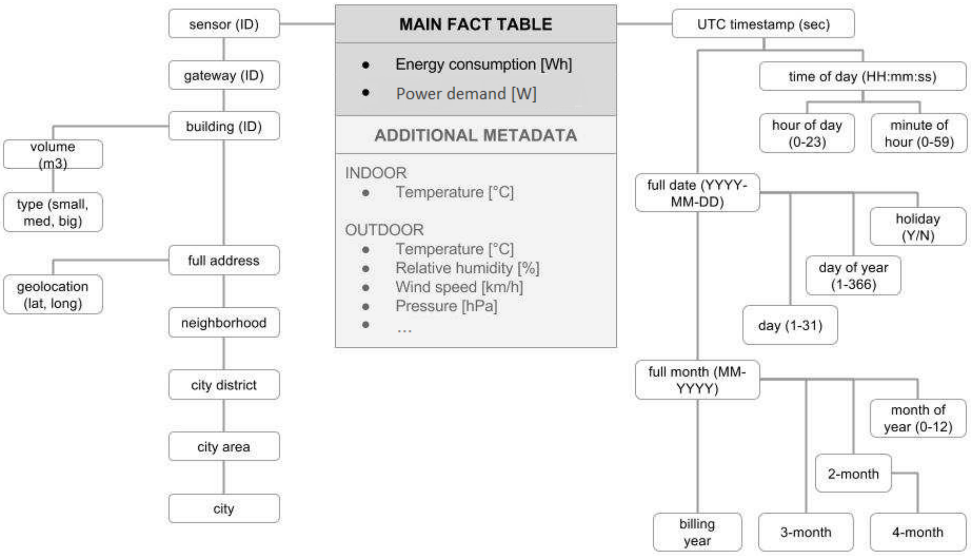


Figure 3.3: The data warehouse design.

In Figure 3.3 the data warehouse conceptual model is presented: the fact table consists of two main measures, the *energy consumption* and the *power demand* in a 5-minute time period, and some additional metadata, about environmental conditions, coming from indoor and outdoor sensors. Two hierarchies are defined: a time-related hierarchy and a space-related one. The former provides many different blends of time spans, from minutes to months and years. The latter starts from the physical sensors inside each monitored building and builds up to the whole city, with the building volume and the geolocation coordinates as related features included in the document.

To analyze the *temporal distribution* of thermal energy consumption, the following time granularities are considered: day, month, 2-month, 3-month, 6-month time periods. Moreover, each day is classified as holiday or not, and the measurement time is aggregated into the corresponding *daily time slot* (morning, afternoon, evening, or night).

To analyze the *spatial distribution* of thermal energy consumption, different space granularities are also considered beyond the building addresses. In addition, each *address* is mapped to the corresponding geographical *coordinates* (longitude and latitude degrees), *neighborhood* and *city district* including that neighborhood.

Descriptive Analytics Layer

The *Descriptive Analytics Layer* component analyzes the collected data and produces useful feedbacks to users. DA-BOR focuses on building performance evaluation through the computation of KPIs that are described in detail in the following Section 3.4.2.

In this layer, the analysis tasks are performed using the *space-time data aggregation* and *data analysis* stages of the MuSTLE framework (Section 2.3), to evaluate energy performance at different space-time granularities: from the single building to the entire district, and from hours and days to months.

3.4.2 Energy efficiency indicators with MuSTLE

Based on the number of employed variables, we can define two classes of energy KPIs.

The first class (*simple KPIs*) makes use of a single energy-related variable (mainly building *energy consumption* for space heating). The computation of *simple KPIs* follows different patterns of spatial and/or temporal aggregation of the building energy consumption.

The second class of KPIs (*multiple KPIs*) is based on the joint analysis of manifold energy-related features. In this study, a widely adopted indicator named *total heat loss coefficient* (K_{tot}) is computed by means of *energy signature* analysis, a world wide recognized method for the comparison of the energy efficiency of nearby/similar buildings and with respect to past trends [65, 83]. The computation of *total heat loss coefficient* makes use of three different variables: the *power demand per unit of volume* (Q_h), the *internal temperature* of the building (T_{in}) and the *external temperature* nearby the building (T_{ex}).

KPIs of both classes can be computed according to three different levels of time interval: (i) the *time period* t_{period} represents the whole analysed time range, from the timestamp of the first to the timestamp of the last available data sample; (ii) the *daily time window*, t_{window} represents, within each day included in t_{period}, a subset of time intervals selected for the analysis (e.g., from 7am to 6pm); (iii) the *time slot*, t_{slot} represents the target data granularity for each variable, which assumes a single aggregated value every t_{slot} and only during the given t_{period} and t_{window}.

Simple KPIs based on space-time aggregation

Four simple KPIs have been defined through spatio-temporal aggregation of energy consumption values.

- *Building KPI.* Average energy consumption of a single building per unit of volume, i.e., total energy consumption of the building divided by the building total volume.

- *Neighborhood KPI.* Average energy consumption of the buildings in the same neighborhood (e.g., a whole district) per unit of volume.

- *Time-slot KPI.* Average energy consumption of the buildings during a given daily slot (morning, afternoon, evening) per unit of volume.

- *Building-type KPI.* Average energy consumption of the buildings of the same type (small, medium, large) and in the same neighborhood per unit of volume.

As an example, consider the *Building KPI* defined above. All KPIs are computed in MuSTLE by exploiting *map*, *reduce* and *finalize* functions as follows. The *map* function determines the key and value pairs emitted by each processed document (Listing 3.1): the key is used to group documents, similarly to the SQL clause *group by*, and in this case it corresponds to the building ID; whereas the value is a more complex object, since to compute an average we need to take both the consumption sum and the building volume. Hence, we put these two values into the value object returned by the *map* function.

Listing 3.1: Map function for Building KPI computation in MapReduce

```
map_function() {
key = this.place.building.id;
value = {
ec: this.energy_consumption,
vol: this.place.building.volume};

emit(key, value);
}
```

The *reduce* function receives a list of values from the map functions with the same key, hence we have a list of objects containing the energy consumption value (*ec*) and the building volume (*vol*), and we need to sum all the *ec* values of the list (Listing 3.2). The building volume is the same for all energy consumption values, since they refer to the same building.

Listing 3.2: Reduce function for Building KPI computation in MapReduce

```
reduce_function(key, values) {
reduced_value = {
ec: 0,
vol: values[0].vol};

for (var i=0; i<values.length; i++)
reduced_value.ec += values[i].ec;

return reduced_value;
}
```

After the reduce phase we have a list of value objects, one for each building id (key), containing the total energy consumption and the building volume. The *finalize* function adds to each object in this list the average value, which is the final result and corresponds to the desired KPI (Listing 3.3).

Listing 3.3: Finalize function for Building KPI computation in MapReduce

```
finalize_function (key, value) {
value.ec_vol = value.ec/value.vol;
return value;
};
```

Total heat loss coefficient based on Energy Signature

The *total heat loss coefficient* K_{tot} is a measure of the rate of heat energy flowing outside the building's envelope with respect to the difference between indoor and outdoor temperatures. K_{tot} can be estimated at different granularity levels, through the approach based on *Energy Signature*, which is the plot of the mean power (or energy) demand of a building versus the mean external air temperature.The *Energy Signature* method is useful to highlight the correlation between these variables, hence it is exploited to study the relationship between real data from smart meters. For a detailed description of Energy Signature method and for the definition of total heat loss coefficient, please refer to [41].

Taking inspiration from Energy Signature, this study analyses the correlation between the mean value of *power demand per unit of volume* supplied by the heating system (Q_h) and the mean difference between the *internal temperature* of the building (T_{in}) and the *external temperature* nearby the building (T_{ex}), at different time granularity levels. As described in [41], we assume the existence of a linear regression (see Section 2.3.3) between Q_h and $T_{in} - T_{ex}$, i.e., $Y = aX + b$, where $Y = Q_h$, $X = T_{in} - T_{ex}$, and the a value corresponds to the building *total heat loss coefficient* K_{tot}.

Given a data set of n samples $(x_i, y_i) \in X \times Y, i = 1, \dots, n$, a *simple linear regression* between X and Y is described by a linear equation in the form $Y = aX + b$ that fits the plot of values (x_i, y_i). The a coefficient (and K_{tot}) can be computed as:

$$a = K_{tot} = \frac{(\sum_{i=1}^{n} y_i)(\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} x_i y_i)}{n(\sum_{i=1}^{n} x_i^2) - (\sum_{i=1}^{n} x_i)^2} \tag{3.1}$$

To evaluate the quality of the linear regression that estimates K_{tot}, the *Standard Error of Regression* S [84] is used:

$$S = \sqrt{\frac{1}{(n-2)}[\sum (y - \bar{y})^2 - \frac{[\sum (x - \bar{x})(y - \bar{y})]^2}{\sum (x - \bar{x})^2}]} \tag{3.2}$$

where $\bar{x}$ and $\bar{y}$ are the sample means, and n is the sample size. Small values of S identify a high accuracy of prediction because the 95% of predicted values fall in a range of $\pm 2S$.

Given the mean power demand values (denoted as y) and the mean temperature difference values (denoted as x) for a single building, to speed up the computation of K_{tot}, we compute in the MuSTLE framework the terms $\sum_{i=1}^{n} x_i$, $\sum_{i=1}^{n} y_i$, $\sum_{i=1}^{n} x_i y_i$ and $\sum_{i=1}^{n} x_i^2$. MapReduce jobs are executed as follows. For each document, the *map* function emits an object containing the values needed to compute the energy signature equation parameters for the related building. The *reduce* function is in a simple sum over all the records of the same building. Finally, a *finalize* function uses the aggregated results to compute the energy signature equation and returns the K_{tot} estimation for each building.

3.4.3 Experimental Results

In this section, DA-BOR is experimentally evaluated on real data collected from the use case described in Section 3.1. Experiments are aimed at both the characterization of the Energy Signature and the computation of the K_{tot} value for a set of buildings.

DA-BOR **implementation**

DA-BOR platform was deployed on a cluster of 8 nodes configured as a MongoDB *sharded cluster* consisting of three different components. The nodes were assigned to each component as follows: (i) Up to 5 dedicated nodes (node4 to node8) were configured as the actual shards in charge of the data storage. (ii) One node (node2) was configured as *query router* (mongos) and were in charge of directing operations to the appropriate shards. (iii) Three nodes (node1 to node3) were configured as *Config servers* (mongod –configsvr) to store the cluster's metadata, such as the mapping of the data set to the shards. Each cluster node is a 2.67 GHz six-core Intel(R) Xeon(R) X5650 machine with 32 Gbyte of main memory running Ubuntu 12.04 server with the 3.5.0-23 kernel.

The algorithms for the computation of KPIs and of K_{tot} value were implemented in MongoDB MapReduce with the JavaScript language.

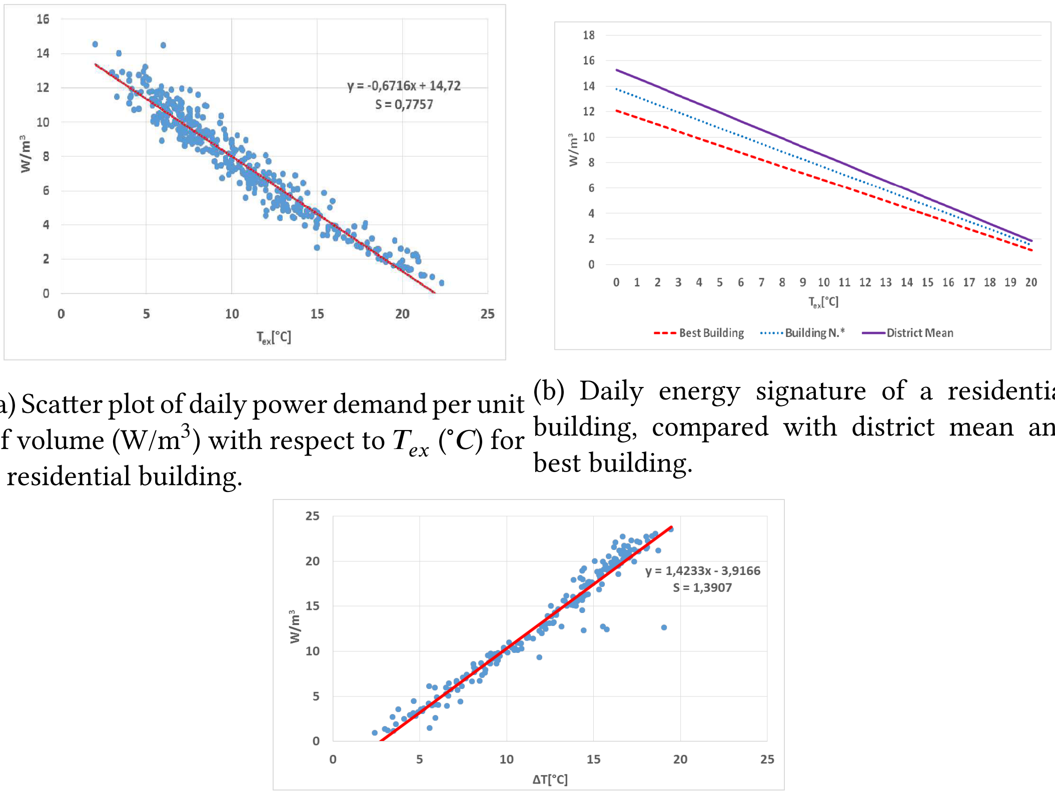
Evaluation of building Energy Signature

As a sample use case, we analysed data from about 4 thousands monitored buildings (see Section 3.1).

The scatter plot in Figure 3.4a shows the daily power demand per unit of volume with respect to the external temperature for a random residential building in the Turin area. The chosen t_{slot} value is 24 hours, hence the analysis considers the daily mean power values per unit of volume with respect to the daily mean outdoor temperature[1].

[1] In most residential buildings, the indoor temperature is not monitored through a sensor network,

t_{period} is set to the full Italian heating season from October 15th, 2013 to April 14th, 2014. t_{window} is been set to the time range from 5pm to 9pm. The red line represents the curve of linear regression and the value of its slope is K_{tot}. A low S value of 0.78 is obtained, whereas the estimated value of K_{tot} is 0.67.



(a) Scatter plot of daily power demand per unit of volume (W/m^3) with respect to T_{ex} (°C) for a residential building.

(b) Daily energy signature of a residential building, compared with district mean and best building.

(c) Residential building, scatter plot of daily power demand per unit of volume (W/m^3) with respect to T_{in}-T_{ex} difference (°C).

Figure 3.4: Energy Signature plots for residential buildings

The plot in Figure 3.4b shows the energy signature of the same building as before (dotted line) with respect to the energy signature of the most efficient building (dashed line) and the average power profile (solid line) within the same district. Such comparative information allows users to rank buildings within districts, immediately putting in perspective the initial value of $K_{tot} = 0.67$: even if this value is better than its district mean value, it is far worse than the best performing building in the same district.

Figure 3.4c shows the daily power demand per unit of volume of a building for which

hence we considered in the analysis a fixed value of 20°C, as a typical value set by local regulations. Being T_{in} fixed, the chart reports T_{ex} only.

indoor temperature T_{in} measures were available. The analysis has been performed by considering t_{period} of almost two heating seasons in Turin, being data measured from October 15^{th}, 2013, to February 28^{th}, 2015. The analysis has been performed by considering values from 5pm to 9pm and data are aggregated over daily t_{slot}. The value of K_{tot} estimated from the linear regression is 1.42 ($S = 1.39$), indicating a poor energy performance, at least when compared with residential buildings.

3.4.4 Computing performance of algorithm

We evaluated the scalability of the proposed algorithm for the computation of the building Energy Signature by measuring the speed-up achieved for different numbers of shards in the MongoDB cluster (from 1 to 5 nodes). The speed-up is computed as the ratio between the time needed to process the whole task with 1 computing node and the time needed to process the same task with n nodes. The MongoDB *chunk size* parameter that determines the sharded data balance among the nodes was left to its default value of 64 MB, being already more than three orders of magnitude smaller than the total data collection size of almost 300 GB, and thus leading to finely-grained balanced shards. The chosen shard key for the experiments is the *Building ID*.

Figure 3.5 reports the speedup achieved with the 2,000 building data set. The black line represents the (positive side of the) ramp function, under which the computing speedup would be identical to the number of shards or, equivalently, the computing time for a cluster would be equal to that employed by a single shard divided by the number of shards in the cluster. The achieved results show that the algorithm scales roughly linearly with the number of nodes and the speedup approximately corresponds to the number of cluster nodes.

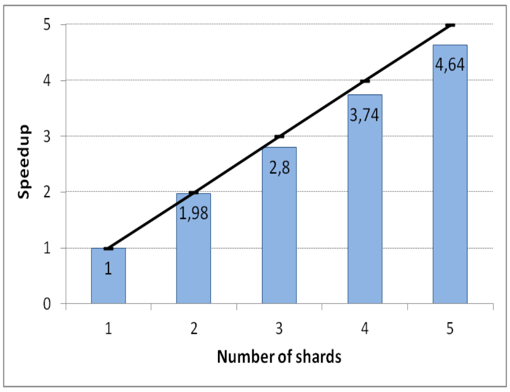


Figure 3.5: Speedup on a 300GB-sized data set (2,000 buildings).

3.5 Predictive analytics in real time for operational energy rating

In this section, the research activity on operational energy rating in a DHS is focused on the prediction of future thermal *power demand* and *energy consumption* values of a building, at different time granularities.

The activity led to the design and development of a platform called *Predictive Analytics for Buildings Operational Rating* (PA-BOR). It uses *predictive analytics* algorithms to predict the values of thermal *power demand* of buildings, based on real forecasts about environmental conditions and on past power demand data. Predicted values can be used to support energy managers in taking appropriate actions in advance.

The experimental validation of the proposed PA-BOR platform has been conducted on the use case presented in Section 3.1.

The prediction of the building power demand is a very complex task to be addressed, due to the high variability of the power profiles of buildings characterized by different daily *heating cycles*. Therefore, innovative and mixed analytics solutions are needed to effectively address the prediction of both the building *power exchange* and the building *peak power demand* during the heating cycles. Each heating cycle is composed by two main *operational phases*: the *OFF-line* phase, when the heat exchange is turned off, and the *ON-line* phase, when the heat exchange is on. The ON-line phase is then further structured in the alternation of two sub-phases, named *ON-line transient state* and *ON-line steady state*. Figure 3.6 reports three examples of daily power profile for *Single*, *Double* and *Triple* daily *Heating Cycle*. Note that consecutive heating cycles can show different peak power values.

The PA-BOR methodology gathers and integrates physical measures collected through *sensors* and *smart meters* with third-party information provided by Web services. In particular, fine-grain power consumption data have been integrated with the most common meteorological indicators (like air temperature, relative humidity, precipitations, wind speed, atmospheric pressure) collected through weather stations distributed throughout the city.

To deal with the high variability and mixed trend of the power profiles of buildings and to achieve accurate predicted values, PA-BOR implements a prediction model composed by three contributions applied in cascade. (i) First, the automatic identification of the *operational phases* described above (*OFF-line*, *ON-line transient* and *ON-line steady*) is automatically performed. Then, it forecasts (ii) the *peak power* values during the *ON-line transient* phase and (iii) the *power level* during various time instants of the *ON-line steady* phase.

3.5.1 The PA-BOR platform architecture

The overall system designed to implement PA-BOR consists of a five-layered architecture, shown in Figure 3.7, with: (i) *Source Layer* and (ii) *Middleware Layer* within

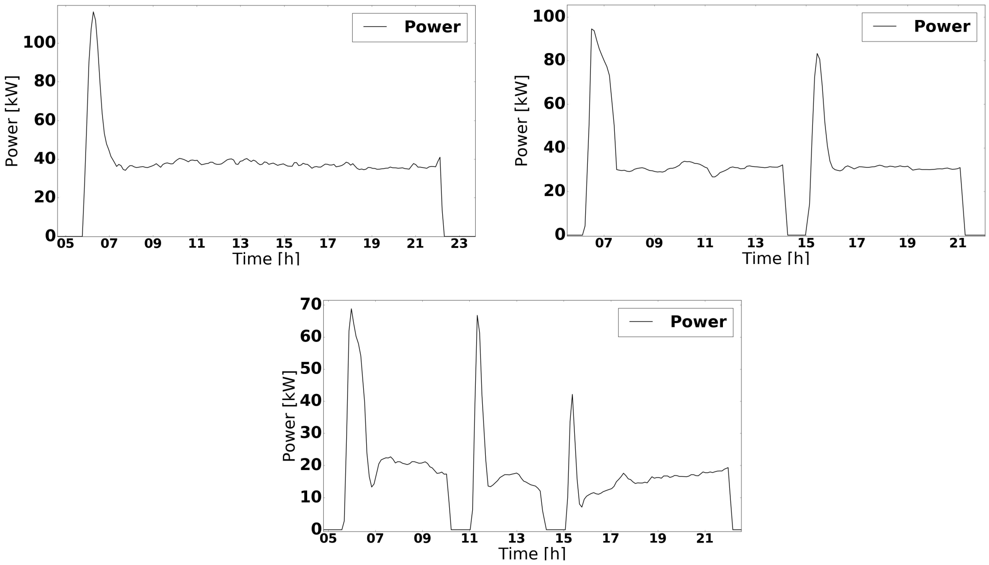


Figure 3.6: Heating Cycles in a day

the Data Collection block; (iii) *Integration Layer* and (iv) *Storage Layer* within the Data Management block; and (v) *Predictive Analytics Layer* within the Application block.

The PA-BOR platform is based on the general architecture presented in Section 3.3. In particular, the Data Collection block is the same as described before, while the Data Management and the Application blocks are tailored to the analysis presented in this section.

Compared to the architecture described in Section 3.4.1, it includes some important features in the Data Management block to support the processing of data in (near-)real-time: (i) the additional *Integration Layer* to support synchronization of data streams before storage and analysis, and (ii) the removal from the Storage Layer of the staging area (*Sensor Data SQL DB* in Figure 3.2), bypassed by data streams that are synchronized in real time and stored in the *operational database*.

The *Integration Layer*, the *Storage Layer* and the *Predictive Analytics Layer* are described below.

Integration Layer

Data coming from different sources are not yet ready for the analysis, but they need to be properly *cleaned* and *synchronized*.

First, missing (or erroneous) values are set (or replaced) with the value of the previous measure from the same sensor.

Then, duplicated records (same value and same timestamp) measured by the same sensor are discarded.

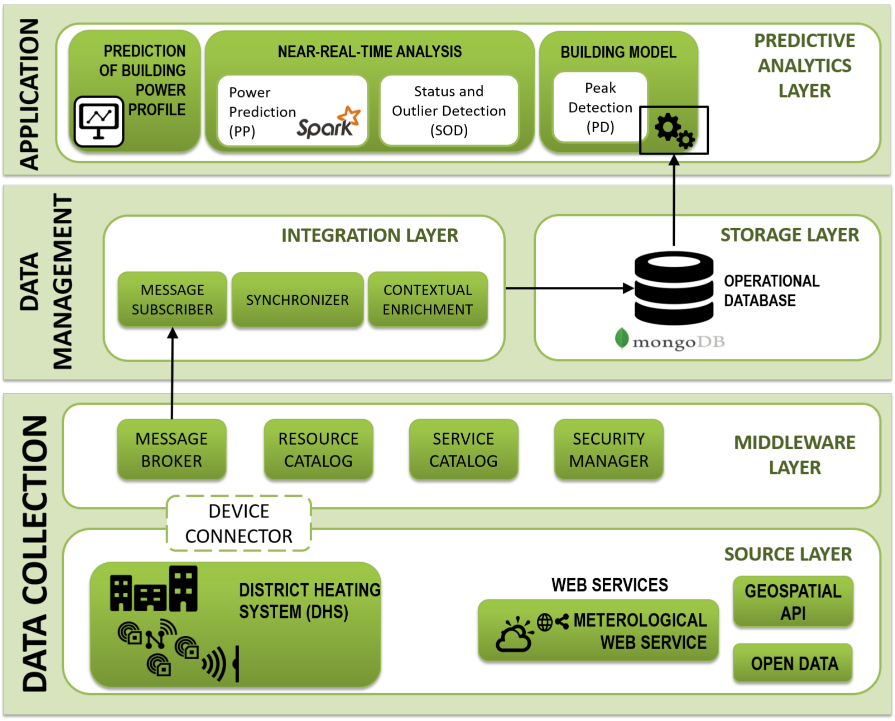


Figure 3.7: The PA-BOR architecture for predictive analytics of buildings energy data.

For each data source, monitoring nodes may be deployed in different city areas and they may adopt a different timeline in sampling values. Thus, a proper strategy should be devised for the spatio-temporal integration of the acquired measurements.

The *Synchronizer* module manages the time alignment between weather data and power data, before storing them together in the data repository. The synchronization is performed in real-time, without using a data staging area like the one in Figure 3.1 of Section 3.4.1. Specifically, each *power measure* collected for a building is associated with a set of *weather measures* (e.g., temperature, humidity, and pressure) that describe the climate condition when the power measure was collected. For each weather station, only weather measures received during a small time neighborhood of the power data timestamp are used and associated with the closest power measure in time. The weather values associated with power data (e.g, temperature) are calculated as the weighted mean values of the measures acquired from N weather stations located near the building and during the given time neighborhood. A weight is associated with each weather station based on its spatio-temporal proximity to the building. It expresses the relevance of the measure provided by the weather station. The weight is higher for stations closer to the building and for measures closer to the power data timestamp, since they provide a more accurate value on the climate condition at the bulding proximity.

Storage Layer

Due to the different kinds of collected data and to easily manage more data types in the future, enriched data are stored as JSON documents in the *Historical Datastore*, which corresponds to the data warehouse described in Section 3.4.1. The collection of historical data is then exploited to create models of the energy consumption for the buildings and for the near real-time data analysis, including building power prediction.

According to the objectives of the data analysis tasks described in Section 3.5.1, we evaluated as optimal choice the adoption of the data processing framework Apache Spark [85] upon MongoDB data repository (see Section 3.5.3). Indeed MongoDB stores data across different nodes (called shards), thus supporting parallel processing by Spark. This distributed architecture provides higher levels of redundancy and availability, which are fundamental when operating in (near-)real-time, and to scale and satisfy the demand of a higher number of read and write operations. Since both Spark and MongoDB adopt a document-oriented data model, they exchange data in a seamless way by making use of the JSON serialization format. This way, Spark jobs are executed directly against the Resilient Distributed Datasets (RDD) created automatically from the MongoDB data repository, without any intermediate data transformation process. Moreover, due to the real time nature of the data analysis, input data sets vary rapidly in time. To improve the performance of the several queries to be executed, MongoDB rich indexing functionalities is exploited in Spark, like secondary indexes and geospatial indexes, that allow to efficiently filter data according to the geospatial coordinates of buildings and nearby weather stations.

Predictive Analytics Layer

In this study, PA-BOR is used to predict fine-grained power level values during the heating cycle of buildings. The data prediction process is structured into three main blocks: (i) *data stream processing* to support (near-)real-time data analysis, (ii) *prediction analysis*, and (iii) *prediction validation*. The main functionalities of the three blocks are briefly presented below and detailed in Section 3.5.2.

Data stream processing. Since thermal energy consumption is monitored roughly every 5 minutes in the HDN, a large volume of energy-related data is continuously collected for each building. To efficiently and effectively analyze such large data collection, the PA-BOR engine performs the power level prediction task through the data stream analysis over a sliding time window, separately for each building. Every time a new measure of power level is collected, one single time window, sliding forward over the data stream of energy-related data, is considered for the prediction task. This window contains: the recent past *energy-related data* for the building heating system, corresponding to thermal *power levels*; and data about *weather conditions* when power measures were collected.

Prediction analysis. This block entails to predict the average future power levels for each building. A prediction model is built for each building separately by considering the energy-related data in the current sliding time window. The building model is then exploited for forecasting the average power level at a given time instant in the near future.

In a HDN, the heating cycle of a building includes two main operational phases: the *OFF-line* phase, when the power exchange is turned off, and the *ON-line* phase, when the power exchange is on. The ON-line phase is then further structured in the alternation of two sub-phases, named the *transient state* and the *steady state*. More in detail, a large exchange of power between building and HDN (*transient state*) interleaves a quasi-constant power exchange between building and HDN (*steady state*). To deal with this mixed trend and achieve an accurate predicted value, PA-BOR proposes a prediction model composed of three contributions applied in cascade:

- *Step 1.* The *Status and Outlier Detection* (*SOD*) algorithm automatically identifies the operational phases of the heating cycle of a building (Section 3.5.2). Given a power measurement in a time instant, the SOD algorithm labels its operational phase as *OFF-line* or *ON-line* (further categorized as *transient* or *steady* state).

- *Step 2.* The *Peak Detection* (PD) algorithm predicts the peak power value in the *ON-line transient* state (Section 3.5.2).

- *Step 3.* The *Power Prediction* (PP) algorithm predicts the average power profile in the whole *ON-line* phase (Section 3.5.2).

Prediction validation. This block measures the ability of the PA-BOR engine to correctly predict the energy consumption values achievable by a building in an upcoming time instant. To this aim PA-BOR integrates two metrics named *Mean absolute percentage error* (*MAPE*) and *Symmetric mean absolute percentage error* (*SMAPE*). Every time a real power level value is received, their values are updated to include the prediction error for the new measure.

Data flow for predictive analytics tasks

This paragraph describes the data flow implemented in PA-BOR to gather data from different sources and to support the predictive analysis in (near-)real-time. The flow is rather different from the one implemented for the use case of Section 3.4, due to the presence of streams of data collected in real time.

As shown in Figure 3.8, the MQTT protocol is used to publish energy-related measures as messages with an associated topic. A message includes the power value measured on the heating system of a building, the identifier of the same building and the timestamp of the measurement. Messages are asynchronously collected by the *Message Broker*, using the publish/subscribe mechanism, and distributed to all interested subscribers. Therefore, subscriber nodes are responsible for gathering data notifications about new power measures published by IoT devices to the *Message Broker*.

In our scenario, among the subscribers there are the (near-)real-time algorithms, i.e., *Power Prediction* (PP) and *Status and Outlier Detection* (SOD). Each algorithm independently receives energy-related data, sent by IoT devices, from the Message Broker and retrieves meteorological information from third-party web services through REST interfaces [79]. Furthermore, the algorithms gather data from the *Building Model*, included results from the *Peak Detection* (PD) algorithm, which works with already collected historical data. Finally, the results of (near-)real-time algorithms are stored into the MongoDB *Historical Datastore*.

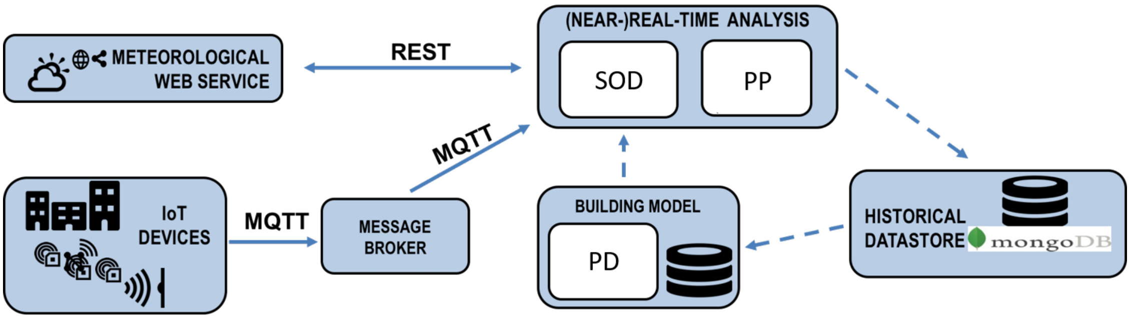


Figure 3.8: Data flow feeding the three algorithms, Status and Outlier Detection (SOD), Peak Detection (PD), Power Prediction (PP).

The PP algorithm uses the energy-related measures received from the *Message Broker* to develop the building model for the prediction of future power values. The PP algorithm periodically updates all building models with the newly available power measures. PP contextually exploits the received measures to calculate the errors of the predictions previously performed for the related time slots, in order to validate the model. It computes the prediction error based on the expected power values according to the prediction model and the actual power value just received. After data have been processed, they are stored in the *Historical Datastore* together with the produced outcomes.

3.5.2 Power demand prediction algorithms

The purpose of our analytics methodology is to predict the future power profiles in the heating cycles of the building heating systems. To achieve this objective, PA-BOR integrates the three algorithms introduced in Section 3.5.1: *Status and Outlier Detection* (*SOD*), *Peak Detection* (*PD*), and *Power Prediction* (PP). All the algorithms elaborate building models based on, and trained with, a collection of historical data retrieved from the Historical Datastore. Each algorithm defines an appropriate *time window* in the past from which data are taken for training. The adoption of the windowing approach allows considering only the recent past data, while excluding a lot of too old samples. As a consequence, the training phase becomes faster, and the generated model fits the behaviour of the heating system just during the selected period.

Since the prediction of power values is actually performed during the third phase of PA-BOR (PP), the following paragraphs and the experimental section will be mainly focused on the PP algorithm.

Status and Outliers Detection (SOD)

The *Status and Outlier Detection* (SOD) algorithm aims at automatically identifying the current operational phase for the building heating system. SOD also allows to detect abnormal values of the instant power measurements potentially occurred in the steady state.

The operational phases of the heating cycle are the *OFF-line* and *ON-line* phases, with the latter characterized by the alternation of a *transient* and a *steady* state. The *transient* state is characterized by a rapid increase of exchanged power. It usually occurs in the early morning when the heating is turned on. The *steady* state occurs after a *transient* state. It is a relatively constant exchange of power. The SOD algorithm relies on such expected trends in the power exchange to detect the operational phase based on the measured instant power values. Specifically, SOD adopts the *Exponentially Weighted Moving Average* (EWMA), proposed by [86] to filter noise and the effects of dynamic transient for the identification of faulty sensors. In this case EWMA is applied to detect the dynamic transient of the heating cycle and those variations in the steady state that can be filtered similarly to noise in a signal.

The SOD algorithm also detects and removes abnormal power values measured during the steady state. An abnormal value is an observation that lies outside the expected range of values. It may occur either when a measure does not fit the model under study or when an error in measurement occurs (e.g., caused by faulty sensors). SOD categorizes this abnormal value as an outlier. When the operational phase is the steady state, a single isolated power measure is categorized as outlier if its value is out of the current range characterizing the steady state.

Peak Detection (PD)

The *Peak Detection* (PD) algorithm aims at forecasting the peak power value in the transient state and identifying the peak power time instant, separately for each building. The PD algorithm is employed to forecast the peak power value in each transient state (also more times per day).

To predict the peak power value in the transient state, the PD algorithm hypothesizes a relation between two quantities, named ψ and τ. ψ is the ratio between the peak power value in the transient state and the mean power value in the previous steady state. τ is the mean external temperature value in the previous steady state and OFF-line phase. To properly model the relationship between the ψ and τ values for any of the three classes of buildings, the PD algorithm relies on the modified version of *Multivariate Adaptive Regression Spline* (MARS) [87] proposed in [88]. MARS is a step-wise linear regression for fitting variables in distinct intervals by connecting different splines with knots, thus it is suited to model a wide class of non-linear relations between variables. PD learns a regression model for each building and for each peak using as training set the data collected in the past days. ψ and τ represent respectively the dependent and independent variables of the regression. Since all the other quantities of ψ and τ are known (from past data), the peak power value of the transient state appearing in ψ is the final target of the prediction.

The PD algorithm also infers the instant at which the peak power will occur. To this aim, PD computes the mean time where the past peaks have occurred, by considering a sliding window of fixed size preceding the current instant of time.

Power Prediction (PP) with multiple regression

On the basis of the outcomes of the SOD and PD algorithms, the *Power Prediction* (PP) algorithm exploits the multiple version of the *Linear Regression with Stochastic Gradient Descent* (LR-SGD) [89] to predict the average power levels in (near-)real-time, based on data from the Historical Datastore.

PP defines a *building model* based on a linear dependency between weather data and power level. PP relies on the assumption that the average power exchange for a building heating system at a given time instant is likely to be correlated with the surrounding weather conditions [41]. The considered variables satisfy all the assumptions of linear regression (linearity and additivity, statistical independence of residuals, homoscedasticity, normality of residuals) except one. In particular, the *statistical independence of residuals* is not always satisfied during the *ON-line steady* phase. Nevertheless, the results of Energy Signature analysis obtained in Section 3.4.3 highlight a clear linear dependence between the two variables in most of the analyzed buildings (see Figure 3.4c). Therefore, this study aims at assessing the capacity of the proposed algorithm to predict instantaneous power values based on the theoretical concept of *Energy Signature*.

PP trains a multiple linear regression model for each building using historical data on weather conditions and power level. The training set is built using a fixed width sliding window mechanism, so the samples not older than a certain amount of time before the current time instant are included in the *training window*. For collected samples, we assumed to split the time window in slots of the same duration (*slot duration*).

For each time slot, a single value is computed for each variable (power and weather parameters) as the mean value of the measures taken during that slot. Data sampling is performed for both training and test (i.e., future time slots) datasets. The PP regression models of all buildings are rebuilt every *slot duration*, in order to include the samples newly collected.

The LR-SGD algorithm is characterized by a set of input features expressed through a n-dimensional vector $\mathrm{x} = [x_1, \dots, x_n] \in \mathbb{R}^n$ and a target variable $y \in \mathbb{R}$ representing the objective of the prediction. The LR-SGD algorithm builds a hypothesis function $h : \mathbb{R}^n \rightarrow \mathbb{R} \mid y = h(\mathrm{x})$ so that, given an input vector x, function $h(\mathrm{x})$ provides a good estimation of the value of y. In our study, features in x correspond to the weather variables (air temperature, humidity, precipitations, wind speed, pressure), while y is the power level. Since power consumption and meteorological values differ in scale and measurement unit, data have been normalized. To preserve the original data distribution without affecting the prediction accuracy, the *Z-Score* standardization technique has been adopted.

The PP algorithm is structured into two phases: (i) *building model learning*, considering a collection of historical values for variables x and y; (ii) *prediction* of the future values of y, using the model generated in the first phase. The two phases are described below.

Model learning. This phase takes as input a training set where each training sample includes both the input vector x of meteorological data values and the corresponding known target variable y. The training set is built using a fixed width sliding window mechanism. Given a time instant t_i, the *training window* includes an ordered sequence of m data samples collected in t_i and in the previous $m-1$ instants t_j ($t_j < t_i$). If the width of the training window (*training window size*) is very short, then almost instantaneous evaluation of the building's consumption is performed. Instead, a too large training window allows analyzing many data on past building energy performance, but it may introduce noisy information in the prediction analysis. Since the data of training window are sampled in slots, the time interval between two consecutive training samples is fixed (*slot duration*). Given time t_i, we define as prediction time t_p the subsequent instant at which PP predicts the average power consumption. The time gap $\|t_p - t_i\|$ defines the *prediction horizon*.

In a training set of m samples defined over a training window, each sample $s^{(j)}$ is expressed by the pair ($\mathrm{x}^{(j)}$, $y^{(j)}$). For the LR-SGD algorithm, the hypothesis function $h(\mathrm{x})$ is expressed as follows:

$$h(\mathrm{x}) = w_0 + w_1 \cdot x_1 + \ldots + w_n \cdot x_n \tag{3.3}$$

where $w_1, \ldots, w_n$ are the weights characterizing the relationship between the average power consumption y and meteorological data values in x (i.e., $x_1, \ldots, x_n$), while w_0 is the intercept value. Without lack of generality, by defining x_0=1 Equation 3.3 can be expressed using the following concise expression:

$$\begin{gathered} h(\mathrm{x}) = \sum_{i=0}^{n} w_i \cdot x_i = \mathrm{W}\mathrm{x}^T, \\ \mathrm{W} = [w_0, \ldots, w_n],\ \mathrm{x} = [x_0, \ldots, x_n]. \end{gathered} \tag{3.4}$$

In the training phase, the LR-SGD algorithm learns the values of weights in vector W. The least-squares cost function $J^{(j)}$ in Equation 3.5 is used to measure the distance between the actual value of y and the computed value $h(\mathrm{x})$ for each training sample ($J^{(j)} = y^{(j)} - h(\mathrm{x}^{(j)})$). The overall least-squares cost function on the whole training set is computed as

$$J(\mathrm{W}) = \frac{1}{2}\sum_{j=1}^{m}(J^{(j)})^2 = \frac{1}{2}\sum_{j=1}^{m}(y^{(j)} - h(\mathrm{x}^{(j)}))^2. \tag{3.5}$$

Algorithm 1 reports the process for weight computation in LR-SGD. The algorithm iteratively considers the samples in the training set. It progressively updates the values of weights w_i in W by following the direction of steepest decrease of $J^{(j)}$. The algorithms is driven by two user-specified parameters: the *learning rate* α and the *number of iterations* on the whole training dataset.

ALGORITHM 1: Weights update in Stochastic Gradient Descent

```
for $j$ = 1, ..., $m$ do
    for $i$ = 0, ..., $n$ do
        $w_i := w_i + \alpha \cdot ((y^{(j)}) - h(\mathrm{x}^{(j)})) \cdot x_i^{(j)}$
    end
end
```

Unlike Batch Gradient Descent (BGD), which updates weights after the whole training set is processed, with the Stochastic Gradient Descent (SGD) approach the overall cost function $J(\mathrm{W})$ quickly converges to a value close to the minimum.

Prediction. Once the learning model has been created, it is used to predict the future power level y using the corresponding vectors of known input features x representing meteorological data values $\mathrm{x}^{(j)}, j = m + 1, \ldots, +\infty$. Hence, given the prediction of the weather variables for a future target time ($\hat{\mathrm{x}}^{(j)}$) and the hypothesis function for the model $h(\mathrm{x})$, the estimation of the corresponding power value is calculated as:

$$\hat{y}^{(j)} = h(\hat{\mathrm{x}}^{(j)}) = \sum_{i=0}^{n} w_i \cdot \hat{x}_i^{(j)}. \tag{3.6}$$

The prediction of weather variables are collected, together with other weather data, from Meteorological Web services [42]. Alternative approaches for the weather time series forecast include the use of ARMA and ARIMA models [90]. However, these approaches are more effective when large time periods are considered for the analysis (some years) and weather data are expressed with coarse granularity (e.g., average monthly value). Such values are not compatible with the objective of the analysis. Indeed, the experimental analysis of PA-BOR is limited only to a 5-months period (autumn-winter) and is aimed at forecasting power values (and weather variables) with a far higher time resolution (*slot duration*).

The PP algorithm also relies on the outcome of the SOD and PD algorithms. Through SOD, PP can identify when the power prediction is performed for the transient or the steady state. Moreover, since during transient state the power values might not have a clear linear dependence from weather data, PP uses the outcome of PD algorithm to better approximate the transient power profile, through a linear interpolation.

To measure the ability of the proposed IoT-based engine to correctly predict the average power consumption values achievable by a building, PA-BOR integrates two metrics: (i) *Mean Absolute Percentage Error* (*MAPE*) and (ii) *Symmetric Mean Absolute Percentage Error* (*SMAPE*). The two corresponding expressions are respectively reported in Equation 2.4 and in Equation 2.5 (Section 2.4).

3.5.3 Experimental Results

We experimentally evaluated PA-BOR on real data collected from the use case described in Section 3.1. Experimental validation has been designed to address the following issues: (i) the error of Power Prediction at different time horizons (Section 3.5.3); (ii) the sensitivity and robustness of the analytics methodology (Section 3.5.3); and (iii) the scalability of PA-BOR with respect to the number of nodes in the cluster (Section 3.5.4).

Results are shown for a smaller cluster of 12 buildings with different heating cycles: (i) 5 buildings with a Single Heating Cycle, (ii) 2 buildings with a Double Heating Cycle and (iii) 5 buildings with a Triple Heating Cycle. To evaluate the computational scalability of the algorithms, we considered a larger data set of 300 buildings (see Section 3.5.4).

PA-BOR **implementation**

PA-BOR's current implementation runs on Apache Spark [85] upon MongoDB [91] data repository supporting parallel and scalable processing and analytics tasks.

The current implementation of PA-BOR includes different software components: (i) software-based gateways, (ii) the datastore layer, (iii) all the analytics algorithms discussed in Section 3.5.2.

The developed software-based gateways work also as *Device Connectors* and push real data into PA-BOR exploiting the publish/subscribe approach [80]. Thus, each software-based gateway retrieves from the Service Catalog the end-points for the Resource Catalog and the Message Broker. Next, each gateway registers with the Resource Catalog all the devices and resources it manages, and every 5 minutes it publishes in the Message Broker the data about the status of the gateway box. Thus, an IoT network is emulated where each device sends real data about the status of real heat-exchangers. Real-time algorithms subscribe to Message Broker to receive, process and store the incoming thermal energy data in the historical *datastore*.

The datastore has been designed and implemented in a cluster running MongoDB 2.6.7. All experiments have been performed on our cluster, which has 8 worker nodes, and runs Spark 1.4.1. The current implementation of all analytics algorithms in PA-BOR is a project developed in Python, exploiting the Apache Spark framework. For the results reported in this study, the PA-BOR engine has been configured as follows. We consider thermal power levels related to the 5 months between 1 November 2014 and 31 March 2015, in the time frame from 5:00 am to 11:00 pm. For status detection in SOD, the *sliding window* size has been set to 30 samples and the *transition threshold* to 20 minutes, while in PD one week is the default value of *sliding window* size to estimate the peak instants. To configure the LR-SGD in MLlib, we set the learning rate $\alpha = 1.0$, and 100 total iterations of gradient descent (*stepSize* and *numIterations* in Spark). We used two values for the training window size *trWdw* = 7 and 14 days, which determines the overall training set which is entirely used at each iteration (*miniBatchFraction* = 1.0 in Spark). The sensitivity of prediction error with respect to this and other parameters is described in section 3.5.3. No initial values are provided for the weights vector W of the weights of the hypothesis function $y = h(\mathrm{x})$.

Power prediction results

The values reported in Table 3.1 represent the average prediction errors for the 12 analyzed buildings. In particular, the MAPE and SMAPE values refer to the power prediction performed using the PP algorithm described in Section 3.5.2. The average prediction errors are reported for each building and for each heating cycle of the day. Moreover, for each building, the overall MAPE and SMAPE values are reported, which include all predictions for both the transient and the steady state phases.

The reported values suggest an overall higher precision for predictions made on buildings with a single-cycle, since both overall MAPE and SMAPE increase with the number of heating cycles (even though some double-cycle buildings have lower error values than single-cycle buildings and some others have higher error values than triple-cycle buildings). This overall trend can be motivated by two mutually dependent reasons: (i) more heating cycles mean more (even if shorter) transient states, with higher prediction errors influencing the average values; (ii) more heating cycles mean also more separated steady states (rather than a continuous one) with different behaviors of the same heating system, also with similar weather conditions, depending on the period of the day.

The plots in Figures 3.9-3.10 show the comparison between the real and predicted power values of single buildings, during a single day, plotted as the average values over intervals of 15 minutes. The plot in Figure 3.9 refers to a single-cycle building and the power values are forecast with a prediction horizon of 1 hour. Even though the peak is predicted with a 15-minute delay, its value is very near to the real one, while the prediction of the overall trend of the transient phase is similar to the real one, even though some points are sensibly different. The error in the steady phase is constantly low and close to zero in some points. This high level of precision is favored by the regular trend of the single steady phase in single-cycle buildings, both in a single day and from one day to another.

The plot in Figure 3.10 refers to a triple-cycle building and the power values are forecast with a prediction horizon of 1 hour. In this case, except for the first cycle, the

Table 3.1: MAPE and SMAPE values of PP algorithm applied to the 12 test buildings

Heating cycles	Building ID	Overall		First cycle		Second cycle		Third cycle	
		MAPE	SMAPE	MAPE	SMAPE	MAPE	SMAPE	MAPE	SMAPE
Single	1	15.56	6.78	15.56	6.78	-	-	-	-
	2	18.58	7.95	18.58	7.95	-	-	-	-
	3	20.48	8.35	20.48	8.35	-	-	-	-
	4	22.38	9.32	22.38	9.32	-	-	-	-
	5	20.42	8.46	20.42	8.46	-	-	-	-
Double	6	23.24	9.62	28.81	10.95	20.58	8.06	-	-
	7	22.02	9.56	36.98	13.35	15.52	7.10	-	-
Triple	8	23.11	9.72	35.35	13.90	17.38	7.67	18.33	7.63
	9	27.96	10.62	28.46	10.90	24.73	10.14	25.87	10.85
	10	33.75	11.64	39.70	14.40	38.44	14.49	26.53	10.21
	11	29.05	11.83	31.89	11.98	37.53	13.99	23.23	9.58
	12	27.26	11.56	32.62	13.26	28.39	11.42	23.01	9.27

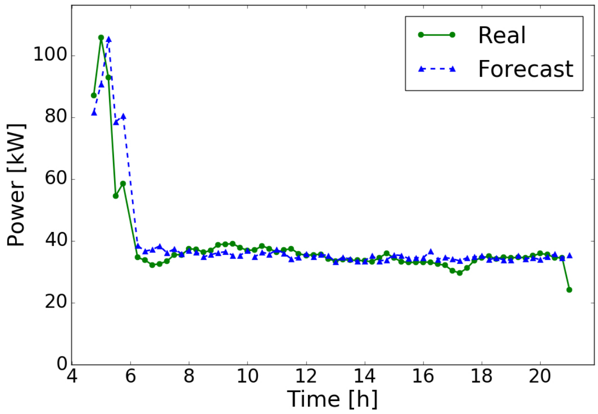


Figure 3.9: Daily 15 minutes average power prediction for a single-cycle building with 1 hour advance (5% maximum error on weather forecast)

trends of the predicted transient phases are very similar to the real ones and in the third cycle the predicted peak value is very near to the real one. The error in the steady phase is higher than in Figure 3.9, but still acceptable.

The plots in Figure 3.11 represent the cumulative frequency of *Absolute Percentage Error* (APE) and of *Symmetric Absolute Percentage Error* (SAPE) of predictions for a single-cycle building during steady and transient states. These two metrics are the terms of the sums in the MAPE and SMAPE formulas respectively (see Section 3.5.2) and represent two measures of percentage error for single predictions. Over 90% of the predictions have a APE lower than 17% in the steady state and lower than 30% in the transient state. For the same percentile, SAPE is less than 8.6% in the steady state but about 33.7% in the transient state. However, in the same state a SAPE of just 15% is the 70th percentile. Therefore, roughly 90% of samples are predicted with a limited error, especially in the steady state. The steep initial growth of the two graphs in Figure 3.11

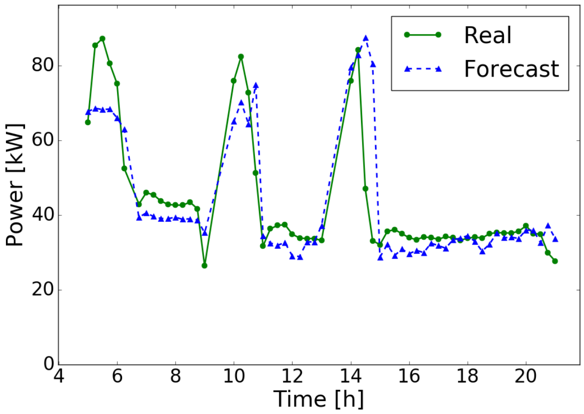


Figure 3.10: Daily 15 minutes average power prediction for a triple-cycles building with 1 hour advance (5% maximum error on weather forecast)

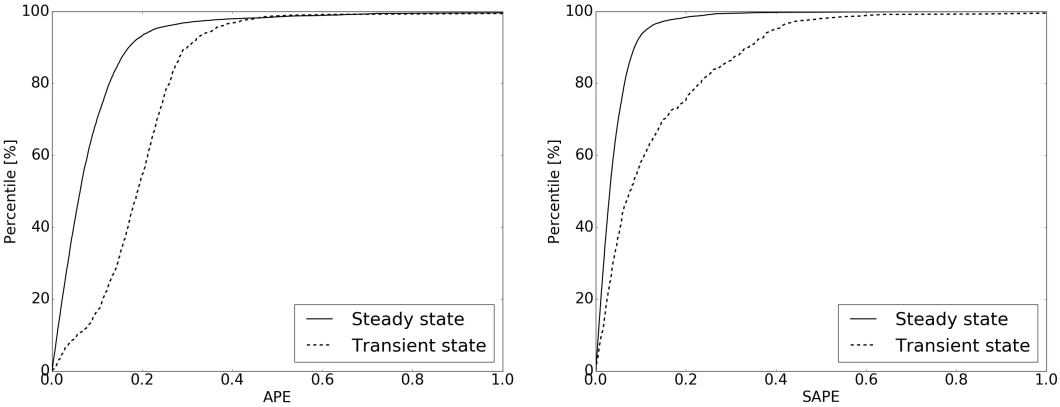


Figure 3.11: Percentile distribution of APE and SAPE over the whole season for a single-cycle building

shows that only a very small number of predictions have high error values. Indeed, over 98% of the predictions have APE and SAPE lower than 50%, in both steady and transient states, while, among the remaining 2%, APE can have very high values (while SAPE ≤ 100% by definition). This suggests how few bad predictions can affect the overall MAPE and SMAPE values and explains why median error values are always lower than the corresponding means.

Sensitivity analysis

Here we analyze the robustness of the Power Prediction (PP) algorithm to the variation of its parameters. For each parameter (i.e., *training window size*, *slot duration*, *prediction horizon*, and *weather maximum error* described below), a set of experiments were run to find, when possible, a good input parameters setting. The *training window size* (*trWdw*) was set to 7 and 14 days; The *slot duration* (*slDur*) was set to 15, 30 and 60 minutes; For each value, the daily timeline is split in fixed time slots, hence with a granularity of 15 minutes the slots start at 00:00, 00:15, 00:30, and so on. A similar partitioning is done for granularities of 30 (00:00, 00:30, etc.) and 60 minutes (00:00, 01:00, etc.). Finally, even if (near-)real-time predictions are based on forecasts of weather data, validation was performed with real measures of past weather data. Therefore, to take into account the prediction error, a random percentage value was added to such measures. The percentage error was modeled as a uniform random variable W with a support defined by the *weather maximum error* (*weErr*) parameter, i.e., $W \sim U[-weErr, +weErr]$. The value of *weErr* was set to 0%, 5% and 10%. Finally, the *prediction horizon* (*prHor*) has been set to 1, 2, 4, 8 and 24 hours and analyzed in combination with the other parameters. These five values were chosen to consider not only short-term, but also medium-term predictions, which even with lower precision values can still be of interest for some end users.

Tables 3.2 to 3.4 show how percentage errors, i.e. mean (MAPE) and median values, vary with respect to the aforementioned parameters.

Table 3.2 highlights the variation between the two different values of *training window size* (which determines the amount of training data). A wider training window (14-days) corresponds to lower error values, in both transient and steady states. Indeed, the prediction algorithm learns from a larger training set and can fit overall a more accurate hyper-plane. Wider training window sizes (e.g. 30 days) have been tested too, but they are not reported in Table 3.2 because no significant improvement has been noticed. The difference with the 7-day window is reduced for shorter *prediction horizons* and becomes negligible for short term predictions (only 0.27% the overall MAPE for *prHor*=1), with a trend reversal in the steady state, where the lowest values of mean and median errors are registered with the 7-day window. This means that a stricter training window can be preferable for predictions over a shorter horizon (1 hour or less) to make the algorithm fit the most recent samples better. Moreover, such a short training window prevents, or at least mitigates, the impact of seasonality on prediction accuracy when the target of prediction is in the transition between two seasons. Hence, we selected 7 days as the default value for *trWdw*.

Table 3.3 reports the variation of prediction errors with respect to the *slots duration*. Overall, the prediction error for *slDur*=60 is always substantially higher than for the other two values (between 0.68% and 1.37%). The lowest values of prediction error for all the *prediction horizons* are obtained with *slDur*=30 instead. This is true both in the steady state and for the overall errors. The transient state exhibits a higher variability

Table 3.2: Sensitivity analysis on *training window size*

prHor	*trWdw*	Overall error (%)			Transient error (%)			Steady error (%)		
(hours)	(days)	mean	median	std dev	mean	median	std dev	mean	median	std dev
1	7	10.76	6.58	22.52	24.05	19.48	34.21	9.24	5.96	20.21
	14	10.49	6.96	20.37	19.80	19.05	22.38	9.42	6.30	19.84
2	7	11.38	6.81	27.15	23.52	18.44	37.43	9.99	6.17	25.34
	14	10.84	7.10	23.74	19.75	18.29	30.14	9.82	6.44	22.67
4	7	12.28	7.13	31.31	23.53	18.34	38.43	10.99	6.44	30.12
	14	11.31	7.29	26.81	19.64	18.17	31.09	10.36	6.63	26.10
8	7	13.43	7.57	35.70	23.53	18.34	38.43	12.27	6.85	35.19
	14	11.98	7.53	29.92	19.64	18.17	31.09	11.10	6.84	29.66
24	7	14.76	8.13	34.32	24.00	18.81	36.68	13.70	7.39	33.88
	14	12.90	7.85	36.90	20.25	18.69	32.13	12.06	7.14	37.31

prHor: prediction horizon in hours
trWdw: training window size in days

and no particular trend can be detected. Hence, we selected 30 minutes as the default value for *slDur*.

Table 3.3: Sensitivity analysis on *slots duration*

prHor	*slDur*	Overall error (%)			Transient error (%)			Steady error (%)		
(hours)	(min)	mean	median	std dev	mean	median	std dev	mean	median	std dev
1	15	10.45	6.64	21.94	22.99	19.30	32.34	9.25	6.06	20.27
	30	10.46	6.77	20.23	21.47	19.44	28.31	9.12	6.15	18.57
	60	11.54	7.33	21.98	20.31	18.92	21.41	10.03	6.33	21.72
2	15	10.93	6.83	25.68	22.62	18.42	38.28	9.81	6.27	23.83
	30	10.86	6.94	23.50	20.54	18.08	32.49	9.68	6.31	21.87
	60	12.23	7.47	28.26	21.08	18.78	25.58	10.71	6.47	28.42
4	15	11.70	7.11	29.84	22.62	18.42	38.28	10.66	6.51	28.69
	30	11.44	7.17	26.07	20.54	18.08	32.49	10.33	6.50	24.95
	60	12.79	7.74	31.93	20.88	18.21	30.87	11.40	6.76	31.90
8	15	12.71	7.44	34.63	22.62	18.42	38.28	11.76	6.82	34.11
	30	12.33	7.50	30.10	20.54	18.08	32.49	11.34	6.80	29.65
	60	13.39	8.03	31.88	20.88	18.21	30.87	12.10	7.06	31.88
24	15	13.74	7.82	36.47	22.41	18.70	34.38	12.91	7.19	36.56
	30	13.67	8.02	35.57	21.66	18.51	32.82	12.70	7.28	35.77
	60	14.44	8.56	32.75	22.17	19.02	37.05	13.11	7.51	31.76

prHor: prediction horizon in hours
slDur: slot duration in minutes

Table 3.4 reports the variation of prediction errors with respect to the *weather maximum error*. In this case, mean error (MAPE) and median error have opposite trends. While MAPE is lower for higher values of *weErr* (especially for longer *prediction horizons*), the median values exhibit more straightforward behavior, as they are lower for lower values of *weErr* with a monotonic trend, i.e. $error(weErr = 0\%) < error(weErr =$

5%) $< error(weErr = 10\%)$. In this case, a wise setting is to use higher values of *weErr* for longer prediction horizons.

Table 3.4: Sensitivity analysis on *weather maximum error*

prHor	*weErr*	Overall error (%)			Transient error (%)			Steady error (%)		
(hours)	(%)	mean	median	std dev	mean	median	std dev	mean	median	std dev
1	0	10.75	6.50	25.95	22.14	19.47	25.36	9.45	5.89	25.70
	5	10.49	6.74	19.55	22.40	19.32	34.31	9.12	6.10	16.52
	10	10.64	7.07	18.09	21.23	18.98	26.45	9.42	6.39	16.43
2	0	11.45	6.71	31.89	21.35	18.57	26.39	10.32	6.07	32.26
	5	10.92	6.91	22.76	22.52	18.49	42.53	9.58	6.27	18.79
	10	10.97	7.24	20.42	21.03	18.06	31.09	9.81	6.60	18.46
4	0	12.30	6.96	35.93	21.18	18.50	26.42	11.28	6.31	36.72
	5	11.61	7.17	27.52	22.36	18.37	43.10	10.38	6.48	24.83
	10	11.49	7.55	22.41	21.22	17.99	33.43	10.38	6.83	20.48
8	0	13.94	7.36	44.77	21.18	18.50	26.42	13.11	6.65	46.34
	5	12.21	7.48	27.09	22.36	18.37	43.10	11.05	6.79	24.33
	10	11.97	7.83	22.85	21.22	17.99	33.43	10.91	7.09	21.04
24	0	15.04	7.76	47.04	21.87	18.97	28.66	14.25	7.06	48.65
	5	13.49	7.91	31.32	23.13	18.84	43.26	12.39	7.18	29.44
	10	12.98	8.26	25.03	21.37	18.30	29.70	12.02	7.55	24.25

prHor: prediction horizon in hours
weErr: weather maximum percentage error

3.5.4 Computing performance of algorithm

The models for SOD and PD algorithms are built once a day, using the data of the past days, and their execution times are lower than 1s. Therefore, their contribution to the overall execution time can be considered negligible. On the other hand, the PP algorithm updates its model every time a new measure is received, still keeping all the historical data for regression analysis, which requires much more computational effort. Therefore, this section focuses only on the PP algorithm to discuss the computational scalability of PA-BOR. The PP algorithm has been executed by evenly distributing buildings data across computing nodes, so that each building is associated with just one single node (we suppose $N_{buildings} > N_{nodes}$). This way, since power prediction for a generic building requires nothing but the data of the same building, which are all stored into a single node, computing nodes need no further information from other nodes and they can process data independently from each other without any communication overhead. Hence, the overall execution time of the prediction algorithm, for each time slot, is (inversely) dependent only from the number of nodes.

Due to the (near-)real-time nature of the predictions, it is interesting to understand how many buildings a single node can handle, yet delivering results in time at every time slot. Therefore, for a single node, we evaluated how the execution time varies with respect to the number of buildings, considering that, for each building, the algorithm provides 5 different power predictions every time slot (one for each *prediction horizon*). Since 2 regression models are used per cycle (one for steady and one for transient state), for a triple cycle building (with 6 models overall), when each prediction refers to a different state, 5 regression models must be trained in the current slot. Similarly, double and single cycle buildings may need, respectively, up to 4 and 2 different models. As a result, the execution time depends also on the type of the building, hence the number of buildings for each cycle type must be considered: N_{1C} for single-cycle buildings, N_{2C} for double-cycle buildings, N_{3C} for triple-cycle buildings.

We estimated the average time to elaborate a single prediction for both steady and transient states. We performed measurements for a variable and increasing number of buildings per node (1 to 128). The computing nodes are 2.67 GHz six-core Intel® Xeon® X5650 machines with 32 GB of main memory running Ubuntu 12.04 server with the 3.5.0-23 kernel. The results highlight a clear linear dependency of execution times with respect to the number of buildings per node (see Figure 3.12). The slope of the linear regression equation among the average execution times is used as the *mean execution time* per single regression model: $t_{ST} = 2.833s$ for steady state and $t_{TR} = 3.187s$ for transient state. The maximum execution times for the three types of buildings are:

$$\begin{aligned}
&t_{1C} = t_{ST} + t_{TR} = 6.02s \\
&t_{2C} = 2 \times t_{ST} + 2 \times t_{TR} = 12.04s \\
&t_{3C} = 2 \times t_{ST} + 3 \times t_{TR} = 15.227s
\end{aligned}$$

To estimate the maximum number of buildings that can be handled in real time with a single node, we suppose predictions start at the beginning of each time slot, using all the data samples received in the previous slots (without the samples of the current slot). If this is acceptable, all predictions have to be performed within *slDur*, i.e., the following condition must be satisfied:

$$\sum_{i=1}^{3} (N_{iC} \times t_{iC}) \leq slDur \tag{3.7}$$

Conversely, when the number of buildings is fixed, it is interesting to estimate the minimum number of required nodes and how long in advance the algorithm should start running in order to deliver results in time. In our scenario, a total amount of 300 buildings were considered, with ($N_{1C} = 69, N_{2C} = 36, N_{3C} = 195$) and an overall execution time of roughly 63 *min* 38 *s*. If *slDur* = 15 *min*, at least 5 computing nodes are required to guarantee that predictions are always computed by the end of the time slot (still assuming an inverse linear dependence between the execution time and the number of computing nodes). On each node, predictions must start roughly 12 *min* 44 *s* before the end of the time slot, with an overall node utilization of around 85%.

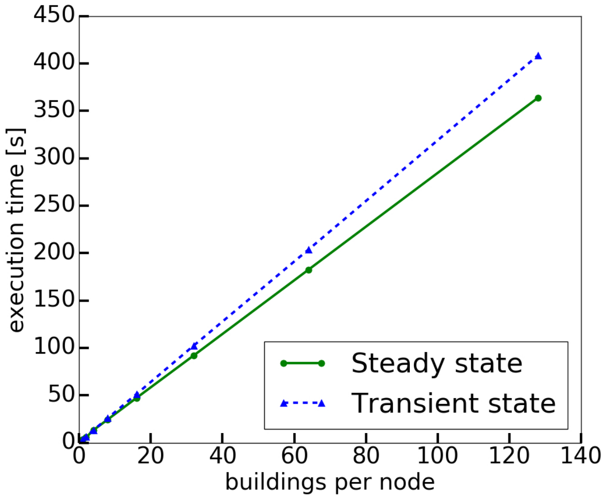


Figure 3.12: Execution time of the Power Prediction algorithm with respect to the number of buildings per node

3.6 Discussion

This section discusses the experimental results presented in this chapter, addressing the usefulness of the results; the most significant weather elements affecting energy consumption; the overall prediction precision and the robustness of PA-BOR; the computing performance of DA-BOR and PA-BOR algorithms.

Exploitation of the mined knowledge. The DA-BOR platform is suitable to characterize energy consumption of buildings through the computation of KPIs from real energy-related data. The use of different *training windows* and *time slots* makes KPIs relevant for various patterns of energy use and helps in modeling multiple behaviors, such as those during specific seasons, morning hours, work hours, etc. Moreover, the aggregation of buildings in neighborhoods allows the comparison between different areas of the city. The distributed computation of *total heat loss coefficient* K_{tot}, based on Energy Signature analysis, enables an advanced characterization of buildings energy performances, which takes into account the variability of climate conditions. K_{tot} is a concise parameter that facilitates the comparison of energy efficiency for a same building across time windows and between many buildings in different cities.

Concerning predictive analytics, PA-BOR can provide reliable forecasts of power demand values of the heating system of buildings in a HDN, with a maximum time granularity of 15 minutes and with a prediction horizon of up to 24 hours. This knowledge can be exploited by many stakeholders of the energy domain, to support their decision-making processes.

From a business perspective, the algorithms can be used by energy managers to estimate the overall energy demand of the day after. The higher time granularity of predictions, performed building by building, make possible also other predictions. Knowing in advance the power exchanged by each heating system, energy managers can devise proper strategies to satisfy their energy demand during the entire day. Furthermore, they can address a more accurate sizing of the HDN for each city district, providing a more reliable service.

The peak power demand occurs always during some specific time slots for most buildings connected to the network. Therefore, knowing in advance these measures for all buildings allows energy analysts to better estimate the overall peak value and to adopt suitable countermeasures to avoid the interruption of the heating distribution. Finally, PA-BOR can be used also by city administrators to better manage public buildings. For instance, the start of the heating cycle can be shifted (anticipated or postponed) to re-balance the peak demand of the network.

Impact of climate conditions. An important information that comes out from experiments is the influence of each weather element on energy consumption. This can be deduced from the regression models, by analyzing the coefficients (weights) of the linear equation that correlates the values of energy consumption and power demand with the (normalized) values of weather variables. The higher the weight, the more the weather variable affects the power value.

The weather variable with the highest weight (absolute value) is *external temperature* (-0.780). The minus sign means an inverse correlation, as the power needed to heat a building is higher at lower external temperatures. *Atmospheric pressure*, which is inversely correlated with temperature, has just more than half of its weight (0.437). The sign is positive because pressure increases when temperature decreases and power demand increases consequently. Other variables have very low weights. *Humidity* has less than 1/10 the weight of temperature (-0.075) and it negatively affects the value of power demand. *Precipitation rate* (0.059), *total precipitations* (0.050), *wind gust* (-0-040), and *wind speed* (-0.025) have negligible weights.

The lower impact of other elements, compared with *external temperature*, is mainly due to their poor impact over the *indoor temperature*, which is the real target parameter of a building heating system. While indoor temperature is constantly influenced by the external one, especially for inefficient buildings (e.g., with high transmittance of walls), elements like precipitations and wind have an impact on the outdoor environment but they can barely affect the indoor conditions.

Accuracy of power prediction. Overall, results in Section 3.5.3 demonstrate the effectiveness of the proposed approach to predict the power levels with a limited error (9.62% is the average SMAPE for all buildings). Roughly 90% of samples are predicted with a limited error, especially in the steady state, and only a very small number of predictions have high error values. This suggests how few bad predictions can affect the overall error and explains why median error values are always lower than the corresponding means.

The PP algorithm has a higher precision with single-cycle buildings and, for Triple Cycle Buildings, it works better in the third cycle. This trend can be motivated by two mutually dependent reasons: (i) more heating cycles imply more transient states, with higher prediction errors influencing the average values; (ii) more heating cycles also lead to more separated steady states (rather than a continuous one) where the heating system can have different behaviors, even with similar weather conditions, depending on the period of the day.

Plots in Figures 3.9-3.10 show how the PP algorithm is capable to reproduce the whole daily power profile, especially in the steady state. The estimated power profile in the transient state is accurate as well, also thanks to the peak power value estimated by the PD algorithm.

Relative errors are very low for most predictions (e.g., APE $<$ 15% and SAPE $<$ 8% are satisfied by 90% of predictions). The error increases for predictions during the transient state, though we are only interested in finding the peak value there. Indeed, in such state, the peak value represents the main critical issue, because it could severely affect the effectiveness of heating network.

Parameters settings and robustness to external variables. The sensitivity analysis in Section 3.5.3 allows to test the robustness of the PP algorithm to the variation of its parameters.

Results about *training window size* demonstrate that it's not necessary to use the whole historical dataset to train the algorithm, but a training window including data of the last 14 days is enough to obtain an accurate regression model. Moreover, for steady state, a training window of just 7 days guarantees accurate predictions as well. The difference between the two window sizes becomes negligible for short term predictions. Therefore, a narrow training window can be used for predictions over a short horizon (1 hour or less) to make the algorithm fit better the most recent samples and to elaborate few data in a short time.

Concerning the *slots duration*, it is worth to notice that predictions for large slots (60 minutes) are less precise than those for small slots (15 minutes) during steady state, but they are more precise during transient state. In both states, the best performance is reached in the middle (30 minutes), while high values of slot duration produce an excessive approximation of the original data (expressed with a granularity of 5 minutes).

Sensitivity analysis on *weather maximum error* tests the robustness of the algorithm with respect to the errors of weather predictions. It is not a controllable parameter, but an external variable that can be computed only in the aftermath. This issue is even more significant when longer prediction horizons are considered, as they can be characterized by higher errors of weather forecasts. The tests show that regression models built with weather data affected by slight errors provide similar predictions of power exchange. Indeed, a relative error of 10% for weather variables produces power predictions with a MAPE even lower than those obtained with $weErr = 0\%$. Such tests confirm the robustness of the algorithm to slight errors in the forecast of weather variables.

Computing performance and scalability. Tests described in Section 3.4.4 show that the computation of KPIs in DA-BOR, using the MuSTLE framework, scales roughly linearly with the number of nodes. The algorithm speed-up approximately corresponds to the number of cluster nodes. From a design perspective, the almost optimal speedup can be tracked back to the choice of the sharding key: it keeps data locality among map and reduce iterations, since most of the energy signature computation is on a per building basis and only district and city averages involve different nodes.

Also PA-BOR can distribute computational load across parallel executors, as it runs on Apache Spark. Tests performed with hundreds of buildings (Section 3.5.4) prove the linear computational scalability of PA-BOR and, in particular, of the PP algorithm. If we double the number of buildings, it is sufficient to double the computational nodes as well. Thus, provided that additional nodes are available, PA-BOR can be used also to analyze data of all the buildings of a big city, still being capable to return results in time.

Chapter 4

Energy demand modeling for buildings asset rating

One of the main factors affecting the energy consumption of buildings is their intrinsic inefficiency due, for instance, to the low thermal insulation of their envelope and to other bad design principles. The positive and negative effects of many building properties on energy efficiency are already known. However, an accurate quantification of the energy consumption that can be obtained by improving some significant features can represent a really profitable information for buildings design and refurbishment.

Therefore, while the research activities presented in Chapter 3 were focused on measured energy consumption, this chapter describes the *Heating Energy Demand Estimation for Building Asset Rating* (HEDEBAR) methodology, designed and developed during the PhD study to explore *building features*. The purpose is twofold: (i) discover the main features that affect the *building energy demand* and (ii) predict the energy demand of new buildings with a reduced set of relevant features (asset rating). Specifically, the activity has been conducted through the analysis of data from Energy Performance Certificates (EPCs). EPC is considered as a major benchmark by regulatory authorities worldwide, which want to foster the improvement of buildings energy efficiency through the adoption of new construction techniques and energy systems.

This activity is complementary to that presented in Chapter 3, because it combines descriptive and predictive analytics techniques but for *asset rating*. Therefore, the two activities together make possible a complete description of buildings energy efficiency, by considering building features that affect real consumption.

EPC includes several features of a building and of its energy systems, one or more numeric parameters indicating its energy demand and, sometimes, a label to assign an efficiency score/class to the building. The analysis of data from EPC can be useful to estimate, during the early design phase, how different features affect the building energy efficiency [92]. For existing buildings this knowledge would be useful to quickly evaluate the suitability of a refurbishment plan. In the proposed methodology, to assess the building thermal energy efficiency, the total *Primary Energy Demand for space heating*

PED_h is considered, being a parameter commonly used in EPCs [93].

The HEDEBAR methodology is based on a two-layers approach. It is structured into two sequential phases, named *Segment estimation* and *Local energy demand prediction*.

The *Segment estimation* phase, identifies the expected (discrete) *segment of energy demand* of the building among *low-*, *high-*, and *very high-demand*. This task has been modeled as a *classification problem*. A classifier is used to assign each building to the corresponding segment of energy demand based on the building features.

The *Local energy demand prediction* phase predicts the (continuous) numeric value of PED_h for the building, based on its features. This second task is formalized as a *regression problem*. A different regression model is created for each segment to locally predict the PED_h value.

The two-layers approach of HEDEBAR allows us to maximize the accuracy in the prediction of the PED_h value for a building. In fact, a different prediction model is used based on the expected segment of energy demand of the building, rather than a single prediction model for all buildings regardless of their energy demand.

For the creation of the classification and regression models, respectively in the first and second layers, a comparative study has been conducted among four different machine learning algorithms, which demonstrated good prediction performances in several contexts, as *Artificial Neural Networks* (ANN), *Reduced Error Pruning Tree* (REPT), *Random Forest* (RF), and *Support Vector Machines* (SVM).

During the PhD activity, the HEDEBAR methodology has been validated on a data set of real energy certificates of almost 90 thousands buildings in the Piedmont region of Italy [94, 95]. Experimental results show that HEDEBAR is an effective methodology for buildings asset rating, with both descriptive and predictive purposes. The proposed methodology can estimate the PED_h value for a building with an acceptable error. The information extracted can be used by domain experts, public authorities and regulatory bodies to plan future energy policies that leverage on specific building features.

Despite the data set contains data for a very large number of buildings, its volume is still not comparable with the typical values of Big Data. Therefore, the current implementation of the HEDEBAR methodology is not based on an cloud/cluster computing architecture, neither it needs to employ the MuSTLE framework for data aggregation and analysis. However, the methodology is general enough to be used also with other EPCs issued with other certification systems. The provided results can be used, during the design of buildings, to focus on the features that mostly affect the energy demand for each class/segment and to quantify the improvements that can be obtained by varying their values.

This chapter is organized as follows. The building features of the analyzed data set are described in Section 4.1. Section 4.2 presents the related research work on the analysis of data from EPCs. Section 4.3 describes all the steps of the HEDEBAR methodology. Section 4.4 presents the experimental results, which are discussed in depth in Section 4.5.

4.1 Building characterization through Energy Performance Certificates (EPCs)

The Energy Performance Certificate (EPC) describes the different features of the building affecting its energy performance as well as the variables used to quantify the building energy consumption. The following four main categories of features can be identified: (i) *building geometric features*, (ii) *physical features of building envelope*, (iii) *building historical information*, and (iv) *energy related variables*. Each category is briefly described below, while Table 4.1 reports some examples of relevant attributes for each category.

Building geometric features. The attributes in this category describe the different geometric features of the building, which have an impact on the building energy performance. The category includes attributes such as average ceiling height, heat transfer surface and volume of the building.

Physical features of building envelope. The attributes in this category are related to the physical properties of the building envelope, which impact on the capacity of the building to retain heat inside its environments. Example attributes are the thermal transmittance values of the opaque and transparent building envelope.

Building historical info. This category includes attributes like the building construction year, last refurbishment year (if any), and other specific operations on the heating system or sub-systems, which can have a direct or indirect impact on the building energy efficiency.

Energy related variables. This category includes the features of the energy (sub-)systems and the amounts of energy consumption. The former features refer to the space heating system and its subsystems (generation, control, distribution, emission). The latter features are described by means of a set of variables. Among them, the *Primary Energy Demand for space heating* (PED_h) is related to the energy consumption of the building.

For its important role, the PED_h value has been selected as the target variable for analysis and prediction in this study. PED_h is an energy related variable defined for benchmarking purposes. It is an estimation of the amount of real energy consumption of a building in standard use conditions and it is used to assign an energy class label to the same building. The PED_h value is estimated starting from the features included in the energy certificate, which can be used to compare different buildings. The PED_h value usually refers to a period of one year and it is normalized by the building floor area. It contributes to the evaluation of the overall Primary Energy Demand of buildings (PED) together with the Primary Energy Demand for domestic hot water (PED_w).

The HEDEBAR methodology has been validated on a real data collection of EPCs for buildings located in the Piedmont region, North Western of Italy, related to 2013. The dataset includes approximately 90,000 energy certificates, each one characterized by 62 features, included those described above and the target variable PED_h. Analyzed

Table 4.1: Starting list of attributes selected to characterize the building heating energy demand with the proposed HEDEBAR methodology.

Category	Name	Symbol	Unit	Range
		Explanatory variables		
Geometry	**Floor area**	A	m^2	$\mathbb{R}^+$
	Heat transfer surface	S	m^2	$\mathbb{R}^+$
	Average ceiling height	H	m	$\mathbb{R}^+$
	Gross Heated Volume	V	m^3	$\mathbb{R}^+$
	Aspect ratio	R	m^{-1}	$\mathbb{R}^+$
Envelope	**Average U-value of vertical opaque envelope**	U_o	$W/(m^2 \cdot K)$	$\mathbb{R}^+$
	Average U-value of the windows	U_w	$W/(m^2 \cdot K)$	$\mathbb{R}^+$
	Quality of building envelope	q_{env}	-	$\{1,2,3,4,5\} \subset \mathbb{N}$
History	**Construction year**	y_c	a	$\mathbb{N}$
	System installation year	y_{sys}	a	$\mathbb{N}$
	Heating generator installation year	y_{gen}	a	$\mathbb{N}$
	Last refurbishment year	y_{ref}	a	$\mathbb{N}$
Energy	**Average global efficiency for space heating**	η_h	-	$[0,1] \subset \mathbb{R}$
	Renewable Energy quota	ρ_{ren}	-	$[0,1] \subset \mathbb{R}$
	Installed Heating Power	P_h	kW	$\mathbb{R}^+$
		Target variable		
Energy	**Normalized primary energy demand for space heating**	PED_h	kWh/m^2	$\mathbb{R}^+$

buildings, both detached houses and flats in condos, are distributed across the Piedmont region in 25 different cities. EPCs were issued in the first six months in 2013.

4.2 Related work

Within the scientific context, several research activities have been performed on buildings energy performance assessment, for: (i) *prediction of energy demand* [92, 96] and *energy class* [97], (ii) *rating* and *benchmarking* [98, 99, 100], (iii) individuation of representative buildings [101, 102], (iv) improvement of existing methods [97, 103], and (v) comparative analysis of new models based on data mining algorithms, like linear regression analysis, decision trees, ANNs, and clustering.

Several works have proposed a benchmarking of different types of buildings. Dall'O' et al. [98] analyse a real dataset of energy certificates to assess the energy performance, to detect anomalies in the registered certificates and to quantify the energy retrofit potential in existing buildings. Chung, Hui, and Lam [99] developed a benchmarking process for energy efficiency of commercial buildings by means of Multiple Regression Analysis (MRA). Gao and Malkawi [101] use clustering to classify buildings according to multiple features, like physical properties, environmental conditions, occupancy.

Lara et al. [102] adopt the cluster analysis to find out a few samples representative of about 60 buildings, in order to optimize the energy retrofit measures. Hong et al. [100] use an approach based on case-based reasoning, MRA, ANN and GA, to produce a methodology for operational rating with higher explanatory power and higher prediction accuracy at the same time. Tso and Yau [104] compared the accuracy of linear regression, ANN, and decision tree in predicting average weekly electricity consumption for both summer and winter in Hong Kong. Koo et al. [92] use the finite element method to estimate the heating and cooling demand of buildings, using data about building envelope design. In [96] a decision tree is used to model the real consumption of residential buildings in order to predict the energy use of newly designed buildings. Melo et al. [97] use ANN to improve the accuracy of surrogate models for labelling purposes, based on simulations results. Authors in [103] tackle the problem of uniformity of criteria among different certificates, therefore they use ANNs to predict the heating energy demand and to validate a dataset of energy certificates.

The study of real data from EPC databases has been performed in several countries. Fabbri, Tronchin, and Tarabusi [105] discuss about the effects of EPBD Directive and Italian EPC system on the real estate market prospective. Hjortling et al. [106] propose a study to define the current energy consumption baseline for buildings in Sweden, using data from 186k energy performance certificates issued for commercial buildings and based on energy bills rather than on theoretical calculations. The paper puts in evidence that real energy consumption is often higher than the one stipulated by the building code. Xiao, Wei, and Jiang [107] proposes a cluster analysis of the energy consumption (EUI excluding District Heating) of office buildings in China, to study its statistical distribution characteristics. It was found that the distribution of energy consumption has quite different characteristics than in Japan and the US. Other analyses of EPCs aimed at defining the current energy consumption baseline of existing buildings in Greece and Spain are presented respectively by Dascalaki et al. [108] and by Gangolells et al. [109].

Contribution of the research activity. The PhD activity described in this chapter brings a significant contribution in the use of data mining techniques for the asset rating of buildings, both in methodological and analytical terms.

From the *methodological* perspective, this research proposes a *two-layer approach* to characterize the energy demand of buildings using multiple *independent models* for different *building segments*. Models for energy demand characterization are generated in both layers using four different data mining algorithms.

From the *analytical* perspective, the proposed approach estimates the building energy demand with acceptable errors, comparable with those of previous works [103], but using a *smaller set of building features*. Indeed, HEDEBAR keeps only the most relevant features affecting energy demand to build the prediction models. Moreover, HEDEBAR has the advantage to produce an *interpretable classification model*, as it employs the Reduced Error Pruning Tree (REPT) algorithm [110]. The model provides useful information about the most relevant building features affecting energy demand.

4.3 HEDEBAR methodology

The HEDEBAR methodology is aimed to model the yearly *Primary Energy Demand for space heating* PED_h of residential buildings as a function of few influencing variables relying on a large data set of energy certificates in Piedmont region. The methodology considers different categories of features and selects those that mostly affect the energy demand. The relative impact of each feature can vary from one building to another according to the energy efficiency. A 2-layer approach has been adopted. The logical components of HEDEBAR are represented in Figure 4.1 and they are briefly described below.

Data collection and preprocessing include all the preliminary tasks necessary to provide the proper datasets to the algorithms that operate in the later phases. *Data collection* takes data from the energy certificates and other contextual information. *Data preprocessing* includes removing records with errors and missing values and enriching energy certificates with contextual information. *Features selection* aims at reducing the dimensions of the dataset, in order to keep only those features that are highly correlated with the heating energy demand. Such features are the *explanatory variables* of the models that will be generated in the following steps.

The *Segment estimation* is the first step of the 2-layer approach. Different classification algorithms have been trained during this step, to learn a classification model that properly assigns buildings to different predefined segments, considering only the explanatory variables. A suitable test dataset has been used to assess the classification performance of each algorithm in order to select the best one. The interpretability of each model depends on the algorithm used to produce it. Therefore, when two or more algorithms have similar prediction performances, the most interpretable one is preferred.

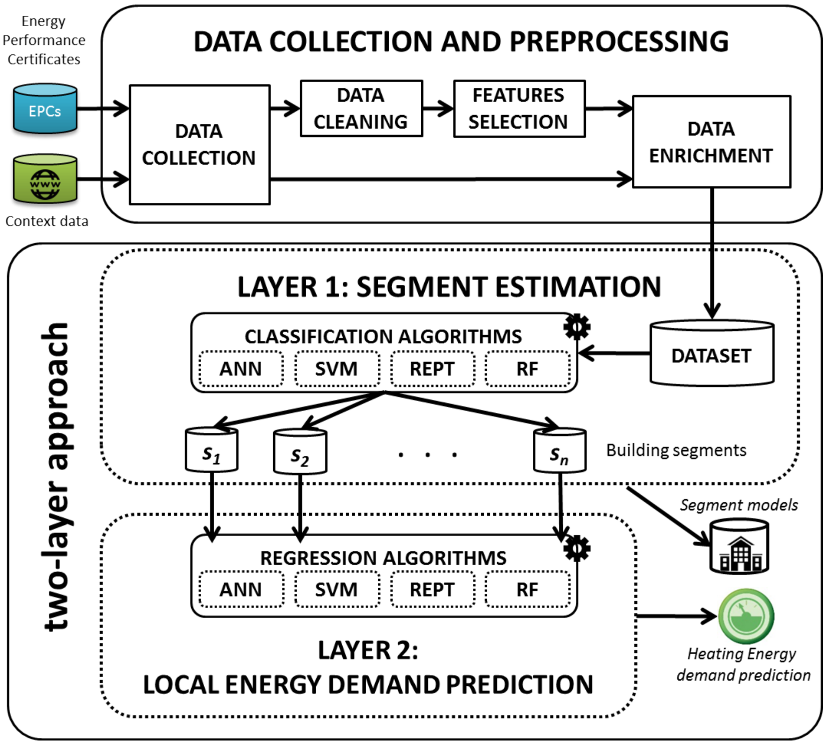


Figure 4.1: The proposed HEDEBAR methodology for automatic asset rating of buildings

The *Local energy demand prediction* is the second step of the 2-layer approach. It uses regression algorithms to learn a regression model for estimating the exact value of heating energy demand considering only the explanatory variables. An independent model for each segment of the first layer has been trained.

The 2-layers methodology provides a twofold output: the *segment model* for the analysed buildings, useful to understand the features with the highest explanatory power with respect to the energy demand and to highlight the differences among the segments; the *heating energy demand prediction* for new buildings.

4.3.1 Data preprocessing

The whole raw dataset derived from EPCs usually includes many building features, represented through variables of different data types such as numeric (integer or real), nominal, textual, and boolean. However, some features could be not relevant or even misleading for the subsequent data analysis phase and their inclusion in the features set would increase the computational cost of the data analysis task. Moreover, datasets derived from energy certificates filled by auditors could contain data errors which can badly affect the quality of the extracted knowledge.

To address the above issues and to improve both effectiveness and efficiency of the data analytics phase, HEDEBAR includes a preprocessing step. This step aims to (i) *clean* the original data collection to remove errors in data and (ii) *select the most relevant features*, thus reducing the data dimensionality and providing an handier and more reliable dataset. Moreover, collected data are (iii) *enriched* with additional *contextual information* to cope with external environmental conditions that could differently affect the estimation of the PED_h value for each building. These three steps are better described below.

Data cleaning. The whole data set is firstly inspected based on the advice of domain experts to remove irrelevant features. Then, the dataset with the remaining features is analysed: a building is discarded when its EPC includes attributes with values outside the allowed ranges, either for physical reasons or because they can be considered as outliers.

Feature selection. After the cleaning process, the feature selection task is conducted on the remaining buildings and attributes using the *minimal-Redundancy Maximal-Relevance* (mRMR) approach [111] through the evaluation of the *Pearson Product-Moment Correlation* (PPMC) coefficient [35]. HEDEBAR analyses the degree of correlation between attributes and the target variable PED_h, with the aim of discarding attributes not relevant or redundant for the estimation of the PED_h value.

Other approaches like Principal Component Analysis (PCA) [112] have been discarded due to the need of keeping the original building features as input of the algorithms, without combining them into new variables, to preserve the interpretability of the generated models. Indeed, PCA extracts linear combinations of the original features, to maximize the variability of the data into fewer new variables. Therefore, PCA introduces new variables, different than building features and with a different meaning, which is harder to be explained in a physical sense. Therefore, even if the use of PCA for feature selection could have provided a better performance of prediction, it would have definitely decreased the interpretability of the models.

The mRMR approach has been successfully adopted in other works to select the most discriminant subset of variables for different classification methods and in different application domains (as for example SVM [113] and ANN [114]). According to the *Maximal-Relevance* principle, in HEDEBAR features strongly correlated to the target variable PED_h are kept, while those with a negligible correlation are discarded. Then, following the *minimal-Redundancy* approach, a further dimensionality reduction is applied on remaining features. Specifically, between two or more features with high mutual correlation, the one with the highest correlation with PED_h is retained while the others are discarded.

The Pearson Product-Moment Correlation (PPMC) coefficient [35] is adopted in HEDEBAR to analyse the pairwise correlation between all the building features and between each building feature and the target variable PED_h. PPMC is a measure of linear dependence not influenced by the unit of measure of the features. PPMC is defined as follows. Let $X = \{x_i, i = 1, ..., n\}$ and $Y = \{y_i, i = 1, ..., n\}$ be two building features, where i value determines a specific building. The PPMC coefficient $\rho_{X,Y}$ is a measure of linear dependence between X and Y, defined through Equation 4.1.

$$\rho_{(X,Y)} = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}}. \tag{4.1}$$

The PPMC value varies in the range [-1,+1], where values near to +1 indicate a high positive correlation, values near to -1 indicate a high negative correlation. The higher the $|\rho_{(X,Y)}|$ value, the stronger the correlation (either negative or positive, according to the sign of the coefficient). $\rho = 0$ means a total lack of correlation among the variables.

Two different correlation thresholds for features selection have been set. Firstly, a correlation threshold ρ_{PED_h} has been used to keep relevant building features, i.e., each feature X with an absolute value of correlation with PED_h such that $|\rho_{(X,PED_h)}| \geq \rho_{PED_h}$. Features with correlation values lower than ρ_{PED_h} have been discarded. Then, the selected features have been further explored to discard the redundant ones. A correlation threshold ρ_{mutual} has been used to identify pairs of mutually correlated features, i.e., each pair of features X and Y with an absolute value of mutual correlation such that $|\rho_{(X,Y)}| \geq \rho_{mutual}$. For each pair of correlated features X and Y, the feature X with the lowest value of correlation $|\rho_{(X,PED_h)}|$ with variable PED_h has been classified as redundant and thus discarded.

Data enrichment. As the last step in the preprocessing phase, data collected on buildings are enriched with additional contextual information acquired from external open data sources. To cope with external environmental conditions that could differently affect the estimation of the PED_h value for each building, PED_h is recalculated in a reference standard climatic condition. Specifically, to normalize PED_h we used the ratio between the Degree Days value of a reference city and the Degree Days value of the city where the building is located. In this way, PED_h is expressed for all buildings as if they were located in the same reference city. Therefore, comparisons among buildings can be done regardless of their location. In the use case considered in this study, the Degree Days in the city of Turin is considered as reference value.

4.3.2 Two-layers approach for the estimation of heating energy demand

The HEDEBAR framework executes the prediction of the PED_h value for each building in a two-levels fashion. HEDEBAR makes use of features from energy certificates as explanatory variables to predict the PED_h value. Each variable has a (different) degree of influence over the energy demand and buildings with similar values of some significant variables should have close values of PED_h. The two-layers approach is based on the hypothesis that the degrees of influence of features over the energy demand vary across different *segments of buildings*. Therefore, given a new building, HEDEBAR first identifies the most suitable model to predict its PED_h, then it actually estimates the PED_h value using that (local) model.

The *two-layers approach* of HEDEBAR is structured into two sequential phases, named *Segment estimation* and *Local energy demand prediction*.

The *Segment estimation* phase, identifies the expected (discrete) segment of energy demand of the building, such as low-, medium-, and high-energy demand. This task has been modeled as a classification problem. A classifier is used to assign each building to the corresponding segment of energy demand based on the building features.

The *Local energy demand prediction* phase predicts the (continuous) numeric value of PED_h for the building, based on its features. This second task is formalized as a regression problem. A different regression model is created in advance for each segment to locally predict the PED_h value.

Thus, a new building (with unknown energy demand) is first classified into a segment through the *segment estimation* phase. Then, the PED_h value of the new building is estimated through the *local energy demand prediction* phase, using the regression model assigned to its segment and trained with the corresponding training subset.

To generate the classification and regression models, the HEDEBAR system can easily integrate most classification and regression algorithms currently available in literature. To select the most appropriate algorithms, two complementary aspects have been considered: (i) the ability of the algorithm to accurately predict the segment and the PED_h value, and (ii) the interpretability of the model it generates.

The algorithms used in the two phases are: *Artificial Neural Network* (ANN), Reduced Error Pruning Tree (REPT), *Random Forest* (RF), and *Support Vector Machine* (SVM) [110]. The choice is motivated by the good performances they provide in several applications. Moreover, REPT algorithm has an interpretable model, which makes possible a better understanding of the relationship between building features and the target variable PED_h.

4.4 Experimental results

In this section we validate the effectiveness and the usability of the proposed HEDEBAR methodology focusing on the following aspects: (i) the ability to correctly estimate the segment of a building; (ii) the ability to accurately predict the PED_h value for each building, (iii) the selection of the classification and regression algorithms integrated in the two layers of the system, (iv) the performance comparison with other approaches, (v) the impact of the system configuration parameters. Additional objectives have been pursued during the analysis, in order to enhance the explanatory capacity of the methodology: (vi) the identification of the most significant attributes and (vii) the explanation of the main variables that determine the membership to a segment.

The overall dataset is split into a training and a test set. The first one is used by the algorithms to actually generate the regression models, using a k-fold cross-validation. Hence, k different validation subsets are extracted in turn from the training set. The purpose of the training phase is minimizing the mean prediction error for the training set. The test set is used to evaluate the capacity of each model to predict the heating energy demand of new buildings.

The open source Rapid Miner v5.3.0 toolkit [115] has been used for correlation analysis and for classification and regression tasks. Due to a limitation of Rapid Miner, the regression task with RF has been implemented in R software [116].

4.4.1 Data dimensionality reduction using HEDEBAR

This section presents a subset of performed experiments to show the ability of HEDEBAR in selecting the subset of features relevant for estimating the PED_h value. We experimentally evaluated that the data analysis steps in HEDEBAR perform better after applying the feature selection than when considering all attributes available in the original dataset.

As described in Section 4.3.1, the feature selection task was carried out using the mRMR principle, while to assess the explanatory power of each attribute we calculated the PPMC coefficient between the attribute and the PED_h. To select relevant features, we set the threshold value $\rho_{PED_h} = 0.1$ for PPMC and we took only the attributes for which $|\rho_{(X,PED_h)}| \geq \rho_{PED_h}$, listed in Table 4.2.

The features selection process allowed us to select ten attributes from those in Table 4.1, while the other eight attributes were discarded. The selected attributes are reported in Table 4.2 and they are briefly described below. Selected attributes include a subset of the original attributes in the *Energy*, *Geometry*, and *History* categories, and all attributes in the *Envelope* category.

Results show a significant positive correlation between five attributes from the *Geometry* and *Envelope* categories and the PED_h value ($\rho_{(X,PED_h)} \geq \rho_{PED_h}$). Due to their positive correlation with PED_h, higher values of these variables lead to higher values of PED_h. More specifically, three attributes are from the *Geometry* category: *heat transfer surface* (S) ($\rho_{(S,PED_h)} = 0.319$), *average ceiling height* (H) ($\rho_{(H,PED_h)} = 0.187$) and *aspect ratio* ($R = S/V$) ($\rho_{(R,PED_h)} = 0.516$). The *heat transfer surface* is the overall surface of the heated environments of the building exposed to outdoor (not heated) environments, while the *average ceiling height* affects the overall volume to be heated. A wider surface towards outdoor as well as an higher volume to be heated increase the amount of heat transfer. The *aspect ratio* ($R = S/V$) is the ratio between the heat transfer surface and the volume. Instead, the two *U-values* attributes are from the *Envelope* category and they indicate thermal transmittance of opaque (U_o) ($\rho_{(U_o,PED_h)} = 0.599$) and transparent ($U_w$) ($\rho_{(U_w,PED_h)} = 0.421$) envelope.

Results point out also a significant negative correlation between five attributes from the *History*, and *Envelope*, *Energy* and *Geometry* categories and PED_h ($\rho_{(X,PED_h)} \leq -\rho_{PED_h}$). Due to their negative correlation with PED_h, lower negative values of these variables lead to higher values of PED_h. More in detail, the attributes are the *quality of building envelope* (q_{env}) ($\rho_{(q_{env},PED_h)} = -0.372$) from the *Envelope* category, the *average global efficiency for space heating* (η_h) ($\rho_{(\eta_h,PED_h)} = -0.315$) from the *Energy* category, and the *floor area* (A) ($\rho_{(A,PED_h)} = -0.282$) and the *gross heated volume* (V) ($\rho_{(V,PED_h)} = -0.241$) from the *Geometry* category. *Average global efficiency for space heating* (η_h) is the average yearly efficiency of the heating system, which considers the performances of its subsystems, i.e., emission, distribution, control, and generation. The *floor area* (A), the *average ceiling height* (H), and the *gross heated volume* (V) are respectively the overall walkable surface, the average distance from floor to ceiling, and the volume of all the heated environments of a building. Therefore, the primary energy demand per square meter is on average higher in smaller buildings with high-quality envelopes and efficient heating systems.

All the three features discarded from Table 4.1 have a PPMC coefficient with PED_h lower than 0.1: for *System installation year* (y_{sys}) $\rho_{(y_{sys},PED_h)} = -0.022$, for *Heating generator installation year* (y_{gen}) $\rho_{(y_{gen},PED_h)} = 0.002$, and for *Installed Heating Power* (P_h) $\rho_{(P_h,PED_h)} = -0.005$.

Redundant features have been detected too, but among those features that were already discarded in the *data cleaning* step. For instance, the *efficiency of emission subsystem* (η_e) is highly correlated with η_h ($\rho_{\eta_e,\eta_h} = 0.538$). However, its correlation with PED_h ($\rho_{(\eta_e,PED_h)} = -0.261$) is lower than the correlation between η_h and PED_h ($\rho_{(\eta_h,PED_h)} = -0.315$), hence, for the principle of *minimal-Redundancy*, η_x would have been discarded anyway.

4.4.2 Characterization of building segments

In this study, three reference *segments of energy demand* have been considered, representing respectively *low energy demand buildings* (segment s_1), *high energy demand buildings* (s_2), and *very high energy demand buildings* (s_3). Dataset splitting into segments has been done according to the reference value range of PED_h specified in [94, 95]. Segment s_2 includes buildings with PED_h values between 0 and $100kWh/m^2$, while buildings in segment s_2 have $100kWh/m^2 \leq PED_h \leq 300kWh/m^2$, and in segment s_3 $PED_h \geq 300kWh/m^2$.

The use case dataset has been partitioned into the three segments according to the values of variable PED_h with the aim of grouping together building with similar performance of energy efficiency.

To analyse the cardinality of each segment, Figure 4.2 plots the distribution of the PED_h values in the dataset. Dashed vertical lines delimit the ranges of each segment. Approximatively, segment s_1 corresponds to the first quartile of PED_h; segment s_2 covers the second and third quartiles, and segment s_3 corresponds to fourth quartile. The three segments result into sets with the following cardinalities. The larger segment is s_2 including medium efficient buildings (39,003 buildings), followed by s_1 with highly efficient buildings (25,930) (s_1), and the s_3 with inefficient buildings (21,176).

Table 4.2: Features selected for the prediction of PED_h. PPMC coefficient values with respect to target variable PED_h are expressed for each input feature.

Category	Name	Symbol	Correlation with PED_h
Envelope	**Average U-value of vertical opaque envelope**	U_o	0.599
	Average U-value of the windows	U_w	0.421
	Quality of building envelope	q_{env}	-0.372
Geometry	**Aspect ratio**	R	0.516
	Heat transfer surface	S	0.319
	Average ceiling height	H	0.187
	Gross Heated Volume	V	-0.241
	Floor area	A	-0.282
Energy	**Average global efficiency for space heating**	η_h	-0.315
History	**Construction year**	y_c	-0.466

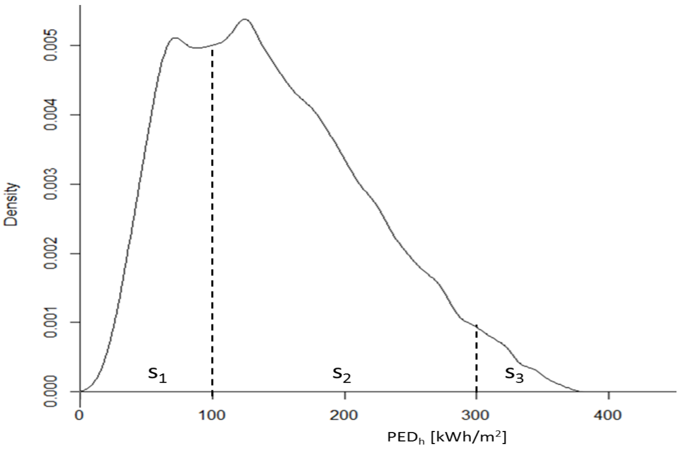


Figure 4.2: Distribution of PED_h variable in the considered dataset. The dashed lines delimit the three segments.

4.4.3 Segment estimation

The classification task aims at assigning each new building into the correct building segment. The classes of the classification task are the three segments presented in Section 4.4.2, identified by the nominal labels s_1, s_2, and s_3. All the four classification algorithms integrated in HEDEBAR (i.e., ANN, REPT, RF and SVM) have been experimentally evaluated for the classification of buildings into segments. The algorithm providing the classification model with the highest accuracy has been selected as reference for this phase.

To validate the results of the classification process four established quality measures [117] have been considered. The overall quality of the classification model is evaluated in terms of *accuracy*. This measure counts the overall number of buildings correctly assigned to their corresponding segment. However, the unbalanced distribution of buildings in the three segments could lead to a biased value of accuracy, as it could be mostly influenced by bigger segments. Therefore, other measures have been also used. For a more accurate evaluation of the classification model, per-class classifier predictions were evaluated according to *precision*, *recall*, and *F1-measure*. $Precision(s_i)$ indicates the percentage of buildings that are correctly revealed as in segment s_i. $Recall(s_i)$ indicates the number of buildings assigned to segment s_i with respect to the total number of buildings actually in s_i. The *F1-measure*(s_i), which is computed as the harmonic average of precision and recall, quantitatively estimates the balancing between $Recall(s_i)$ and $Precision(s_i)$ for segment s_i. In the experiment evaluation, we computed the precision, recall, and F1-measure values for each class label corresponding to each of the three segments.

A good trade-off between recall and precision values - in the assignment of a building to a segment - is needed to properly predict the PED_h values for a new building in the subsequent *Local energy demand prediction* task. On the one side, *high precision values* on most (all) segments are crucial to foster an accurate prediction of the PED_h values in the subsequent regression task. Indeed, the correct classification of a building into the corresponding segment facilitates the subsequent prediction of the PED_h value for the building. In fact, this prediction is performed through a model trained using data of buildings with similar efficiency. A low $Precision(s_i)$ value indicates that many buildings were mistakenly classified into segment s_i. This would result in erroneous predictions of PED_h values in the second step. On the other hand, achieving *high recall values* on most segments is desirable as well. A low $Recall(s_i)$ indicates that few buildings of segment s_i are correctly classified into s_i, and they have been wrongly assigned to a segment other than s_1. This wrong assignment would result into an erroneous predictions of PED_h values due to the selection of a less appropriate prediction model in the second step.

Tables 4.3 and 4.4 report the results achieved by the four classification algorithms integrated into HEDEBAR. Tables show the accuracy on the overall dataset as well as precision, recall, and F1-measure for the three segments.

The RF classifier provides the highest accuracy value (85.67%) followed by REPT (82.03%), ANN (67.51%) and SVM (67.24%). Moreover, RF achieves also the best F1-measure on all segments (88.87%, 84.05%, and 82.76% in segments s_1, s_2 and s_3 respectively). More in detail, RF obtains the highest precision value for all segments (90.52%, 82.65%, and 83.58% for segments s_1, s_2, and s_3 respectively). RF also provides the highest recall values for two segments (87.27% and 85.49% for segments s_1 and s_2 respectively), while the recall obtained on segment s_3 (81.96%) is very close to the value provided by algorithm REPT (82.53%), which is the highest recall value among the four algorithms. Since the RF classifier achieves the highest values for almost all performance parameters, we chose it as reference algorithm for creating the model which classifies a new building into the corresponding segment.

REPT is the second best algorithm for almost all performance parameters, providing accuracy, precision and recall values lower than those of RF, but still more than acceptable. Ad additional key point of REPT is the fact that this algorithm builds an interpretable classification model. This model is a decision tree from which human-readable classification rules can be extracted. Thus, domain experts can use the model not only to automatically classify a building into the corresponding segment but also to analyse the most relevant properties that characterize each segment as well as to understand why a building has been classified into a segment.

The SVM and ANN algorithms provide the worst values for all performance parameters, which are significantly lower than those obtained with RF and REPT algorithms.

Therefore, according with the experimental evaluation, we decided to include two different classification models into the *Segment estimation* layer of the HEDEBAR framework. The RF classifier is used to automatically label a new building with the corresponding segment. Based on the assigned segment, the proper regression model is selected in the subsequent layer (*Local energy demand prediction*) to predict the PED_h value for the building. Instead, the REPT model is used to provide domain experts with a qualitative analysis of the impact of variables characterizing buildings on the primary heating energy demand. This aspect will be discussed in detail in Section 4.4.6.

Table 4.3: Overall classification accuracy of ANN, REPT, RF and SVM algorithms

	ANN	**REPT**	**RF**	**SVM**
Accuracy [%]	67.51	82.03	**85.67**	67.24

4.4.4 Local energy demand prediction

The regression task aims at estimating the value of PED_h using the regression model selected for each building in the *Segment estimation* phase. The ANN, SVM, REPT, and RF algorithms have been experimentally evaluated for this task.

Table 4.4: Overall percentage classification precision, recall and F1-measure of ANN, REPT, RF and SVM algorithms for each building segment.

	ANN	REPT	RF	SVM
		Segment s_1		
Precision [%]	77.71	87.70	**90.52**	82.49
Recall [%]	70.03	83.84	**87.27**	61.97
F1-measure [%]	73.67	85.73	**88.87**	70.77
		Segment s_2		
Precision [%]	62.11	80.40	**82.65**	60.68
Recall [%]	75.54	80.56	**85.49**	81.74
F1-measure [%]	68.17	80.48	**84.05**	69.56
		Segment s_3		
Precision [%]	68.65	78.60	**83.58**	70.62
Recall [%]	49.62	**82.53**	81.96	46.98
F1-measure [%]	57.60	74.93	**82.76**	56.42

Table 4.5 displays the errors of the four algorithms in predicting PED_h for each segment. Three different measures of prediction error, usually adopted in literature to evaluate regression algorithms, have been used [84]: (i) *Mean Absolute Error* (MAE) is the mean of all the absolute values of the errors obtained with the test samples; (ii) *Mean Absolute Percentage Error* (MAPE) expresses the mean absolute error in percentage terms; (iii) *Root Mean Square Error* (RMSE) is the square root of the mean of the square of all the errors obtained with the test samples. While MAE refers only to the mean value of the distribution of absolute errors, RMSE is affected also by the standard deviation of such distribution. Compared to MAE, RMSE amplifies and severely punishes large errors. Table 4.5 shows that the RMSE is always higher than MAE, meaning that, for a few samples, the prediction error of the four algorithms is very high. On the other hand, other test samples are predicted with a high precision.

Best values for each segment are reported in bold in Table 4.5. REPT produces the overall lowest error values for the three measures (MAPE = 16.64%, RMSE = 33.12 kWh/m^2, MAE = 22.21 kWh/m^2) and it has also the best performances in each segment. In relative terms, REPT performs better in segments s_2 and s_3, where MAPE is 14.75%, and 15.90% respectively, while it has a substantially lower performance in segment s_1, where MAPE = 20.25%. The second best algorithm is RF, with an overall MAPE of 16.89%, while SVM and ANN provide higher error values (MAPE = 21.52% and MAPE

= 27.02% respectively). Therefore, REPT has been selected for local energy demand prediction, in order to better characterize groups of buildings with similar features.

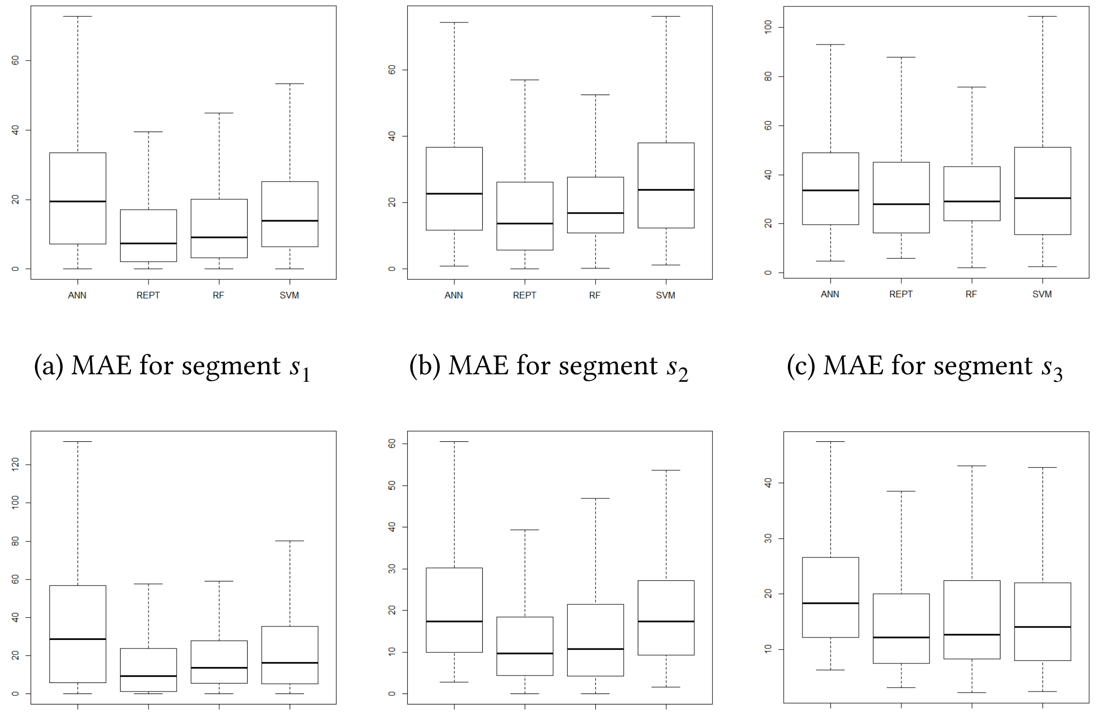
Table 4.5: Errors in predicting PED_h for ANN, REPT, RF, and SVM algorithms and for each building segment.

	ANN	**REPT**	**RF**	**SVM**
		Overall		
RMSE $[kWh/m^2]$	39.85	**33.12**	33.83	38.40
MAE $[kWh/m^2]$	29.67	**22.21**	22.35	27.41
MAPE [%]	27.02	**16.64**	16.89	21.52
		Segment s_1		
RMSE $[kWh/m^2]$	30.99	**21.99**	22.16	28.95
MAE $[kWh/m^2]$	23.04	**13.45**	13.88	18.83
MAPE [%]	40.76	**20.25**	20.47	27.32
		Segment s_2		
RMSE $[kWh/m^2]$	37.80	**29.72**	30.87	37.03
MAE $[kWh/m^2]$	28.23	**20.57**	21.52	28.02
MAPE [%]	22.33	**14.75**	15.62	20.37
		Segment s_3		
RMSE $[kWh/m^2]$	49.76	**47.69**	49.84	50.31
MAE $[kWh/m^2]$	38.78	**36.26**	37.53	37.76
MAPE [%]	20.87	**15.90**	17.18	17.19

Figure 4.3 analyses more in depth the distribution of prediction errors, by reporting the box plots for absolute and percentage error of the four algorithms over the three segments. The difference between REPT and the other algorithms is clear especially in segments s_1 and s_2.

4.4.5 Comparison with other approaches

Compared to a single step approach based on a single regression model for all building segments (Table 4.6), the two-layers approach is capable to provide a notable reduction of errors in the prediction of PED_h. Indeed, REPT algorithm applied to the overall dataset has still the best performances with a MAPE of 29.82% (compared to 16.64% of the two-layers approach with REPT). Also RMSE and MAE are higher with



(a) MAE for segment s_1 (b) MAE for segment s_2 (c) MAE for segment s_3

(d) MAPE for segment s_1 (e) MAPE for segment s_2 (f) MAPE for segment s_3

Figure 4.3: Box plots of MAE and MAPE of the estimation of energy demand for each algorithm and for the three different building segments.

the single step approach (respectively, 48.92 kWh/m^2 and 35.24 kWh/m^2) than with the two-layers approach (respectively, 33.12 kWh/m^2 and 22.21 kWh/m^2).

Table 4.6: Errors in predicting PED_h for ANN, REPT, RF and SVM algorithms using a single step regression.

	ANN	REPT	RF	SVM
RMSE [kWh/m^2]	108.73	**48.92**	49.79	55.83
MAE [kWh/m^2]	95.91	**36.74**	37.11	38.49
MAPE [%]	87.51	**29.82**	30.20	33.20

The HEDEBAR methodology was compared also with other approaches proposed in [103, 118]. Both works make use of ANNs to analyse data from EPCs and to estimate the energy demand and both use 12 features to describe buildings, even if the two feature sets are not identical. Khayatian, Sarto, and Dall'O' [103] use more detailed features of the Envelope category (e.g., U-values of roof and basement, opaque and glazed surfaces), but none of the Energy category. Instead, the work of Buratti, Barbanera, and Palladino [118] is more focused on the Energy category (e.g., type of fuels, type and power of heating, CO_2 emissions), but excludes variables of other categories like construction year, gross heated volume and quality of building envelope. Moreover, Buratti, Barbanera, and Palladino [118] predict the value of *global energy performance index*, which considers also the domestic hot water, therefore it is not comparable with HEDEBAR.

Compared to [103], HEDEBAR produces a slightly higher value of MAPE (16.64% versus 14.44%) in the estimation of PED_h, yet using fewer building features (10 rather than 12). Moreover, the use of REPT algorithm is a key advantage, since the tree models that we obtain for each building segment facilitate the interpretation of the certification method, highlighting the most relevant features that affect the energy demand for each class of energy efficiency. The higher error values obtained using our methodology can be motivated with the lower number of features used by our model and with the potentially different qualities of the data sets (e.g., a lower uniformity in the assignment of parameters for our certificates). In fact, the comparison would be even more significant if the two approaches used the same EPC data set.

4.4.6 Interpretation of the Segment estimation model

This section provides a qualitative analysis of the impact of explanatory variables (building features) on the heating energy demand. The analysis focuses on the *Segment estimation* phase and exploits the interpretable REPT model. To better understand how the REPT model assigns a given building to a *segment of energy demand*, the decision tree of the resulting *Segment estimation* model was inspected. The analysis of the primary rules of the tree is performed for the classification layer.

The first four levels of the REPT model for Segment estimation are illustrated in Figure 4.4. The tree has an overall size of 342 nodes, with a maximum root-to-leaf path length of 20 nodes. The *Average U-value of vertical opaque envelope* parameter (U_o) is the one mostly affecting the energy demand. Also the *aspect ratio* (R) and the *construction year* (y_c) appear at the first three levels of the tree. The *Average U-value of the windows* (U_w) and *Average global efficiency for space heating* (η_h) appear only at the fourth level.

The descriptive power of the REPT model comes from its capacity of highlighting the features that mostly affect the energy demand, according to the analyzed certification system. Indeed, each path of the REPT model includes a subset of building properties. Therefore, the classification rules inferred from the main paths of the tree facilitate the model interpretation by bringing out its main features. Table 4.7 resumes the main rules of the REPT model. Rules are structured in two parts: (ii) the *rule antecedent* includes the buildings features and the corresponding ranges of values; (ii) the *rule consequent* includes the energy demand segment associated to buildings that satisfy the conditions

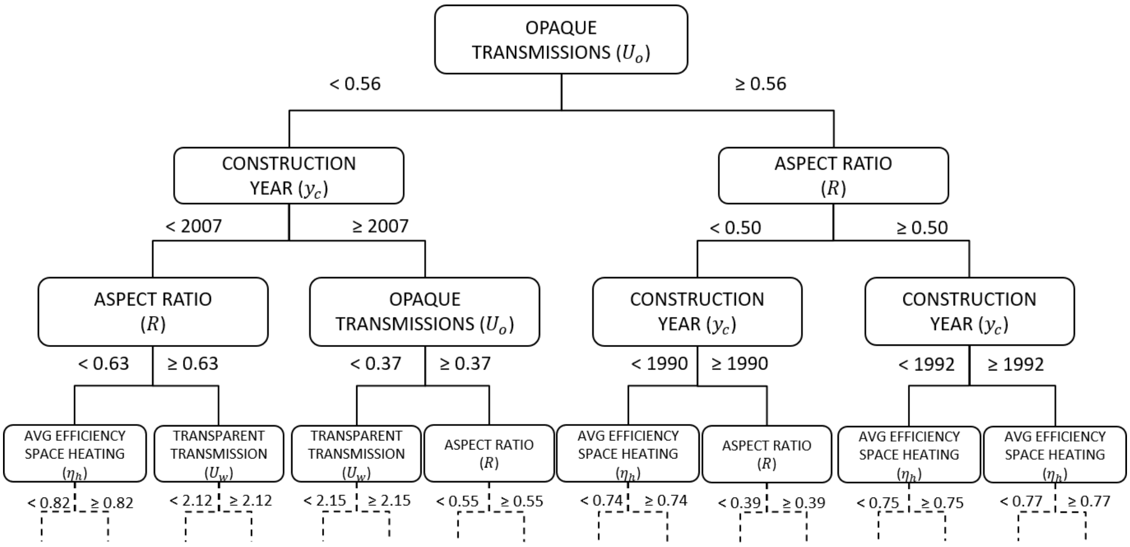

OPAQUE TRANSMISSIONS (U_o)
< 0.56
≥ 0.56
CONSTRUCTION YEAR (y_c)
ASPECT RATIO (R)
< 2007
≥ 2007
< 0.50
≥ 0.50
ASPECT RATIO (R)
OPAQUE TRANSMISSIONS (U_o)
CONSTRUCTION YEAR (y_c)
CONSTRUCTION YEAR (y_c)
< 0.63
≥ 0.63
< 0.37
≥ 0.37
< 1990
≥ 1990
< 1992
≥ 1992
AVG EFFICIENCY SPACE HEATING (η_h)
TRANSPARENT TRANSMISSION (U_w)
TRANSPARENT TRANSMISSION (U_w)
ASPECT RATIO (R)
AVG EFFICIENCY SPACE HEATING (η_h)
ASPECT RATIO (R)
AVG EFFICIENCY SPACE HEATING (η_h)
AVG EFFICIENCY SPACE HEATING (η_h)
< 0.82
≥ 0.82
< 2.12
≥ 2.12
< 2.15
≥ 2.15
< 0.55
≥ 0.55
< 0.74
≥ 0.74
< 0.39
≥ 0.39
< 0.75
≥ 0.75
< 0.77
≥ 0.77

Figure 4.4: REPT model of the classification phase. The first four levels of the tree are illustrated and, for each path, the histogram illustrates the number of leaves assigned to each segment.

of the rule antecedent. For each building segment, we selected the most significant path of the tree, i.e., the one with the highest classification precision among those including more than 500 buildings. For the considered model, these rules have a classification precision ranging from 72.7% to 93.7%.

The classification rules bring out the most representative building properties of each segment and their ranges of values. On the other hand, by applying a few rules, it is possible to estimate the segment of a building, i.e., whether its energy demand is low, high, or very high, and what features cause such classification. Moreover, with a view to improving the efficiency of a building, the model makes possible to individuate the features that mostly cause its high (or very high) energy demand. A proper change of their values (e.g., by performing targeted refurbishment actions), can substantially increase the energy efficiency of the building. For some buildings, bringing the values of few features within the appropriate ranges causes their reassignment to a lower segment.

Hence, rules like those in Table 4.7 are an important source of information about the classification model. For instance, the rule for segment s_1 is based on the transmittance of the opaque and transparent envelopes and on the construction year. More specifically, the rule states that, if the building envelope provides a very high thermal insulation it has low heat dissipation. Moreover, buildings that satisfy this rule were built with construction standards adopted from 2007 onwards, thus guaranteeing an overall energy efficiency that is classified into segment s_1. The rule for segment s_2 includes also the aspect ratio and the global efficiency for space heating. This rule shows that, for high energy demand buildings, aspect ratio has intermediate values, while the global efficiency is always lower than 0.77. The U-value of opaque envelope has a minimum value of 0.56, which is higher than the maximum value used in the previous rule of s_1 (0.37), thus implying always a higher transmittance. Moreover, the rule includes high energy demand buildings constructed since 1992, i.e., the minimum construction year for this rule is 15 years lower than the one for the previous rule (2007). The rule selected for segment s_3 has very high values of aspect ratio, starting from a minimum of 0.8 which is higher than the maximum value for s_2 (0.68). This feature itself highly affects the energy efficiency, as the very high energy demand is due to a wider dispersant surface for the same volume unit. Additional negative factors are represented by the high lower bounds for U-values intervals and the construction year always before 1991.

Table 4.7: Main rules of the REPT model for classification. For each row, intervals are specified only for the variables used by the corresponding rule. The last column contains the segment assigned by the rule.

		Rule	**antecedent**			**Rule consequent**
U_o	y_c	R	U_w	η_h	q_{env}	**Segment**
$[0,0.37[$	$[2007, +\infty[$		$[0, 2.15[$			$\Rightarrow s_1$
$[0.56, +\infty[$	$[1992, +\infty[$	$[0.5, 0.68[$		$[0,0.77[$		$\Rightarrow s_2$
$[0.78, +\infty[$	$]-\infty, 1991]$	$[0.63, 0.98[$	$[3.41, +\infty[$	$[0, 0.75[$	$[1.5, 5]$	$\Rightarrow s_3$

4.4.7 Parameters setting of algorithms

This section describes how the main parameters of the four algorithms have been tuned for both phases, i.e., *Segment estimation* with the objective of maximizing classification accuracy and *Local energy demand prediction* with the aim of minimizing MAPE, MAE, and RMSE. For each of the four algorithms, the tuning procedure produced similar optimal configurations between the two phases. As an example, this section describes the results of parameters tuning during the *Local energy demand prediction* phase.

ANN. According to the experimental results, a single hidden layer has been adopted for ANN, since using more than one layer didn't bring any improvement of accuracy. Some common rules of thumb for the size of the hidden layer are suggested by different works like [119], where the number of neurons are related to the number of attributes and output variables. Overall, the size of the hidden layer should be high enough to let the ANN model the problem correctly, but also low enough to ensure generalization. In our tests, we used an increasing number of neurons, ranging in the interval [4,100] until the prediction error starts to grow due to over-fitting. The other parameters for the ANN configuration have been set as follows: $learning\_rate = 0.3$, $training\_cycles = 10^3$, $\epsilon = 1 \times 10^{-5}$. The values of prediction errors for different sizes of the hidden layer are reported in Figure 4.5 (top-left plot). 16 neurons for the hidden layer provide the lowest values of the three errors, RMSE, MAE, MAPE.

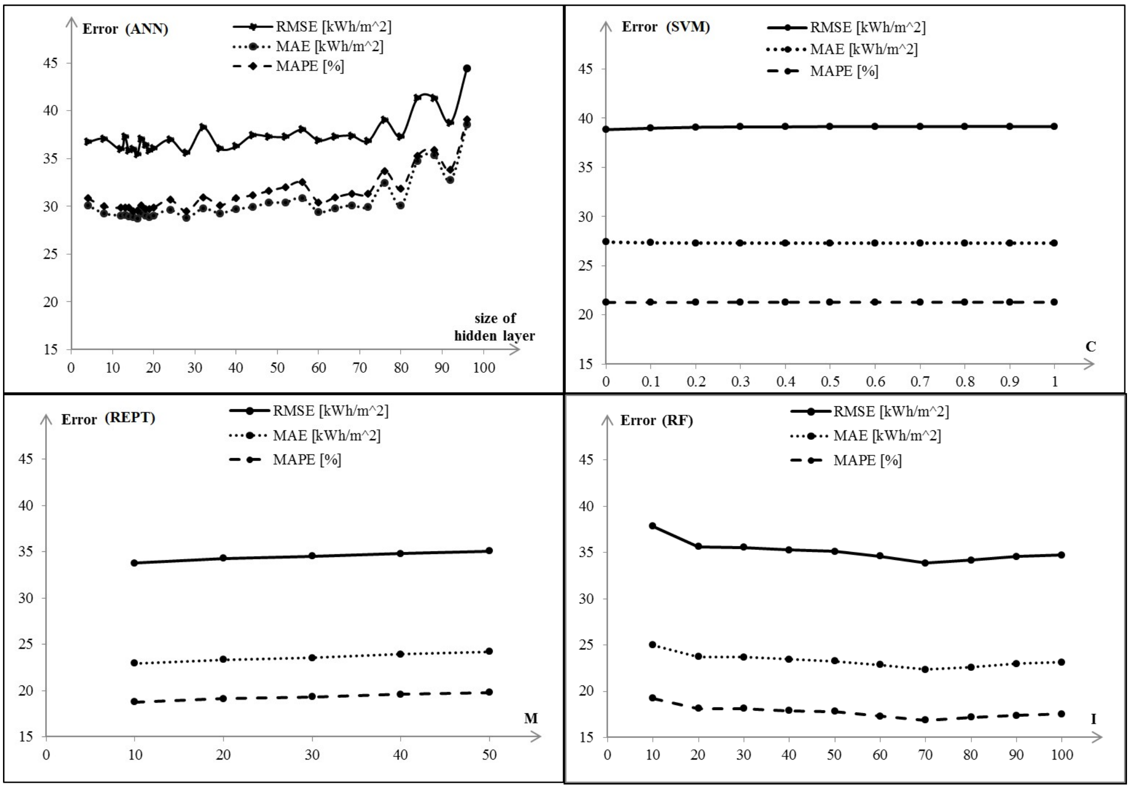


Figure 4.5: Overall prediction errors for algorithms parameter tuning: (top-left) ANN algorithm with respect to the size of the hidden layer; (top-right) SVM algorithm with respect to the complexity constant C; (bottom-left) REPT algorithm with respect to the minimum number of instances per leaf M; (bottom-right) RF algorithm with respect to the number of trees.

SVM. For SVM regression, we considered a linear kernel function and we tested the variation of prediction errors with respect to the complexity constant C. This variable is used to set a degree of tolerance for misclassification of training samples. A too large value of complexity constant can lead to over-fitting, while too small values may result in over-generalization. In our tests, values for C have been selected in the range [0,10]. The other parameter settings of the SVM are: $max\_iterations = 10^4$, convergence $\epsilon = 1\times10^{-3}$. Different kernel functions have been tested (*Polynomial*, *Dot Product*, *Gaussian*, and *Radial Basis Function* (RBF)). The *Dot Product* kernel provided the lowest prediction error (MAPE = 21.52%), with *RBF* having similar performance (MAPE = 21.80%), while

Polynomial kernel produces higher error values (MAPE = 31.95%). The *Gaussian* kernel was discarded because it employed too much time to complete the training. The values of prediction errors are reported in Figure 4.5 (top-right plot). The trends of the three error measures are nearly constant, but, we have a slightly lower value of RMSE for $C = 0$.

REPT. In REPT, we fixed the dimension of the pruning subset to one third of the training set, hence we used three folds in the algorithm ($N = 3$). No maximum tree depth has been set instead. We tuned the algorithm by varying the minimum number of instances per leaf ($M \in \{10, 20, 30, 40, 50\}$). The values of prediction errors are reported in Figure 4.5 (bottom-left plot). The three error measures slightly yet constantly increase together with M. Therefore we set $M = 10$.

RF. In RF we considered the previous settings of REPT for all the decision trees. We analysed the variation of prediction error with respect to the number of trees I in the range $[10, 100]$. The values of RMSE, MAE and MAPE are reported in Figure 4.5 (bottom-right). $I = 70$ provides the lowest error values.

4.5 Discussion

The HEDEBAR system presented in this chapter is a methodology for the automatic asset rating of the buildings energy efficiency. In particular, based on data from energy certificates, it extrapolates an overall model that reflects the (implicit) criteria used to compute the ideal *thermal energy demand* assigned into the same certificates.

Experimental results show that HEDEBAR is able to compute a thermal energy demand value with a limited and acceptable error with respect to the value calculated by the auditor. The main added value of this methodology is that it allows an automatic computation of the building energy demand using a small set of building features. This is a great advantage from the perspective of building design, as it is very important to estimate buildings energy performance in a quick and reliable way, for different combinations of structural parameters, even when data about real energy consumption are not available. Moreover, the analysis pointed out the most relevant building features (*opaque and transparent U-values*, *aspect ratio*, *global efficiency for space heating*), and those that are less important, according to the considered rating system.

From a methodological perspective, the proposed two-layer approach allows to obtain a higher performance in the estimation of the energy demand. Indeed the segmentation of the entire dataset in three different groups of buildings with similar features and energy requirements makes possible to produce differentiated models for PED_h prediction, each one targeted to the specific characteristics of its own segment. RF algorithm produces the highest classification accuracy, while REPT algorithm produces the lowest error values in predicting PED_h. REPT produces also a good classification accuracy. The combination of RF and REPT turned to be the most suitable in describing the process of estimation of the energy demand of buildings from the variables included in the energy rating dataset. Furthermore, the interpretability of REPT models makes the obtained results understandable and exploitable even if the involved users are not domain experts.

The differentiated analysis of buildings segments, even with variable prediction errors, highlighted the features that mostly affect PED_h for each segment of *low*, *high* or *very high* energy demand.

The transmittance of opaque surfaces (walls) has the highest importance for the first two building segments. Moreover, in the first segment, also transparent surfaces (windows) are very important, since for low levels of energy demand, the smaller contribution of heat loss through windows becomes crucial. On the other hand, the overall efficiency of the heating system η_h has a higher importance for those buildings with high energy demand (segment s_3). The segmented analysis highlighted also the main features impacting on energy demand for different segments of buildings and the related threshold values in the REPT model. With such information, domain experts can accurately quantify the improvements that can be made during the design of new buildings or during the refurbishment of existing ones. Also public authorities and regulatory bodies can benefit from HEDEBAR methodology, to plan future energy policies that leverage on specific building features. The provided information can support more targeted actions to reduce energy demand according to different classes of energy efficiency. Indeed the proposed methodological process allows to extract useful knowledge according to physical driving variables that can effectively support the definition of targeted retrofitting strategies (e.g., for each segment identified) in the context of regional financial investment policy.

In the considered model, *Primary Energy Demand for space heating* is not referred to real consumption, but it becomes a parameter for the comparison of buildings based on features about their structures and energy systems (asset rating). An advantage of this approach is that it learns an overall model from data about previous certificates and applies the same model also to new buildings. Therefore, it guarantees automatic rating less subject to error and it can be used to evaluate whether a certificate released by an auditor underestimates or overestimates the energy efficiency of the building, thus validating the uniformity of criteria adopted by the same auditor. In this perspective, the HEDEBAR methodology can be useful in the real estate market, since it can be used as unbiased information to determine the real market value of a building.

Chapter 5

Characterization of relevant urban topics from social networks

Several aspects of a smart city can be continuously monitored and characterized by means of data measured by sensors, smart meters and other devices physically located in the urban context. Nevertheless, useful information about the city can be obtained also from other sources outside the urban environment, like *Social Networking Sites* (SNS), i.e., online platforms that allows users to create a public profile, publish multimedia contents and interact with other users. SNS can provide meaningful information about users' perception of several aspects of a city. Similarly, the variation of the perception of the same aspects among different cities can be analysed as well.

Unlike the scenario described in Section 3.1, where multiple hardware and software entities provide heterogeneous kinds of data, here a single source is used, i.e., the Web service that provides data through the Twitter Stream Application Programming Interface (API). Such data are continuously generated by Twitter users, many of them posting contents at any time and from anywhere in the city.

In this chapter, the PhD activity is focused on the analysis of data from SNS to provide useful information about the relationship, in several respects, between citizens and various popular topics relevant for the city. In particular, the study proposes a methodology to explore large collections of posts on Twitter (*tweets*) along three dimensions (i.e., *text content*, posting *time* and *place*) to support context-aware topic trend analysis. Twitter is a popular SNS where users publish small multimedia contents like short text messages (microblogging). This characteristic eases the generation of an impressive amount of tweets about various topics, which can be analysed to understand the opinions and preferences of users on different topics. This work was published in [120].

The purpose of this study, within the overall research activity on *urban data mining*, is: (i) to find the main topics discussed by users of SNS about a given event; and (ii) to provide a characterization of topics distribution over *time* and across *space*, or, better yet, between *cities* and/or city *districts*. This makes possible to compare several cities to highlight the differences in terms of popular topics among citizens and of their temporal trends. Similarly, different districts of a same city can be compared as well.

This chapter is organised as follows. The overall context with the description of available data and of the targeted analysis are described in Section 5.1. Section 5.2 presents the related research work on the analysis of data from Twitter. Section 5.3 introduces the components of the TCharM architecture and the employed algorithms. Section 5.4 Section 5.5 illustrate respectively the experimental results and the computing scalability of the clustering algorithm. Section 5.6 provides a theoretical and an analytical comparison between TCharM and four previous studies on clustering Twitter data. Section 5.7 discusses in depth the experimental results.

5.1 Context for the analysis of relevant urban topics from Twitter

In the last few years, the application of data mining algorithms to collections of data from SNS has become an hot research topic, as microblogs like Twitter have become a popular platform with millions of users. The conciseness of their text messages allows a very large number of tweets to be published at extremely low cost, thus making Twitter a timely and fresh source of data.

Two distinct parts of data can be extracted from Twitter (and in general from a SNS): *social structure*, represented by a graph, denoting the relationship and interaction between users; and *user-generated social media*, such as texts, photos, and videos, which contain rich information about a user's behaviors and interests [1]. This study considers the second part of data with the aim of characterizing shared interests among users through the analysis of their tweets. Three features of a tweet are analysed: *text content*, *temporal feature* and *spatial feature*.

- Tweet *text content* is the text message, long 140 characters at most[1], published by the users. Due to the limited size of the single message and to the high dimensionality of many text content representations, the represented samples are inherently sparse.
- Tweet *temporal feature* is extracted from the timestamp associated with the tweet and includes date and time instant when the user posted the tweet.
- Tweet *spatial feature* represents the spatial position of people right when posting the tweet and is acquired from GPS enabled devices, with localization enabled, as geographic coordinates (i.e., latitude and longitude).

[1] At time of conducting this study.

The research activity described in this chapter led to the definition of a methodology, named *Tweets Characterization Methodology* (TCharM), addressing the analysis of *text*, *time* and *space* information of users *tweets*, to explore the distribution over time and space of frequent patterns of activities and interests. TCharM is based on two exploratory data mining techniques: (a) *Cluster analysis*, to identify cohesive groups of tweets with similar text content posted from nearby geographical areas and at close time instances, and (b) *Association rule analysis*, to find significant patterns that concisely describe each computed cluster.

Differently from the works of Chapter 3, here three features of crowd-sourced data are analysed, as text is added to time and space. The discovery of frequent *text-spatio-temporal* patterns relies on the aggregation of tweets through the *K-means clustering* algorithm [117], to generate clusters of tweets that can be concisely represented by their centroids. Each cluster can potentially reveal a group of people interested in a same topic, during a limited time interval and within a limited urban area. A suitable distance measure, named Text And Spatio-TEmporal (TASTE), has been defined to drive the clustering process by making joint use of the tweet spatio-temporal features and text content. Through TASTE, spatial and temporal distances between tweets are used to modulate the text content distance.

TCharM then locally investigates each computed cluster to mine significant patterns which reveal underlying correlations among frequent topics, tweeting times and places that simultaneously emerge from clusters. This task has been carried out using association rule analysis [117], an exploratory data mining technique to extract correlations among data items. The extracted patterns describe the cluster content using a concise and clear knowledge representation and better highlight the different topics discussed by people from different cities and at different times.

To validate the proposed approach, it was analytically compared with existing tweets clustering algorithms in terms of *clusters cohesion* over the three dimensions (Section 5.6).

The experimental evaluation of TCharM was conducted on a real collection of Twitter data related to an event that involved people from several cities worldwide, i.e., the FIFA World Cup held in Brazil in 2014. Even if not strictly related with the urban context, this use case has been selected for various reasons: (i) it included a variety of events (e.g., football matches with different teams, players, ceremonies, celebrities statements) spread over a long time period; (ii) its popularity makes possible to collect several tweets posted worldwide; (iii) people's involvement in, and perceptions of, this kind of event may vary depending on the country and the city they live in; (iv) the qualitative validation of mined clusters and rules is easy when they point out some of the interests and reactions of sports fans that were in some cases predictable (e.g., the disappointment for their team's defeats). Therefore, the considered use case allows to test the capacity of TCharM to discover how popular topics (of any kind) discussed on Twitter vary in time and among different urban areas. Performed experiments pointed out the main benefits provided by the TCharM methodology.

From the *information* perspective, the mined patterns can effectively summarise in a concise way people's perception of different events and how it varies across different cities. This information can support a proper definition of differentiated policies among cities.

From the *technical* perspective, results validated the approach in terms of (i) clusters cohesion over the three tweet dimensions (text, time and space) and (ii) computational cost and scalability of the algorithms when a suitable number of execution nodes are employed.

The public stream endpoint offered by the Twitter APIs was monitored over a time period of 27 days from June 18th to July 14th 2014, by tracking a selection of keywords related to the 2014 FIFA World Cup. Tweets in English and with the exact GPS coordinates of the user location were extracted. The resulting collection includes 302,052 tweets.

Experimental results of Section 5.4 demonstrate the effectiveness of TCharM in identifying interesting clusters of tweets about hot topics for users in different cities and time periods. Each mined cluster is timely centered around one event and refers to a specific topic. Moreover, clusters show good spatio-temporal cohesion around their centroid, as demonstrated in Section 5.6.

5.2 Related work

While some research approaches address just the analysis of text content [121, 122, 123, 124, 125, 126], other consider also spatio-temporal information. Different types of analysis have been addressed to (i) discover nearby activities using geo-tagged tweets [127], (ii) detect events through cluster analysis [128, 129], (iii) analyse citizen feedbacks [130, 131], (iv) identify the beginning of information diffusion through social networks [132], and (vi) mine user opinions [133].

Although a large body of research focused on Twitter data analysis has already been proposed [121, 126, 129], the potential impact of mining social data is still largely unexplored because various critical issues are yet to be addressed when analyzing tons of tweets to identify insightful nuggets. (i) Since a large number of tweets are continuously being posted worldwide, the size of tweet collections to be explored grows at an ever increasing rate. (ii) The collection of tweets generally tends to be scattered in spatio-temporal dimensions, and the conciseness of the tweet messages increases the brevity of their textual content (iii) Furthermore, the distribution of tweets can be characterized by different spatial and temporal granularities. (iv) Mined knowledge should be represented using concise and understandable patterns to enable its exploitation by domain experts. Thus, innovative data analytics solutions are needed to effectively and efficiently mine large Twitter data collections.

Various approaches have been proposed to cluster tweets collections taking into account textual content and spatio-temporal information [127, 129], though such works do not jointly exploit all these features in the clustering process. Instead, they typically use a subset of features for clustering, while remaining features are considered either in the post-processing phase, for instance to refine or characterize discovered clusters, or in the preprocessing phase, for example to specify spatial or temporal segments in which tweets are locally clustered based on textual content. Kim et al. [127] cluster tweets based on their GPS coordinates using the K-means algorithm, while Steiger, Resch, and Zipf [129] use a spatio-temporal clustering based on Self Organizing Maps (SOM). In both approaches, discovered clusters are then analysed to identify the main targeted topic. Density based clustering, mainly based on the DBSCAN algorithm, has been also adopted to detect high spatial concentrations or temporal bursts of tweets about specific topics [134, 128, 131, 135]. For instance, Lee, Wakamiya, and Sumiya [131] group user trajectories derived from geo-tagged tweets and explore massive crowd movements, while Sakai et al. [135] extract local bursty keywords and identify their dense areas to enhance local situation awareness.

Contribution of the research activity. Differently from all the works above [127, 128, 134, 136], the TCharM framework *jointly* exploits the spatio-temporal features and tweet textual content to drive the clustering process. Our main purpose is to discover cohesive clusters focused on single topics and, at the same time, with precise spatio-temporal references. Through the TASTE distance measure, TCharM explores the three dimensions characterizing tweets, to discover, in one step, groups of messages with similar content but posted in nearby time and space.

With respect to Lee [128], TCharM includes the spatial information in the clustering algorithm, as the location of the cluster centroid is considered a primary feature for the subsequent characterization. With respect to Cunha, Soares, and Mendes Rodrigues [136], in the TASTE measure spatial and temporal distances are expressed as exponential functions and used to modulate the content distance, in order to significantly penalize farther tweets in time and/or space. Differences between TASTE and the distance measures used by the other two works ([134, 127]) are more evident, as described in Section 5.6.

5.3 TCharM architecture

The main components of the *Tweets Characterization Methodology* (TCharM) architecture are shown in Figure 5.1. The components are briefly introduced below while a more thorough description of each of them is given in the following subsections.

The first activity is *data collection and preprocessing*. All information about tweets, including text content, publication time and user geographic location, are retrieved through the Twitter Stream Application Programming Interfaces (APIs) specifying a set of filter parameters (e.g., keywords, hashtags). The collected data then undergo a preprocessing phase to be represented in a format suitable for the subsequent clustering analysis. The adopted data model is described in Section 5.3.1. The output of the preprocessing is a dataset where each record corresponds to a single tweet and contains basically three features: *text content*, *time* of tweet posting and *location* of the user when posting the tweet.

Once the dataset is ready, the *cluster analysis* elaborates its records in order to partition the tweets collection into cohesive groups (clusters). For this activity, a novel combined distance measure, called Text And Spatio-TEmporal (TASTE), is used to cluster Twitter messages considering their spatio-temporal information and the text content as well.

Finally, TCharM analyses each discovered cluster to mine a set of patterns describing the cluster content. Specifically, through *association rule analysis*, patterns of relevant correlations among tweets text contents, posting times and geographic areas are extracted for each cluster. Extracted rules are then categorized into four classes defined according to the types of modeled correlation among the tweets attributes.

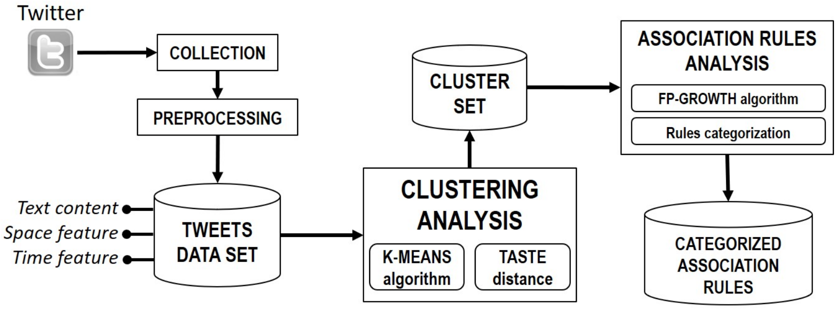


Figure 5.1: The TCharM architecture

The following subsections describe (i) a formal representation of tweets features (Section 5.3.1), (ii) the clustering algorithm with the formalization of the proposed TASTE distance measure (Section 5.3.2), and (iii) the characterization of clusters through association rules (Section 5.3.3).

5.3.1 Twitter data representation

A formal definition of the representation adopted in this study for tweet data introduced in Section 5.1 is reported below.

Definition 5.3.1 (Tweet data representation). *Let $\mathcal{D}$ be a set of tweets and $\Sigma = \{w_1, \ldots, w_k\}$ the set of words appearing in at least one tweet in $\mathcal{D}$. An arbitrary tweet $\tau_i \in \mathcal{D}$ is represented as a triplet $\tau_i = (t_i, s_i, W_i)$ where t_i and s_i are respectively the temporal and spatial features of τ_i, while $W_i \subseteq \Sigma$ is the tweet text content.*

The *temporal feature* t_i is the *timestamp* indicating when tweet τ_i was posted, while the *spatial feature* s_i is the pair of *geo-coordinates* reporting from where tweet τ_i was posted. The *text content* W_i is given by the subset of words w_j ($w_j \in \Sigma$) appearing in tweet τ_i, with their respective frequencies.

Unweighted word frequencies do not properly characterize tweet text content, since words related to more specific events may appear with lower frequency than common words. Therefore, in this study the *Term Frequency - Inverse Document Frequency* (TF-IDF) scheme [137] has been adopted to increase the relevance of specific words for each tweet, while reducing the importance of common terms in the collection. To weight word relevance based on the TF-IDF scheme, the tweet text content is transformed using the following representation [138].

Definition 5.3.2 (Tweet text content representation). *Let $\tau_i = (t_i, s_i, W_i)$ be an arbitrary tweet in collection $\mathscr{D}$. The tweet text content W_i is a vector of k elements corresponding to words in Σ (i.e., $k = |\Sigma|$). Each vector element $W_i[j]$ contains the TF-IDF weight of word w_j for tweet τ_i. $W_i[j]$ is computed as $W_i[j] = TF(\tau_i, w_j) \cdot IDF(w_j)$, where terms $TF(\tau_i, w_j)$ and $IDF(w_j)$ are defined as follows:*

1. *$TF(\tau_i, w_j)$ is the relative frequency of word w_j for tweet τ_i. $TF(\tau_i, w_j) = f(\tau_i, w_j) / \sum_{l=1}^{k} f(\tau_i, w_l)$, where $f(\tau_i, w_j)$ is the number of times word w_j appeared in tweet τ_i and $\sum_{l=1}^{k} f(\tau_i, w_l)$ is the total number of words contained in τ_i.*

2. *$IDF(w_j)$ is the relative frequency of word w_j in $\mathscr{D}$. $IDF(w_j) = log(|\mathscr{D}|/|\mathscr{D}_j|)$ where $|\mathscr{D}|$ is the number of tweets in $\mathscr{D}$ and $|\mathscr{D}_j|$, $\mathscr{D}_j = \{\tau_i \in \mathscr{D} : f(\tau_i, w_j) > 0\} \subseteq \mathscr{D}$, is the number of tweets in $\mathscr{D}$ which contain (at least once) word w_j.*

The TF-IDF weight $W_i[j]$ for word w_j in tweet τ_i is high when w_j appears with high frequency in tweet τ_i but low frequency in tweets in the collection $\mathscr{D}$. When word w_j appears in more tweets, the ratio inside the IDF *log* function approaches 1, and both the IDF(w_j) value and the TF-IDF weight $W_i[j]$ become close to 0. Hence, the approach aims at filtering out common words.

5.3.2 Clustering analysis of tweets

Cluster analysis partitions objects into groups so that objects within the same group are more similar to each other than to the ones assigned to different groups [117].

In TCharM, the *K-means* algorithm is used for clustering tweet data collections [139], as it provides good quality solutions in many application domains and generates clusters of tweets that can be concisely represented by their *centroids*. The K-means algorithm segments data samples into K clusters that can be shortly represented through their *centroids*, given by the mean value of the samples in the clusters.

In TCharM, the algorithm is initialized with a random selection of K tweets of the tweet collection as centroids. The other tweets are assigned to the cluster of the nearest centroid. In the next iterations, the centroids are recomputed as mean values of the tweets features within each cluster and tweets are reassigned accordingly. The process iterates until a convergence criterion is met, e.g., the centroids do not change, or one or more parameters have reached a target value, e.g., maximum number of iterations.

TASTE **distance measure**

In this study, the K-means algorithm uses the Text And Spatio-TEmporal (TASTE) distance measure, that takes into account the three tweet features at once to determine a single overall distance between tweets [120]. The TASTE distance measure is formally defined as follows.

Definition 5.3.3 (TASTE distance measure)**.** *Let* $\tau_i = (t_i, s_i, W_i)$ *and* $\tau_j = (t_j, s_j, W_j)$ *be two arbitrary tweets in collection* $\mathscr{D}$*. The* TASTE *distance measure between tweets* τ_i *and* τ_j *is defined as*

$$d_{TASTE}(\tau_i, \tau_j) = d_W(W_i, W_j) \cdot (k_s \cdot \mathrm{e}^{p_s \cdot d_s(s_i, s_j)} + k_t \cdot \mathrm{e}^{p_t \cdot d_t(t_i, t_j)}) \tag{5.1}$$

where parameters $k_s, k_t, p_s, p_t \in \mathbb{R}$; $k_s, k_t \in [0{,}1]$ and $k_s + k_t = 1$. Terms $d_W(W_i, W_j)$, $d_s(s_i, s_j)$, and $d_t(t_i, t_j)$ measure the distance on tweet text content, spatial feature, and temporal feature, respectively. These distances have been normalized in the range [0,1] using the *min-max* normalization method [117].

TASTE is defined as a measure of dissimilarity. Given tweets τ_i and τ_j, lower values of $d_{TASTE}(\tau_i, \tau_j)$ denote a higher similarity between τ_i and τ_j, while higher values of $d_{TASTE}(\tau_i, \tau_j)$ denote a lower similarity.

In the TASTE measure, spatial and temporal distances ($d_s(s_i, s_j)$ and $d_t(t_i, t_j)$) modulate the text content distance ($d_W(W_i, W_j)$) to determine the overall value of $d_{TASTE}(\tau_i, \tau_j)$. The exponential form is used for $d_s(s_i, s_j)$ and $d_t(t_i, t_j)$ to significantly penalize pairs of tweets with a large space and/or time distance.

The parameters of the TASTE measure can be conveniently tuned to fit scenarios with different spatial and temporal scales. Parameters k_s and k_t weight the relevance of spatial and temporal distances in modulating the text content distance. Parameters p_s and p_t are included as exponents to adjust the (possibly differentiated) growth rates of exponential terms of spatial and temporal distances. For instance, to discover clusters of tweets with a high temporal cohesion, but possibly spread over a large geographic area, suitably higher values should be assigned to parameter p_t to penalize distances in time.

In TASTE, three different measures are used to compute $d_W(W_i, W_j)$, $d_s(s_i, s_j)$, and $d_t(t_i, t_j)$ based on the data type describing tweet text content, spatial feature and temporal feature.

Text content distance measure. The distance between the weighted word frequency vectors W_i and W_j of tweets τ_i and τ_j is evaluated using the *cosine distance measure*, often used to compare documents in text mining [117, 140]. It is defined as

$$d_W(W_i, W_j) = \arccos\,(cos(W_i, W_j)). \tag{5.2}$$

Term $cos(W_i, W_j)$ in Equation 5.2 represents the *cosine similarity* between W_i and W_j, i.e.,

$$cos(W_i, W_j) = \frac{\sum_{l=1}^{k} W_i[l]W_j[l]}{\sqrt{\sum_{l=1}^{k} W_i[l]^2} \cdot \sqrt{\sum_{l=1}^{k} W_j[l]^2}} \tag{5.3}$$

where k is the cardinality of the word set Σ in collection $\mathscr{D}$ ($k = |\Sigma|$).

The value range is [0,1] for the cosine similarity $cos(W_i, W_j)$, while the value range for the content distance measure $d_W(W_i, W_j)$ is $[0, \pi/2]$. When $cos(W_i, W_j) = 1$, then $d_W(W_i, W_j) = 0$ which describes the exact similarity of text content for tweets τ_i and τ_j. When $cos(W_i, W_j) = 0$, then $d_W(W_i, W_j) = \pi/2$ which points out that tweets τ_i and τ_j have completely different texts.

Temporal distance measure. The tweet temporal feature is encoded as an integer number representing the time instant when the tweet was posted. The *Euclidean distance* [117] is adopted here as the distance on temporal features t_i and t_j of tweets τ_i and τ_j respectively. As t_i and t_j are expressed as time instants, the temporal distance measure $d_t(t_i, t_j)$ is computed as the absolute value of their difference, i.e.,

$$d_t(t_i, t_j) = |t_i - t_j|. \tag{5.4}$$

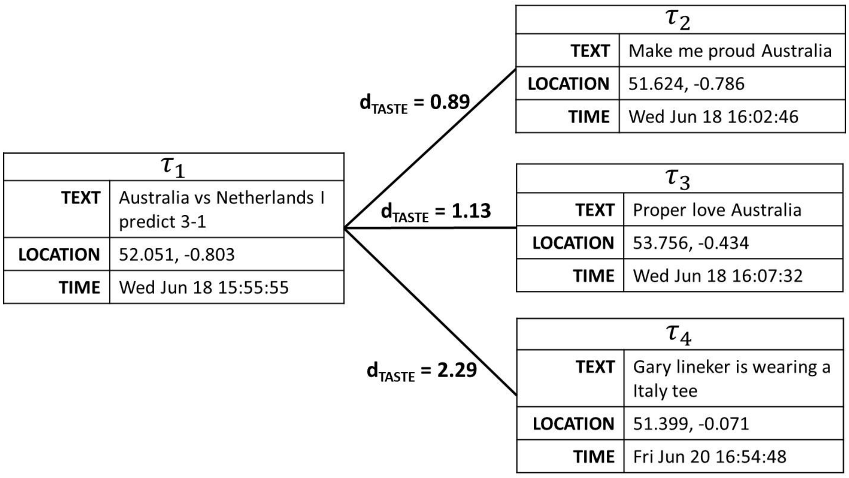
Spatial distance measure. The *Haversine distance* is used here as spatial distance between tweets. It corresponds to the shortest distance over the earth's surface between two points s_i and s_j. Hence, the spatial distance $d_s(s_i, s_j)$ between tweets τ_i and τ_j is computed as

$$d_s(s_i, s_j) = 2 \cdot \boldsymbol{R} \cdot \arcsin(\sqrt{h}) \tag{5.5}$$

where $h = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2)$ and $\Delta\varphi$ and $\Delta\lambda$ are latitudinal and longitudinal differences between the tweets and $\boldsymbol{R}$ is a constant value equal to the Earth's mean radius (6,371 km).

The content, spatial and temporal distance measures defined above satisfy the positivity, symmetry, and triangle inequality properties that characterize a metric [117]. It easily follows that the TASTE measure also verifies these properties. Specifically, the following properties hold. (i) *Positivity*: $d_{TASTE}(\tau_i, \tau_j) \geq 0$ for all $\tau_j, \tau_i \in \mathscr{D}$, while $d_{TASTE}(\tau_i, \tau_j) = 0$ only if $\tau_i = \tau_j$. (ii) *Symmetry*: $d_{TASTE}(\tau_i, \tau_j) = d_{TASTE}(\tau_j, \tau_i)$ for all $\tau_j, \tau_i \in \mathscr{D}$. (iii) *Triangle inequality*: $d_{TASTE}(\tau_i, \tau_j) \leq d_{TASTE}(\tau_i, \tau_k) + d_{TASTE}(\tau_k, \tau_j)$ for all $\tau_i, \tau_k, \tau_j \in \mathscr{D}$.

As an example, Figure 5.2 reports four sample tweets (τ_1 to τ_4) with their text content, temporal and spatial features. The values of d_{TASTE} between tweet τ_1 and the other tweets are also specified. Tweets are about the 2014 FIFA World Cup. It is worth



τ_1	
TEXT	Australia vs Netherlands I predict 3-1
LOCATION	52.051, -0.803
TIME	Wed Jun 18 15:55:55

τ_2	
TEXT	Make me proud Australia
LOCATION	51.624, -0.786
TIME	Wed Jun 18 16:02:46

τ_3	
TEXT	Proper love Australia
LOCATION	53.756, -0.434
TIME	Wed Jun 18 16:07:32

τ_4	
TEXT	Gary lineker is wearing a Italy tee
LOCATION	51.399, -0.071
TIME	Fri Jun 20 16:54:48

Figure 5.2: Sample tweets about 2014 FIFA World Cup with TASTE distance values

noting that tweets τ_2 and τ_3 have a higher similarity with τ_1 than with τ_4. Tweets τ_1, τ_2 and τ_3 have a similar text content as they all talk about the Australia football team. Tweets τ_2 and τ_3 were posted almost at the same time as τ_1, but τ_3 exhibits a farther geographic location from τ_1 than τ_2. This larger spatial distance penalizes the similarity on the text content and finally provides a higher value of d_{TASTE} for tweet τ_3. Conversely, tweet τ_4 exhibits a significantly higher TASTE distance from τ_1 even though it was posted in the neighborhood, as τ_4 has a completely different content from τ_1 and it was posted two days later.

Clustering Evaluation

For the internal validation of clustering results, TCharM adopts the *Sum of Squared Errors* (*SSE*) quality index [141]. The *SSE* index measures the cluster cohesion in prototype-based clusters, i.e., how objects in a cluster are closely related to the corresponding centroid. SSE is defined as the sum of the squared distances between each member of the cluster and its centroid and here it is computed as

$$SSE = \sum_{i=1}^{K} \sum_{\tau_j \in C_i} d_{TASTE}(\tau_j, c_i)^2 \tag{5.6}$$

where c_i is the centroid of cluster C_i, and C_i is included in a cluster set with K clusters. $d_{TASTE}(\tau_j, c_i)$ is the TASTE distance between a tweet $\tau_j \in C_i$ and the centroid c_i of C_i.

5.3.3 Clusters characterization with Association Rules

After the cluster set has been generated, each cluster is then locally explored to characterize its content. Specifically, each cluster is analysed to discover underlying correlations in the text content, and between text content and the spatial and temporal features characterizing tweets. Cluster characterization makes use of *association rules* as reference pattern type [142]. Association rules analysis is an exploratory data mining technique to mine correlations among data items.

To enable the association analysis process, tweets contained in the cluster under analysis are tailored to a transactional data format. Consider an arbitrary cluster C included in the cluster set computed on tweet collection $\mathscr{D}$. The *transactional tweet dataset* $\mathscr{D}_{\mathscr{T}}(C)$ for cluster C is a set of transactions. Each *transaction* $\mathscr{T}_i$ corresponds to a tweet $\tau_i \in C$ and it consists of a set of tweet features called *items*, represented in the form $\{attribute : value\}$. The items of the generic transaction $\mathscr{T}_i$ are (i) each single *word* $w \in W_i$ appearing in the text content of tweet τ_i, (ii) the value of the *spatial feature* s_i of τ_i, and (iii) the value of the *temporal feature* t_i of τ_i.

An *association rule* is an implication in the form $r : X \Rightarrow Y$, where X and Y are disjoint *itemsets* (i.e., sets of items). X and Y are denoted as *rule antecedent* and *consequent*, respectively. Association rules extraction is commonly driven by rule support and confidence quality indexes. Rule support (*supp*) is the percentage of tweets in cluster C that contain both X and Y. Rule confidence ($conf$) is the percentage of tweets in cluster C containing X that also contain Y. In some cases, measuring the strength of a rule in terms of support and confidence values may be misleading. When the rule consequent has a high support value, the rule may be characterized by a high confidence value even if its actual strength is relatively low. To overcome this issue, the *lift* (or correlation) index [117] may be used, beyond the confidence index, to measure the (symmetric) correlation between sets X and Y.

To support the exploration of the mined rule set, TCharM exploits a categorization of rules into few *classes*, built upon the attributes characterizing Twitter data, i.e., tweet spatial feature (denoted $Location$ (L)), tweet temporal feature ($Time$ (T)), and text content of the tweet message (*TextContent* (TC)). Each class refers to correlations among a subset of the above attributes. Specifically, four classes of rules have been defined, aimed at progressively providing more detailed information about the cluster content.

1. *TextContent class (TC).* This class focuses on tweet text content. Patterns model correlations between words in tweet messages and these are aimed at capturing the peculiar characteristics of messages in the cluster (i.e., which topics attract/involve users). This class omits both spatial and temporal details.

2. *Location-TextContent class (L-TC).* This class analyses the correlations between the words in tweet messages and the locations where tweets have been posted. It makes it possible to identify the topics attracting/involving users in a given city.

3. *Time-TextContent class (T-TC).* This class analyses the correlation between words in tweet messages and the time when tweets have been posted so as to discover the topics attracting/involving users in a given time frame.

Dataset	Time window	Geographical partition	Number of tweets	Average tweets length
$\mathscr{D}_{(TW1,UK)}$	1	UK	29,864	8.10
$\mathscr{D}_{(TW1,USA)}$	1	USA	26,447	8.02
$\mathscr{D}_{(TW2,UK)}$	2	UK	15,175	8.43
$\mathscr{D}_{(TW2,USA)}$	2	USA	19,828	8.27
$\mathscr{D}_{(TW3,UK)}$	3	UK	34,392	8.46
$\mathscr{D}_{(TW3,USA)}$	3	USA	50,028	8.06

Table 5.1: Main characteristics of selected reference tweets data sets

4. *Location-Time-TextContent class (L-T-TC).* This class considers all the properties characterizing tweets in order to analyse the correlation between the words in tweet messages, the time when and the location where the tweets were posted. It makes it possible to discover the topics attracting/involving users in a given time frame and city.

5.4 Experimental Results

This section presents the results of the performed experiments, regarding: (i) *assessment of the proposed clustering approach* of the computed cluster sets, (ii) *clusters content characterization* through association rules analysis, and (iii) *performance evaluation* in terms of overall execution time and scalability.

To analyse how the tweet text content developed over time, the tweet collection was partitioned according to three *time windows* following the official time schedule of the football matches. *Time window #1* and *time window #2* cover respectively the first and the second stage time period (i.e., from June 18th to June 27th and from June 28th to July 3rd), while *time-window #3* covers the remaining time period from the quarter-finals to the end (i.e., from July 4th to July 14th). The number of tweets is comparable in the three windows. The tweet spatial distribution was then locally analysed within each of the three time windows based on tweet geo-coordinates. English speaking countries like the United Kingdom (UK), USA, and Central America showed higher tweets concentrations than other areas. Hence, two *spatial partitions*, corresponding to *UK* and *USA*, were selected for each time window. Table 5.1 summarizes the main characteristics of the six resulting datasets.

5.4.1 TCharM implementation

The entire data analysis process has been implemented as a Scala application in the open source computing framework *Apache Spark* (version 1.5) [85]. This framework was selected because it is currently one of the leading platforms for data analytics and it provides a Machine Learning library (MLlib) which has been exploited and extended in this study to support all the functionalities of TCharM.

MLlib is used for the TF-IDF weighting score calculation in the data preprocessing phase. For the subsequent cluster analysis, the K-means algorithm available in MLlib has been extended by integrating the TASTE measure. For association rule analysis, the FP-growth algorithm [143] available in MLlib was adopted to generate association rules from the computed clusters.

The preliminary data collection step relies on Twitter's Streaming Application Programming Interfaces (APIs) to retrieve tweets data.

Experiments were executed on a cluster of 3 master nodes (DELL PowerEdge R620 with 128GB of RAM) and 30 worker nodes (18 DELL PowerEdge R720XD with 96GB of RAM, 2 SuperMicro with 64GB of RAM, and 10 SuperMicro with 32GB of RAM). Each node runs Cloudera distribution based on Apache Hadoop including HDFS and Apache Spark (version 1.5) for Big Data distributed applications on Linux Ubuntu 14.04.02 LTS.

5.4.2 Clustering analysis

The parameters for the clustering analysis were set to best fit the considered use case, the 2014 FIFA World Cup, which involves people worldwide. The same relevance was assigned to spatial and temporal terms in modulating the text distance, i.e., $k_s = k_t = 0.5$. However, as usually happens on Twitter, most reactions to a given event (e.g., a football match) are likely to be published as soon as the same event occurs (or within a short delay), even from quite distant locations. Indeed, while users interested in the same event can be also located in different areas, it is unlikely that they tweet at completely different times. Therefore, to group tweets with very close temporal distances, the weight of the temporal exponent p_t were set to a higher value than the spatial one p_s. $p_s = 3$ and $p_t = 6$ provide the lowest variability of SSE among clusters for different values of K (number of clusters). K was then set to 200 as a good trade-off to minimize SSE and to limit the number of clusters as well.

The resulting cluster set includes one cluster with about 800 tweets, while 16.5% of clusters contain from 200 to 400 tweets, 41.5% of clusters from 100 to 200 tweets, and the remaining 41.5% less than 100 tweets. The mean value of cluster size is 132 tweets, while the median value is 111 tweets.

5.4.3 Clusters characterization

In this section, the content of the extracted clusters is concisely described using association rules to model correlations among tweet features (text content, location, and time). The rules are extracted according to the rule templates defined in Section 5.3.3. To discuss the type of information that can be mined using these patterns, some example rules are reported in the next subsections. These rules have been extracted from (i) clusters mined in time window #1 and from different geographical partitions, and (ii) clusters computed for different time windows from the UK partition. For the rule extraction, $support \geq 1\%$, and $lift > 1$ were enforced to prune both negatively correlated and uncorrelated item combinations.

Analysis on cities across different geographical areas. At first, the variation of people's interest across different locations and during a fixed time window was analyzed. The comparison was between the association rules mined from clusters computed in UK and USA cities, during time window #1 (datasets $\mathscr{D}_{(TW1,UK)}$ and $\mathscr{D}_{(TW1,USA)}$). Some sample rules modeling correlations in the tweet text content (class TC) are shown in Table 5.2, but the following discussion is based on the overall results.

People in the UK cities of Perth and Rugeley commented mostly on matches involving the England football team (e.g., rule $\boldsymbol{R}_1$), or other teams included in the same group as England. Moreover, an odd episode involving a single player was the main topic of various clusters ($\boldsymbol{R}_2$ highlights the popularity of the topic in the city of Rugeley, with a high lift value of 26.9). Instead, clusters from many cities of the USA reveal that people were interested in matches involving various football teams, also those not included in the same group as their national team. For instance, rule $\boldsymbol{R}_3$ refers to the interest of people from Whittier in the match between Italy and Costa Rica; according to rule $\boldsymbol{R}_4$, characterized by a lift of 7.16, people from Banning demonstrated a significant interest in the match involving Nigeria and Argentina.

The behaviour observed may be related to the people's different interests in the two geographical areas (UK and USA). Overall, football is more popular in UK than in USA, where people are mostly interested in other sports. While in UK people particularly focus on events related to their national teams, in USA they show a more general interest in the FIFA World Cup, also for events involving foreign teams.

Rule id	Partition	Rule	supp [%]	conf [%]	lift
R_1	UK	{uruguay} ⇒ {england} *centroid*(T = *2014-06-19*, L = *Perth*)	5.0	100	2.38
R_2	UK	{suarez,someone} ⇒ {bite} *centroid*(T = *2014-06-25*, L = *Rugeley*)	3.0	80	26.90
R_3	USA	{costa,rica} ⇒ {italy} *centroid*(T = *2014-06-20*, L = *Whittier,CA*)	8.3	64	1.67
R_4	USA	{nigeria} ⇒ {argentina} *centroid*(T = *2014-06-25*, L = *Banning,CA*)	2.1	53	7.16

Table 5.2: Example rules (class TC) characterizing clusters in UK and USA areas in time window #1 (datasets $\mathscr{D}_{(TW1,UK)}$ and $\mathscr{D}_{(TW1,USA)}$)

Analysis on cities across time windows. Likewise the previous analysis, but with a fixed geographical area, we analyse how the interests of people in different cities vary for events that occurred in different time windows. We compared rules mined from clusters computed in UK in the three time windows (datasets $\mathscr{D}_{(TW1,UK)}$, $\mathscr{D}_{(TW2,UK)}$, and $\mathscr{D}_{(TW3,UK)}$), adopting the same spatio-temporal data representation used before. Table 5.3 shows some example rules from the TC class, but the discussion is based on the overall results.

It is worth noting how interests varied after the elimination of England team which happened at the end of time window #1. The extracted rules show that people in UK shifted their attention to matches involving other teams. Various clusters in time window #2 are focused on the $Germany - Algeria$ football match (played on June 30th, 2014), and are mostly about the tactics (R_5 in the city of London) and performance (R_6 in the Scottish city of Stirling) of the German team.

During time window #3, the final match became one of the most popular topics (R_7 centered in the city of Newcastle). Nevertheless, the attention of people in UK also moved towards other topics loosely related to the competition. For instance, the latest transfer of player Luis Suarez away from an English club was mainly discussed on July 11th 2014, on the same day as the official announcement (R_8 centered in the city of London), while the next match of the England team, scheduled for November against Scotland (R_9 centered in the city of Broxbourne, near London), became popular just after the final World Cup match, on July 14th 2014.

Rule id	**Time window**	**Rule**	**supp [%]**	**conf [%]**	**lift**
R_1	1	{uruguay} ⇒ {england} *centroid*(T = *2014-06-19*, L = *Perth*)	5.0	100	2.38
R_2	1	{suarez, someone} ⇒ {bite} *centroid*(T = *2014-06-25*, L = *Rugeley*)	3.0	80	26.90
R_5	2	{line,high} ⇒ {germany} *centroid*(T = *2014-06-30*, L = *London*)	2.0	100	1.02
R_6	2	{good} ⇒ {germany} *centroid*(T = *2014-06-30*, L = *Stirling*)	2.0	58	1.22
R_7	3	{world, cup} ⇒ {final} *centroid*(T = *2014-07-13*, L = *Newcastle*)	10.2	99	2.91
R_8	3	{suarez} ⇒ {good,luck} *centroid*(T = *2014-07-11*, L = *London*)	2.3	77	24.40
R_9	3	{november} ⇒ {england,scotland} *centroid*(T = *2014-07-14*, L = *Broxbourne*)	1.8	100	36.71

Table 5.3: Example rules (class TC) characterizing clusters across the three time windows in UK cities (datasets $\mathscr{D}_{(TW1,UK)}$, $\mathscr{D}_{(TW2,UK)}$, $\mathscr{D}_{(TW3,UK)}$)

5.5 Computing performance of algorithm

The execution time for the cluster set computation on the six datasets in Table 5.1 spans from 12m 13s for the smallest dataset ($\mathscr{D}_{(TW2,UK)}$, 15,175 tweets) up to 33m 34s for the largest one ($\mathscr{D}_{(TW3,USA)}$, 50,028 tweets). The execution time for association rules extraction is less variable and has an overall mean value of 53s. Increasing the number of executors does not yield better performance in terms of clustering execution time due to the limited size of these datasets. Thus, experiments for these datasets were performed using one execution node.

The capacity of the clustering algorithm integrating the TASTE measure to scale up to bigger data collections was assessed by measuring the execution time when varying (i) the number of tweets under analysis and (ii) the number of parallel executors. For scalability analysis, to get a larger number of tweets including all (text, temporal, and spatial) features, we have considered the location specified in the user profile as reference location information. Indeed the amount of tweets with geo-coordinates is much less than the number of tweets with location information in the user profile due to the limitation of GPS enabled devices. Geo-coordinates for the location extracted from the user profile have been calculated using Bing Maps Locations API. The resulting dataset, named $\mathscr{D}''$, includes about 23.5 million tweets.

To study scalability by varying the number of tweets, we considered different sample rates of dataset $\mathscr{D}''$ and one executor for process running. Increasing the number of tweets from 50,000 to about 2.35 million (10% of whole $\mathscr{D}''$), we notice an increment of the execution time (from 33m 34s to 14h 31m). However, the growth rate of the execution time (about 25) is almost half the growth rate of the dataset size (about 47).

To study scalability by varying the number of executors, we considered the whole dataset $\mathscr{D}''$. The results show that, when increasing the number of executors from 4 to 8, the K-means algorithm integrating the TASTE measure scales almost linearly. The execution time is about 35h 43m with 4 nodes; it decreases to about 19h 24m with 6 nodes, and to 10h 45m with 8 nodes.

Thus, with a suitable number of parallel executors, the clustering task is capable to handle also bigger data, evenly distributing the load across the nodes. When fewer than 4 executors are used, the process exceeded 48 hours of execution and it was interrupted due to the very large dataset size.

5.6 Analytical comparison of spatio-temporal clustering methods

The proposed clustering approach has been tested through a theoretical and an analytical comparison with four previous studies on clustering Twitter data. These studies have proposed distance measures which combine the same tweet features considered in TASTE, or a subset of them. Specifically, the work in [127] takes into account the tweet spatial feature, while the spatio-temporal features are considered in [134], and both the text content and the spatial feature are evaluated in [128]. A first attempt in considering all the three tweet features was proposed in [136]. Like in TCharM, in these studies the geographic and temporal distances between tweets are computed using the Haversine and the Euclidean distance, respectively. The text content is represented with the *Bag-of-Words* (BOW) model [140] and a scoring scheme is adopted to weight the word relevance (i.e., the TF-IDF in [136] and the BursT in [128]); the cosine similarity is used to compare messages.

For each study we present the objective of the work and the methodology for clustering tweets, including the clustering algorithm, the distance functions used and the strategy adopted for combining tweet features. The distance measures proposed in these studies are summarized in Table 5.4. Then, we discuss the analytical comparison between these works and our approach.

The work in [127] aims at providing (near-)real time information to users about events happening close to them. Tweets are clustered through the K-means algorithm by considering their geographic distance. The discovered cluster set is then analysed to detect clusters that can reveal the occurrence of an event. Computed clusters are then filtered by comparing the spatial and temporal feature values of their tweets. If the number of tweets from a given cluster exceeds far from those from clusters found in vicinity in the past, the cluster is considered unusual and an event may happen there. For tweets included in unusual clusters, the text content is explored to extract representative keywords, which are sent to nearby users to inform them about the possible events.

The study in [134] focuses on discovering spatio-temporal periodic and aperiodic characteristics of events to support situation awareness. Tweets collections are analysed off-line with a DBSCAN based algorithm (GT-DBSCAN) to extract dense clusters of arbitrary shapes. The tweet text content is explored in a preprocessing phase to filter the subset of tweets relevant for the subsequent cluster analysis. Messages about specific events are selected by properly setting keywords for tweets search. To drive the clustering process, three distance measures, considering the tweet temporal and spatial features, are evaluated: (i) a temporal distance, (ii) a geographic distance, and (iii) a geographic-temporal distance, basically a combination of the two above. In this study we focus on the latter distance measure for performance comparison. The geographic-temporal distance is defined as the maximum value between the (normalized) geographic and temporal distances.

The work in [128] proposes a (near-)real time temporal-text clustering approach to detect bursts of tweets representing unexpectedly frequent occurrences of a certain topic in a short period of time. A sliding window of fixed time length is used to filter only the most recent tweets, which are then considered in the analysis. Selected tweets are clustered using the IncrementalDBSCAN algorithm [144], to detect dense clusters with shapes changing over time and to remove uninformative tweets (outliers). Clusters are calculated by evaluating the temporal-text distance between tweets. In [128], the temporal distance is used to module the text content distance by penalizing tweets

Study	Distance measure
Kim-11 [127]	$d_{Kim}(\tau_i, \tau_j) = d_s$
Arcaini-16 [134]	$d_{Arc}(\tau_i, \tau_j) = [\mathrm{Max}(d_s,d_t)]^{\beta}$, $\beta \in (0,1]$ d_s and d_t values expressed as the number of elementary units ϵ_s and ϵ_t, respectively
Lee-12 [128]	$d_{Lee}(\tau_i, \tau_j) = d_W \cdot \mathrm{e}^{\zeta d_t / M}$ M: time unit; ζ: exponential decay rate factor
Cunha-14 [136]	$d_{Cun}(\tau_i, \tau_j) = w_W \cdot d_W + w_t \cdot d_t + w_s \cdot d_s + w_{So} \cdot d_{So}$ $w_W, w_t, w_s, w_{So} \in [0,1]$ and $w_W + w_t + w_s + w_{So} = 1$
TCharM [120]	$d_{TASTE}(\tau_i, \tau_j) = d_W \cdot (k_s \cdot \mathrm{e}^{p_s \cdot d_s} + k_t \cdot \mathrm{e}^{p_t \cdot d_t})$ $k_s, k_t, p_s, p_t \in \mathbb{R}$; $k_s, k_t \in [0,1]$ and $k_s + k_t = 1$

Table 5.4: Distance measures for tweet comparison proposed in four reference previous studies and in TCharM. For a pair of tweets (τ_i, τ_j), their spatial distance $d_s(s_i, s_j)$ is shortly denoted by d_s, the temporal distance $d_t(t_i, t_j)$ by d_t, the content distance $d_W(W_i, W_j)$ by d_W, and the social distance $d_{So}(user_i, user_j)$ by d_{So}.

far distant in time. Finally, geo-spatial keywords are extracted from message in each computed cluster to estimate location of detected events.

The authors of [136] address the problem of identifying and displaying tweets profiles considering four different facets characterizing tweets: temporal, spatial, and context features and user social connections. Tweets are clustered with the DBSCAN algorithm [145] to detect arbitrarily shaped clusters and to remove outliers from the results. The adopted distance measure is a linear combination of the four considered tweet features, i.e., the distance on time, space, text content, and social relations. The social distance term evaluates the connections between users represented as nodes of a graph connected through edges. It is computed as the geodesic distance (i.e., the number of edges of the shortest path) between two nodes in the graph [146].

Based on the purposes of this analysis, we want to evaluate the ability of each distance measure above in discovering cohesive clusters of tweets to be represented through their centroids. Hence, keeping the K-means algorithm used in TCharM as a reference clustering method, we applied in turn each distance measure. Since we aim at discovering cohesive clusters considering temporal and spatial tweet features and text content, we omitted the social distance for the measure proposed in [136]. For the sake of brevity, the resulting clustering methods are denoted by Cunha-14 [136], Lee-12 [128], Arcaini-16 [134], and Kim-11 [127].

We evaluated the cluster cohesion as the average geographic/temporal/text content distance between tweets in the cluster and the cluster centroid. Lower values of these average distances point out a higher degree of cohesion on the corresponding tweet dimension.

The comparison was performed with the $\mathcal{D}_{(TW1,UK)}$ dataset. To produce comparable cluster sets, we forced K=200 as expected number of clusters for all the distance measures (i.e., the same value selected in Section 5.4.2). We suitably tuned the parameters to use each distance measure at its best with the $\mathscr{D}_{(TW1,UK)}$ datasets and with the K-means algorithm. Starting from the configuration proposed in each study (considered as default configuration), we performed several runs to tune the parameters of each distance measure, with the aim of reducing the average cluster SSE as well as the distance values for all the tweet dimensions they consider. Selected parameter values are reported in Figure 5.3. For TCHARM we used the configuration specified in Section 5.4.2.

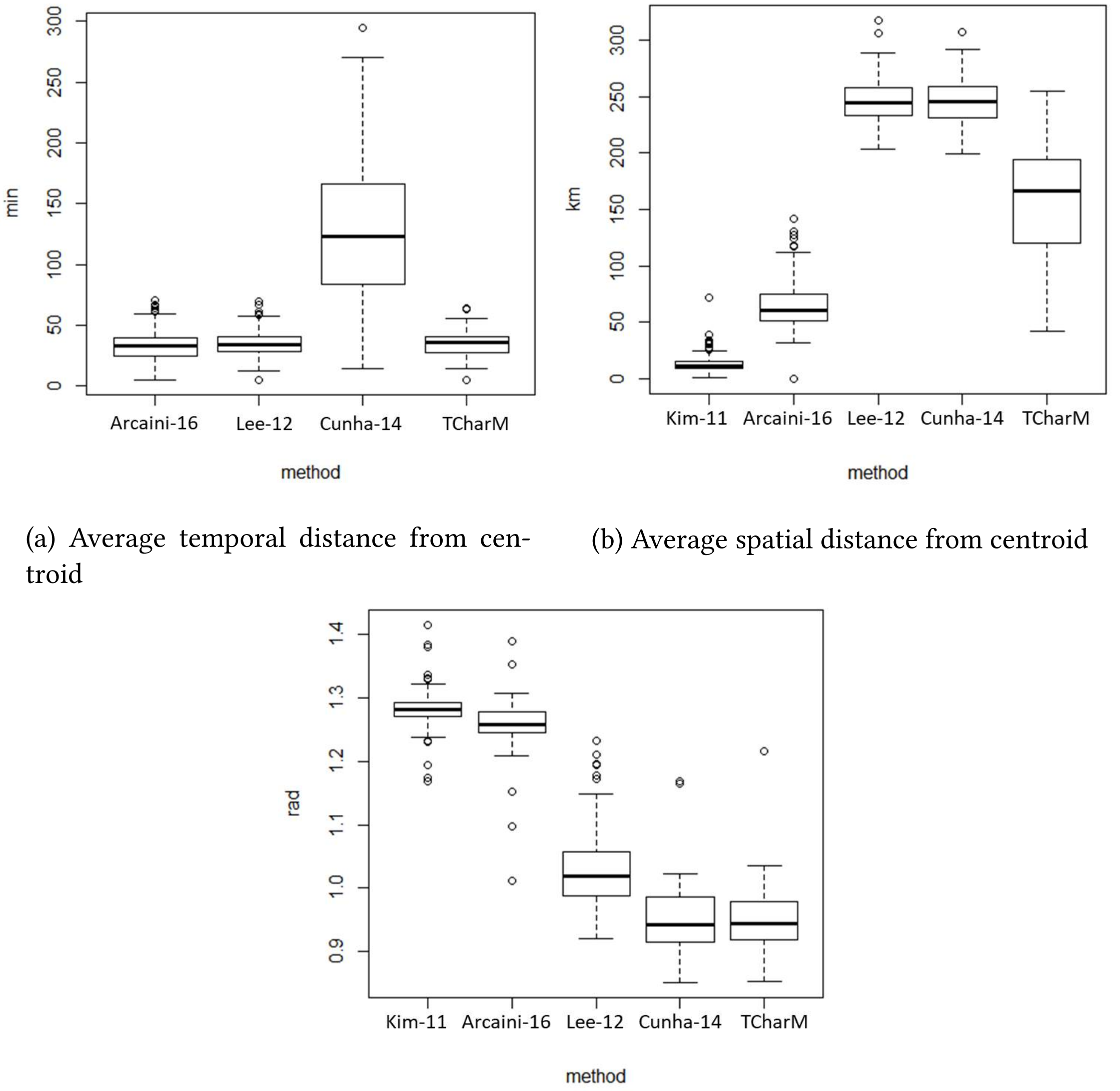
For each method, box plots in Figure 5.3 illustrate the distributions of the average geographic/temporal/text content distance between tweets in each cluster and cluster centroid, while Table 5.5 reports the average values. Note that the temporal box plot for the Kim-11's measure is not represented in Figure 5.3 as its values are too high compared to the other methods.

Method	**Avg time distance (min)**	**Avg GPS distance (km)**	**Avg text content distance (rad)**
Kim-11	3905	**14**	1.28
Arcaini-16	**33**	66	1.26
Lee-12	**35**	246	**1.03**
Cunha-14	126	245	**0.95**
TCHARM	**35**	158	**0.95**

Table 5.5: Average value of mean temporal, spatial, and text content distances between tweets and their centroids for each distance measure.

Clusters with the highest text cohesion are computed with TCHARM, Cunha-14 and Lee-12 distance measures, which provide comparable results. Clusters with the highest temporal cohesion are provided by Arcaini-16, TCHARM and Lee-12, which achieve similar performance. The highest spatial cohesion is given by Kim-11, followed by Arcaini-16, and then TCHARM. These results point out that TCHARM provides clusters with an overall good cohesion on all the three facets characterizing tweets. Computed clusters show the highest cohesion on the text content and on the temporal feature, and the third best spatial cohesion. Yet it should be noted that, when setting parameters in TASTE, we gave more importance to the temporal cohesion than to the spatial one.

Clusters provided by Arcaini-16, Lee-12, and Kim-11 methods show a good cohesion on the tweet features considered in their proposed distance measures, but the cohesion on the remaining features is far lower than in TCHARM. Clusters tend to be spread over a larger geographic area (Lee-12) or a longer time period (Kim-11), or to discuss more different topics (Kim-11, Arcaini-16). These results demonstrate that, to obtain clusters suitable for a subsequent characterization of their spatial, temporal and text features, it is convenient to consider all the three dimensions directly in the clustering phase. Otherwise, further post-processing steps would be required to characterize the clusters



(a) Average temporal distance from centroid

(b) Average spatial distance from centroid

(c) Average text content distance from centroid

Figure 5.3: Distributions of the average temporal, spatial, text content distances from cluster centroids, for each method. The temporal box plot for Kim-11 is not represented as its values are too high. Parameter configurations are as follows. Arcaini-16: $\epsilon_s = 2km$, $\epsilon_t = 1200s$, $\beta = 1$; Cunha-14: $w_s = w_W = 0.25, w_t = 0.5, w_{So} = 0$; Lee-12: $\zeta/M = 12h^{-1}$.

with the features previously left out.

On the other hand, when all three tweet features are considered to cluster tweets, their contributions should be properly weighted in the distance measure. A liner combination of the content, spatial, and temporal distances as the one proposed in Cunha-14 turns out to be less suitable than our approach since discovered clusters manifest a temporal and spatial cohesion lower than in TCharM.

To deepen into the comparison of the methods above, we used the Adjusted Rand Index (ARI) [147] to evaluate the agreement between the cluster sets generated using the TASTE measure and those obtained with the other distance measures. The ARI allows a more accurate estimation of the agreement between two partitions than the standard Rand Index [148]. Basically, it rescales the Rand Index value with respect to its expected value for two independent clustering algorithms. ARI has a maximum value of 1 for two identical partitions, and an expected value of 0 for two independent random partitions. Higher ARI values imply higher levels of agreement between two partitions.

The computed values of ARI report a moderate agreement between the cluster set provided by TCharM and the one computed by Cunha-14 (ARI = 0.45). The agreement decreases with Lee-11 (ARI = 0.13), Arcaini-16 (ARI = 0.03), and Kim-11 (ARI = 0.005) methods which consider a subset of tweet features.

5.7 Discussion

This section provides a discussion on the experimental results and the differences and similarities with previous studies.

Discovery of cohesive spatio-temporal clusters. The experimental findings demonstrate the ability of the proposed methodology to properly analyse large tweet collections distributed over time and space as well as addressing various topics for automatically computing cohesive clusters. TCharM allows data miners to discover clusters useful for identifying what are the most significant topics for users in different cities and times. The 2014 FIFA World Cup use case considered in this study enables a thorough evidence-based validation of the computed clusters due to the availability of a time schedule for the main events (e.g., football matches) and web news about the other events or celebrities somehow involved. Mined clusters are centered in time in correspondence with an event related to the 2014 FIFA World Cup and they mainly include messages about the event. Moreover, the clusters present a good spatio-temporal cohesion around their centroid, as parameters of the TASTE measure were conveniently tuned to best fit the target spatial and temporal granularities.

Comparison with previous studies on tweet clustering. Based on the comparison reported in Section 5.6, the proposed methodology is more suitable than previous studies to discover clusters with a good level of cohesion balanced on the three facets characterizing tweets. This result is due to two main properties of our distance measure. On the one side, all three (space, time and text content) tweet features are weighted to drive the clustering task. On the other side, in TASTE the distances on time and space are suitably applied to penalize the text distance, based on the hypothesis that a tight temporal and spatial proximity can contribute in detecting clusters of tweets about the same topic. Previous studies [127, 134, 128] compute clusters with a good cohesion on the tweet features considered in their proposed distance measures, but the cohesion on the remaining features is far lower than in TCharM. Moreover, a linear combination of the three tweet features, like the one proposed by Cunha, Soares, and Mendes Rodrigues [136], is less effective than the TASTE measure in providing a good cluster cohesion on all dimensions.

From a temporal perspective, clusters computed with other methods can have a higher temporal span than in TCharM. Indeed, while our clusters are centered around events of interest, we observed that clusters of other approaches [127, 136] can include tweets discussing about more than one event (e.g., more football matches). Similarly, clusters can have a lower spatial cohesion than in TCharM [128], and thus they include tweets spread across a larger geographical area. The two aspects above prevent from performing qualitative analyses based on fine-grained temporal and spatial resolutions. Finally, the lower text similarity among tweets in clusters [134] makes it difficult to associate a single prevailing topic with each cluster and to generate significant association rules that can concisely describe the cluster content (i.e., rules with high values of quality indices such as support, confidence and lift). It follows that, with the adoption of other distance measures than TASTE, a further level of segmentation would be required to identify the main topics in each cluster, or to partition the cluster content into subsets which refer to shorter time windows and more limited geographic areas.

Cluster content characterization through rules analysis. The proposed cluster characterization allows data miners to better understand popular topics in various urban areas and over different time windows. Indeed, association rules represent the mined knowledge in a concise and easily understandable form. Rule analysis pointed out some reactions that were in some cases predictable, but it also highlighted some popular topics not so evident a priori, like some celebrities' public statements. The same kind of characterization can be used also to study urban topics and the reactions and interests of people from different cities.

Computing performance. From a computational point of view, TCharM has a major advantage with respect to related works, since it is implemented on Apache Spark and can distribute computational load across parallel executors. Tests performed on big collections of tweets (Section 5.5) prove the good scalability of our platform, in particular of the clustering algorithm. Thus, TCharM can be also applied to use cases with a higher cardinality of data and it is still capable to provide results in a reasonable time.

Exploitation of the mined knowledge. The described findings provide a spatio-temporal overview of people involvement in occurred events. The proposed methodology can effectively enable a deep analysis of spatio-temporal trends on social networks, showing when and where certain topics spread among users. This knowledge, hidden in Twitter data collections, can have a variety of practical applications in different domains. In a smart urban environment, for example, social networks are currently recognized as powerful instruments to enable citizen interaction and participation. Citizens may use Twitter to report information related to a variety of aspects such as urban safety, traffic, and services (e.g., bike sharing or public transport offer). City administration is interested in better understanding where and when citizens report issues about the above aspects, to eventually undertake appropriate and targeted responses to citizens' concerns. The application of TCharM to such collections of tweets would help to find out in which areas of the city and in which periods of time citizens discuss and complain about some issues. Clustering analysis would extract spatio-temporally defined clusters of topics reported by citizens. Rule analysis would then better highlight the degrees of correlation among topics, times and places of discussion and describe how the same topics evolve across different periods and through nearby urban areas.

Future development of TCharM**.** TCharM can be further improved through a semantic modeling of concepts expressed by words and tweets. As an example, the use of *Latent Semantic Analysis* (LSA) [149] can provide higher quality results (e.g., in terms of support, confidence and lift values of association rules). Since TF-IDF vectors tend to be too long, LSA uses the *Singular Value Decomposition* (SVD) technique to reduce their dimensionality. The representation of documents is approximated by (i) filtering out noisy terms (e.g., words that express the same concept are condensed into a unique term) and (ii) keeping the smallest set of relevant terms that represent all the concepts included in the documents. Therefore, by producing a set of concepts related to the documents and terms, LSA can better manage the semantic connection among words and between words and topics. This technique would improve the results of TCharM, providing clusters of tweets with a higher cohesion in the text dimension, not only with respect to individual words, but also to entire concepts.

Chapter 6

Conclusion

The research activity described in this thesis was aimed at mining useful information from huge amounts of data related to the urban context, for different kinds of applications. Heterogeneous data collected from manifold sources have been analysed to obtain significant patterns and insightful descriptions of different urban scenarios, and specifically about energy consumption and people's interests. The performed analyses can effectively support the enhancement of urban services, like the efficient provisioning of energy for the heating of residential buildings, also through a correct understanding of the perception of citizens about such services and other related topics.

In Chapter 2 the MuSTLE framework is proposed to automatize and speed-up the analysis of huge amounts of heterogeneous urban data. Its implementation is based on the distributed database MongoDB and on in-memory computations that exploit the MapReduce paradigm, to reach a linear scalability. The developed engine aggregates data on the fly, exploring multiple combinations of space and time granularities. Explorative analyses are performed on real heterogeneous data to explore and discover more patterns of data and to discover important relationships among different urban features.

MuSTLE may be enriched with advanced data mining algorithms and additional data types. For example, geo-referenced user feedbacks provided through mobile devices or extracted from posts on social networks, to analyse the citizen perception of the urban area, similarly to what is proposed in Chapter 5. To fully support this kind of analysis, MuSTLE shall be extended by implementing the computation of K-means clustering with the MapReduce paradigm.

In Chapter 3, research on energy data led to the development of new methodologies for the estimation of buildings heating energy demand, according to both real consumptions (operational rating) or to physical and structural properties (asset rating). For operational rating, different working phases of the heating systems have been modeled with different algorithms. A system for the operational energy rating of buildings, based on the MuSTLE framework, has been designed to collect and store data from different sources. Two different platforms based on a common architecture and extended according to the characteristics of the analysis have been implemented: DA-BOR for descriptive analytics and PA-BOR for predictive analytics.

In Section 3.4, descriptive data analytics are used to compute different classes of KPIs about thermal energy efficiency of buildings, included the analysis of energy signature.

In Section 3.5, the system is extended to integrate an advanced analytics algorithm for the prediction of heating energy consumption.

Experimental results on real data show the effectiveness and the efficiency of the system in exploiting energy signature analysis to evaluate and rank building efficiency and energy performance over time and to forecast the power demand with a limited error. Moreover, the analysis of computational performance of the proposed algorithms demonstrate that the system can easily scale up to bigger data sets.

For asset rating, different classes of buildings have been separately analysed to extract the main features characterizing their energy demand.

In Chapter 4, a methodology for the analysis of Energy Performance Certificates (EPCs) datasets, called HEDEBAR, is proposed with a twofold purpose: (i) *predictive*, as it defines models the estimation of buildings energy demand; and (ii) *descriptive*, as it highlights the main features that determine the energy demand for various classes of building. Experimental results show that HEDEBAR is able to compute a value for thermal energy demand with a reasonable error and using a smaller set of input variables than the one of EPC.

A possible future development of the presented work is the validation of the proposed HEDEBAR methodology on other data sets of EPCs collected in other cities and issued through a different certification system. This can help to further validate the methodology. Moreover, it makes possible to compare the information extracted from EPCs issued in different cities. Moreover, by further reducing the number of required features, we aim at applying such methodology also to other buildings for which the structural properties are known.

In Chapter 5, the characterization of the urban scenario has been enriched with feedbacks of citizens collected from social media. An extensible framework for the characterization of significant clusters of users and topics has been developed. Such framework makes possible to incorporate and compare different clustering algorithms and distance measures, and to select the one providing more cohesive clusters.

In particular, a novel exploratory data mining methodology to analyse Twitter datasets, called TCharM, is proposed. Its aim is to discover significant and cohesive groups of tweets by considering three facets of Twitter data: spatial, temporal, and text content information. The experimental validation demonstrated the ability of TCharM in efficiently characterizing collections of tweets in terms of distribution of people involvement, topic identification, and correlations among tweet features. As a matter of fact, we managed to isolate groups of tweets focused on a few topics, temporarily associated to actual events and posted from a limited geographical area. Compared with other approaches for tweet clustering, clusters computed using the TASTE measure confirmed an overall better cohesion balanced between the three tweet features.

Possible future research directions concern devising a novel clustering distance measure to consider also the *user information* (such as user characteristics and social relationships) in the cluster analysis. This additional information would be very helpful to discover spatio-temporal patterns of communities of users and to better profile how the user interests evolve over time.

Moreover, TCharM can be applied for the (near-)real time analysis, for instance of tweets collected every hour, to investigate the spatial evolution of clusters and related topics with a low time granularity. This approach would provide a deeper overview of the spatio-temporal dynamics of people's interests. Thanks to the deployment on a cloud-based platform as Apache Spark, TCharM can analyse huge amounts of tweets providing results in a reasonable time consistent with a (near-)real time approach.

The algorithms and data mining architectures proposed in this research activity proved to be effective solutions to get useful knowledge from heterogeneous data in complex urban application domains. The described results confirm the importance of analyzing data with suitable granularity levels, in order to extract patterns and relationships among variables that are significant for the purposes of the analysis. The proposed architectures, based on MapReduce paradigm and on distributed cluster computing, demonstrated the ability to analyse huge data collection with a good scalability.

Overall, an important future work would be the extension and integration of the proposed architectures to support other types of data and applications in the urban context.

Bibliography

[1] Y. Zheng et al. "Urban Computing: Concepts, Methodologies, and Applications". In: *ACM Trans. Intell. Syst. Technol.* 5.3 (Sept. 2014), 38:1–38:55. ISSN: 2157-6904.

[2] A. Oussous et al. "Big Data technologies: A survey". In: *Journal of King Saud University - Computer and Information Sciences* (2017). ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2017.06.001. URL: http://www.sciencedirect.com/science/article/pii/S1319157817300034.

[3] IEA. "Energy Efficiency Indicators". In: (2014). DOI: http://dx.doi.org/10.1787/9789264215672-en. URL: /content/book/9789264215672-en.

[4] A. Attanasio et al. "Fast and Effective Decision Support for Crisis Management by the Analysis of People's Reactions Collected from Twitter". In: *New Trends in Databases and Information Systems.* Ed. by Tadeusz Morzy, Patrick Valduriez, and Ladjel Bellatreche. Cham: Springer International Publishing, 2015, pp. 229–234. ISBN: 978-3-319-23201-0.

[5] MongoDB Inc. *MongoDB Documentation.* 2016. URL: https://docs.mongodb.org/v2.6/MongoDB-manual-v2.6.pdf (visited on 05/31/2016).

[6] K. Chodorow and M. Dirolf. *MongoDB: the definitive guide.* O'Reilly Media, 2010.

[7] J. Dean and S. Ghemawat. "MapReduce: Simplified Data Processing on Large Clusters". In: *Commun. ACM* 51.1 (Jan. 2008), pp. 107–113. ISSN: 0001-0782. DOI: 10.1145/1327452.1327492. URL: http://doi.acm.org/10.1145/1327452.1327492.

[8] A. Attanasio, T. Cerquitelli, and S. Chiusano. "Supporting the analysis of urban data through NoSQL technologies". In: *2016 7th International Conference on Information, Intelligence, Systems Applications (IISA).* July 2016, pp. 1–6. DOI: 10.1109/IISA.2016.7785334.

[9] *Arpa Piemonte.* 2016. URL: https://www.arpa.piemonte.gov.it (visited on 05/31/2016).

[10] Sistema Piemonte. *Air quality in Piemonte.* 2016. URL: http://www.sistemapiemonte.it/ambiente/srqa/conoscidati.shtml (visited on 05/31/2016).

[11] Weather Underground. *Weather Underground web service.* 2016. URL: http://www.wunderground.com (visited on 05/31/2016).

[12] Regione Piemonte. *Smartdatanet*. 2016. URL: http://smartdatanet.it (visited on 05/31/2016).

[13] A. Acquaviva et al. “Enhancing Energy Awareness Through the Analysis of Thermal Energy Consumption.” In: *EDBT/ICDT Workshops*. 2015, pp. 64–71.

[14] G. Re Calegari, I. Celino, and D. Peroni. “City data dating: Emerging affinities between diverse urban datasets”. In: *Information Systems* 57 (2016), pp. 223–240. ISSN: 0306-4379. DOI: http://dx.doi.org/10.1016/j.is.2015.08.001. URL: http://www.sciencedirect.com/science/article/pii/S0306437915001362.

[15] R. O. Sinnott et al. “The Urban Data Re-use and Integration Platform for Australia: Design, Realisation, and Case Studies”. In: *2015 IEEE International Conference on Information Reuse and Integration*. Aug. 2015, pp. 90–97. DOI: 10.1109/IRI.2015.24.

[16] K. Kolomvatsos, C. Anagnostopoulos, and S. Hadjiefthymiades. “An efficient environmental monitoring system adopting data fusion, prediction, fuzzy logic”. In: *Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on*. July 2015, pp. 1–6. DOI: 10.1109/IISA.2015.7388070.

[17] V. Marinakis et al. “Advanced ICT platform for real-time monitoring and infrastructure efficiency at the city level”. In: *Information, Intelligence, Systems and Applications (IISA), 2015 6th International Conference on*. July 2015, pp. 1–5. DOI: 10.1109/IISA.2015.7387958.

[18] A. J. Jara, D. Genoud, and Y. Bocchi. “Short paper: Sensors data fusion for Smart Cities with KNIME: A real experience in the SmartSantander Testbed”. In: *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. Mar. 2014, pp. 173–174. DOI: 10.1109/WF-IoT.2014.6803145.

[19] A. Noulas, C. Mascolo, and E. Frias-Martinez. “Exploiting Foursquare and Cellular Data to Infer User Activity in Urban Environments”. In: *Proceedings of the 2013 IEEE 14th International Conference on Mobile Data Management - Volume 01*. MDM ’13. Washington, DC, USA: IEEE Computer Society, 2013, pp. 167–176. ISBN: 978-0-7695-4973-6. DOI: 10.1109/MDM.2013.27. URL: http://dx.doi.org/10.1109/MDM.2013.27.

[20] G. Cardone et al. “The participact mobile crowd sensing living lab: The testbed for smart cities”. In: *IEEE Communications Magazine* 52.10 (Oct. 2014), pp. 78–85. ISSN: 0163-6804. DOI: 10.1109/MCOM.2014.6917406.

[21] S. Mirri et al. “On Combining Crowdsourcing, Sensing and Open Data for an Accessible Smart City”. In: *2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies*. Sept. 2014, pp. 294–299. DOI: 10.1109/NGMAST.2014.59.

[22] R. Szabó et al. "Framework for smart city applications based on participatory sensing". In: *2013 IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom)*. Dec. 2013, pp. 295–300. DOI: 10.1109/CogInfoCom.2013.6719260.

[23] E. S. Page. "Continuous Inspection Schemes". In: *Biometrika* 41.1-2 (1954), pp. 100–115. DOI: 10.1093/biomet/41.1-2.100. URL: http://dx.doi.org/10.1093/biomet/41.1-2.100.

[24] J. Manyika and H. Durrant-Whyte. *Data Fusion and Sensor Management: A Decentralized Information-Theoretic Approach*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1995. ISBN: 0133031322.

[25] X. Yao, L. Peng, and T. Chi. "A Spatio-Temporal Geocoding Model for Vector Data Integration". In: *Geo-Informatics in Resource Management and Sustainable Ecosystem*. Ed. by Fuling Bian and Yichun Xie. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 566–577. ISBN: 978-3-662-49155-3.

[26] S. Batasova et al. "Preparation of distributed heterogeneous data for data mining". In: *2015 XVIII International Conference on Soft Computing and Measurements (SCM)*. May 2015, pp. 205–207. DOI: 10.1109/SCM.2015.7190457.

[27] Z. Khan, A. Anjum, and S. L. Kiani. "Cloud Based Big Data Analytics for Smart Future Cities". In: *2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*. Dec. 2013, pp. 381–386. DOI: 10.1109/UCC.2013.77.

[28] C. Jacobs-Crisioni, P. Rietveld, and E. Koomen. "The impact of spatial aggregation on urban development analyses". In: *Applied Geography* 47 (2014), pp. 46–56. ISSN: 0143-6228. DOI: http://dx.doi.org/10.1016/j.apgeog.2013.11.014. URL: http://www.sciencedirect.com/science/article/pii/S0143622813002774.

[29] A. Páez and Darren M. Scott. "Spatial statistics for urban analysis: A review of techniques with examples". In: *GeoJournal* 61.1 (Sept. 2004), pp. 53–67. ISSN: 1572-9893. DOI: 10.1007/s10708-005-0877-5. URL: https://doi.org/10.1007/s10708-005-0877-5.

[30] L. Hu and N. A. Brunsell. "The impact of temporal aggregation of land surface temperature data for surface urban heat island (SUHI) monitoring". In: *Remote Sensing of Environment* 134 (2013), pp. 162–174. ISSN: 0034-4257. DOI: http://dx.doi.org/10.1016/j.rse.2013.02.022. URL: http://www.sciencedirect.com/science/article/pii/S0034425713000631.

[31] L. Gómez E. Estrada-Guzman R. Maciel. "NoSQL method for the metric analysis of Smart Cities". In: *IEEE Guadalajara GDL CCD White Papers* (2015).

[32] S. Sathya and M. Victor Jose. “Application of Hadoop MapReduce technique to Virtual Database system design”. In: *Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on*. Mar. 2011, pp. 892–896. DOI: 10.1109/ICETECT.2011.5760245.

[33] G. ElSheikh et al. “SODIM: Service Oriented Data Integration based on MapReduce”. In: *Alexandria Engineering Journal* 52.3 (2013), pp. 313–318. ISSN: 1110-0168. DOI: https://doi.org/10.1016/j.aej.2013.02.007. URL: http://www.sciencedirect.com/science/article/pii/S111001681300029X.

[34] L. Xu, K. Jin, and H. Tian. “MRData: A MapReduce-Based Tool for Heterogeneous Data Integration”. In: *Proceedings of the 2010 International Conference of Information Science and Management Engineering - Volume 02*. ISME ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 269–272. ISBN: 978-0-7695-4132-7. DOI: 10.1109/ISME.2010.252. URL: http://dx.doi.org/10.1109/ISME.2010.252.

[35] K. Pearson. “Note on regression and inheritance in the case of two parents”. In: *Proceedings of the Royal Society of London* 58.347-352 (1895), pp. 240–242.

[36] D.M. et al. Lane. *Online Statistics: An Interactive Multimedia Course of Study*. Ed. by University of Houston Clear Lake Rice University and Tufts University. URL: http://onlinestatbook.com (visited on 05/31/2016).

[37] X. Yan and X. Su. *Linear regression analysis: theory and computing*. World Scientific, 2009.

[38] J. F. Kenney and Keeping E. S. “Linear Regression and Correlation”. In: *Mathematics of Statistics, Pt. 1, 3rd ed.* Princeton, NJ: Van Nostrand, 1962. Chap. 15, pp. 252–285.

[39] H. Sakoe and S. Chiba. “Dynamic programming algorithm optimization for spoken word recognition”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26.1 (Feb. 1978), pp. 43–49. ISSN: 0096-3518. DOI: 10.1109/TASSP.1978.1163055.

[40] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1988. ISBN: 0-934613-73-7.

[41] A. Acquaviva et al. “Energy Signature Analysis: Knowledge at Your Fingertips”. In: *2015 IEEE International Congress on Big Data*. June 2015, pp. 543–550. DOI: 10.1109/BigDataCongress.2015.85.

[42] Weather Underground. *Weather Underground web service*. 2016. URL: http://api.wunderground.com (visited on 07/30/2016).

[43] S. Karnouskos. “The cooperative Internet of Things enabled Smart Grid”. In: *Proc. of IEEE ISCE2010*. 2009.

[44] C. Warmer et al. “Web services for integration of smart houses in the smart grid”. In: *Grid-Interop*. 2009.

[45] D. Guinard et al. “Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services”. In: *IEEE Trans. on Services Computing* 3.3 (2010).

[46] G. Candido et al. “Service-Oriented Infrastructure to Support the Deployment of Evolvable Production Systems”. In: *IEEE Trans. on Industrial Informatics*. Nov. 2009.

[47] G. Candido et al. “Generic Management Services for DPWS-enabled devices”. In: *Proc. of IEEE IECON*. 2009.

[48] E. Patti et al. “Event-Driven User-Centric Middleware for Energy-Efficient Buildings and Public Spaces”. In: *IEEE Systems Journal* (2014).

[49] C. Warmer et al. “Web services for integration of smart houses in the smart grid”. In: 2009.

[50] U. Fischer et al. “Real-Time Business Intelligence in the MIRABEL Smart Grid System”. In: *Enabling Real-Time Business Intelligence*. Ed. by Malu Castellanos, Umeshwar Dayal, and Elke A. Rundensteiner. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–22. ISBN: 978-3-642-39872-8.

[51] L. Siksnys, C. Thomsen, and T. B. Pedersen. “MIRABEL DW: Managing Complex Energy Data in a Smart Grid”. In: *Data Warehousing and Knowledge Discovery*. Ed. by Alfredo Cuzzocrea and Umeshwar Dayal. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 443–457. ISBN: 978-3-642-32584-7.

[52] D. Wijayasekara et al. “Mining Building Energy Management System Data Using Fuzzy Anomaly Detection and Linguistic Descriptions”. In: *Industrial Informatics, IEEE Transactions on* 10.3 (Aug. 2014), pp. 1829–1840. ISSN: 1551-3203. DOI: 10.1109/TII.2014.2328291.

[53] J.S. van der Veen, B. van der Waaij, and R.J. Meijer. “Sensor Data Storage Performance: SQL or NoSQL, Physical or Virtual”. In: *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. June 2012, pp. 431–438. DOI: 10.1109/CLOUD.2012.18.

[54] L. Šikšnys, C. Thomsen, and T. B. Pedersen. “MIRABEL DW: Managing Complex Energy Data in a Smart Grid”. In: *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXI: Selected Papers from DaWaK 2012*. Ed. by Abdelkader Hameurlain et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 48–72. ISBN: 978-3-662-47804-2. DOI: 10.1007/978-3-662-47804-2_3. URL: https://doi.org/10.1007/978-3-662-47804-2_3.

[55] O. Ardakanian et al. “Computing Electricity Consumption Profiles from Household Smart Meter Data”. In: *EDBT/ICDT Workshops’14*. 2014, pp. 140–147.

[56] S.S.S.R. Depuru et al. "A hybrid neural network model and encoding technique for enhanced classification of energy consumption data". In: *Power and Energy Society General Meeting, 2011 IEEE*. July 2011, pp. 1–8. DOI: 10.1109/PES.2011.6039050.

[57] C. Filippín and S. Flores Larsen. "Analysis of energy consumption patterns in multi-family housing in a moderate cold climate". In: *Energy Policy* 37.9 (2009). New Zealand Energy Strategy, pp. 3489–3501. ISSN: 0301-4215.

[58] F. H. Zulkernine et al. "Towards Cloud-Based Analytics-as-a-Service (CLAaaS) for Big Data Analytics in the Cloud". In: *IEEE International Congress on Big Data, BigData Congress 2013, June 27 2013-July 2, 2013*. 2013, pp. 62–69.

[59] J. Z. Kolter and J. Ferreira. "A Large-Scale Study on Predicting and Contextualizing Building Energy Usage". In: *Twenty-Fifth AAAI Conference on Artificial Intelligence*. 2011.

[60] D. Anjos, P. Carreira, and A. P. Francisco. "Real-Time Integration of Building Energy Data". In: *2014 IEEE International Congress on Big Data, Anchorage, AK, USA, June 27 - July 2, 2014*. 2014, pp. 250–257.

[61] C. Wang, M. de Groot, and P. Marendy. "A Service-Oriented System for Optimizing Residential Energy Use". In: *IEEE International Conference on Web Services, ICWS 2009, Los Angeles, CA, USA, 6-10 July 2009*. IEEE, 2009, pp. 735–742.

[62] R. E. Edwards, J. New, and L. E. Parker. "Predicting future hourly residential electrical consumption: A machine learning case study". In: *Energy and Buildings* 49 (2012), pp. 591–603. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2012.03.010. URL: http://www.sciencedirect.com/science/article/pii/S0378778812001582.

[63] S. Lu, Y. Liu, and D. Meng. "Towards a Collaborative Simulation Platform for Renewable Energy Systems". In: *IEEE Ninth World Congress on Services, SERVICES 2013, Santa Clara, CA, USA, June 28 - July 3, 2013*. IEEE Computer Society, 2013, pp. 9–12.

[64] L. Garcia Rios and J. A. Incera Diguez. "Big Data Infrastructure for analyzing data generated by Wireless Sensor Networks". In: *2014 IEEE International Congress on Big Data, Anchorage, AK, USA, June 27 - July 2, 2014*. 2014, pp. 816–823.

[65] L. Belussi and L. Danza. "Method for the prediction of malfunctions of buildings through real energy consumption analysis: Holistic and multidisciplinary approach of Energy Signature". In: *Energy and Buildings* 55 (2012), pp. 715–720.

[66] C. Ghiaus. "Experimental estimation of building energy performance by robust regression". In: *Energy and buildings* 38.6 (2006), pp. 582–587.

[67] Y. Liu et al. “Data-driven based model for flow prediction of steam system in steel industry”. In: *Information Sciences* 193 (2012), pp. 104–114. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2011.12.031. URL: http://www.sciencedirect.com/science/article/pii/S0020025512000102.

[68] K. Li, H. Su, and J. Chu. “Forecasting building energy consumption using neural networks and hybrid neuro-fuzzy system: A comparative study”. In: *Energy and Buildings* 43.10 (2011), pp. 2893–2899. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2011.07.010. URL: http://www.sciencedirect.com/science/article/pii/S0378778811003124.

[69] H. R. Khosravani et al. “A Comparison of Energy Consumption Prediction Models Based on Neural Networks of a Bioclimatic Building”. In: *Energies* 9.1 (2016). ISSN: 1996-1073. DOI: 10.3390/en9010057. URL: http://www.mdpi.com/1996-1073/9/1/57.

[70] G. Dhiman, K. Mihic, and T. Rosing. “A system for online power prediction in virtualized environments using gaussian mixture models”. In: *Design Automation Conference*. June 2010, pp. 807–812. DOI: 10.1145/1837274.1837478.

[71] B. Dong, C. Cao, and S. E. Lee. “Applying support vector machines to predict building energy consumption in tropical region”. In: *Energy and Buildings* 37.5 (2005), pp. 545–553. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2004.09.009. URL: http://www.sciencedirect.com/science/article/pii/S0378778804002981.

[72] M. Domínguez et al. “Dimensionality reduction techniques to analyze heating systems in buildings”. In: *Information Sciences* 294 (2015). Innovative Applications of Artificial Neural Networks in Engineering, pp. 553–564. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2014.06.029. URL: http://www.sciencedirect.com/science/article/pii/S0020025514006574.

[73] J. Sjögren, S. Andersson, and T. Olofsson. “Sensitivity of the total heat loss coefficient determined by the energy signature approach to different time periods and gained energy”. In: *Energy and Buildings* 41.7 (2009), pp. 801–808.

[74] S. Andersson et al. “Building performance based on measured data”. In: *World Renewable Energy Congress-Sweden; 8-13 May; 2011; Linköping; Sweden*. 057. Linköping University Electronic Press. 2011, pp. 899–906.

[75] S Danov et al. “Approaches to evaluate building energy performance from daily consumption data considering dynamic and solar gain effects”. In: *Energy and Buildings* 57 (2013), pp. 110–118.

[76] E. Mangematin, G. Pandraud, and D. Roux. “Quick measurements of energy efficiency of buildings”. In: *Comptes Rendus Physique* 13.4 (2012). Science of nuclear safety post-Fukushima, pp. 383–390.

[77] A. Bogomolov et al. “Energy consumption prediction using people dynamics derived from cellular network data”. In: *EPJ Data Science* 5.1 (Mar. 2016), p. 13. ISSN: 2193-1127. DOI: 10.1140/epjds/s13688-016-0075-3. URL: https://doi.org/10.1140/epjds/s13688-016-0075-3.

[78] Turin GeoPortal. *Available at http://www.comune.torino.it/geoportale/ Last access: March 2015.*

[79] R. T. Fielding and R. N. Taylor. *Architectural styles and the design of network-based software architectures.* Vol. 7. University of California, Irvine Doctoral dissertation, 2000.

[80] Patrick Th. Eugster et al. “The many faces of publish/subscribe”. In: *ACM CSUR* (June 2003).

[81] E. Patti et al. “Distributed Software Infrastructure for General Purpose Services in Smart Grid”. In: *IEEE Trans. on Smart Grid* (2014), pp. 1–8.

[82] R. Kimball and M. Ross. *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling.* 2nd. New York, NY, USA: John Wiley & Sons, Inc., 2002. ISBN: 0471200247, 9780471200246.

[83] J. Vesterberg, S. Andersson, and T. Olofsson. “Robustness of a regression approach, aimed for calibration of whole building energy simulation tools”. In: *Energy and Buildings* 81.0 (2014), pp. 430–434.

[84] D. Lane. “Online statistics education: A multimedia course of study”. In: *EdMedia: World Conference on Educational Media and Technology.* Association for the Advancement of Computing in Education (AACE). 2003. Chap. XIV. Regression, pp. 1317–1320.

[85] M. Zaharia et al. “Spark: Cluster Computing with Working Sets”. In: *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing.* HotCloud’10. Boston, MA: USENIX Association, 2010, pp. 10–10. URL: http://dl.acm.org/citation.cfm?id=1863103.1863113.

[86] S. J. Qin and W. Li. “Detection and identification of faulty sensors with maximized sensitivity”. In: *Proceedings of the 1999 American Control Conference (Cat. No. 99CH36251).* Vol. 1. 1999, 613–617 vol.1. DOI: 10.1109/ACC.1999.782901.

[87] J. H. Friedman. “Multivariate adaptive regression splines”. In: *The annals of statistics* (1991), pp. 1–67.

[88] M. Cheng and M. Cao. “Accurately predicting building energy performance using evolutionary multivariate adaptive regression splines”. In: *Applied Soft Computing* 22 (2014), pp. 178–188. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2014.05.015. URL: http://www.sciencedirect.com/science/article/pii/S1568494614002427.

[89] A. Ng. *CS229 Lecture notes: Supervised Learning.* Stanford University. 2012.

[90] M. Spyros and H. Michele. "ARMA Models and the Box–Jenkins Methodology". In: *Journal of Forecasting* 16.3 (), pp. 147–163. DOI: 10.1002/(SICI)1099-131X(199705)16:3<147::AID-FOR652>3.0.CO;2-X. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291099-131X%28199705%2916%3A3%3C147%3A%3AAID-FOR652%3E3.0.CO%3B2-X.

[91] K. Chodorow and M. Dirolf. *MongoDB: The Definitive Guide*. 1st. O'Reilly Media, Inc., 2010. ISBN: 1449381561, 9781449381561.

[92] C. Koo et al. "An estimation model for the heating and cooling demand of a residential building with a different envelope design using the finite element method". In: *Applied Energy* 115 (2014), pp. 205–215. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2013.11.014. URL: http://www.sciencedirect.com/science/article/pii/S0306261913009070.

[93] A. P.F. Andaloro et al. "Energy certification of buildings: A comparative analysis of progress towards implementation in European countries". In: *Energy Policy* 38.10 (2010). The socio-economic transition towards a hydrogen economy - findings from European research, with regular papers, pp. 5840–5866. ISSN: 0301-4215. DOI: http://dx.doi.org/10.1016/j.enpol.2010.05.039. URL: http://www.sciencedirect.com/science/article/pii/S0301421510004106.

[94] E. Di Corso et al. "Exploring energy certificates of buildings through unsupervised data mining techniques". In: *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE Computer Society, 2017, pp. 991–998. DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData.2017.152.

[95] Capozzoli A. et al. "Data mining for energy analysis of a large data set of flats". In: *Proceedings of the Institution of Civil Engineers - Engineering Sustainability* 170.1 (2017), pp. 3–18. DOI: 10.1680/jensu.15.00051. URL: https://doi.org/10.1680/jensu.15.00051.

[96] Z. Yu et al. "A decision tree method for building energy demand modeling". In: *Energy and Buildings* 42.10 (2010), pp. 1637–1646. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2010.04.006. URL: http://www.sciencedirect.com/science/article/pii/S0378778810001350.

[97] A.P. Melo et al. "Development of surrogate models using artificial neural network for building shell energy labelling". In: *Energy Policy* 69 (2014), pp. 457–466. ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2014.02.001. URL: http://www.sciencedirect.com/science/article/pii/S0301421514000883.

[98] G. Dall'O' et al. "On the use of an energy certification database to create indicators for energy planning purposes: Application in northern Italy". In: *Energy Policy* 85 (2015), pp. 207–217. ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2015.06.015. URL: http://www.sciencedirect.com/science/article/pii/S0301421515002335.

[99] W. Chung, Y.V. Hui, and Y. M. Lam. "Benchmarking the energy efficiency of commercial buildings". In: *Applied Energy* 83.1 (2006), pp. 1–14. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2004.11.003. URL: http://www.sciencedirect.com/science/article/pii/S0306261904002028.

[100] T. Hong et al. "An estimation methodology for the dynamic operational rating of a new residential building using the advanced case-based reasoning and stochastic approaches". In: *Applied Energy* 150 (2015), pp. 308–322. ISSN: 0306-2619. DOI: http://doi.org/10.1016/j.apenergy.2015.04.036. URL: http://www.sciencedirect.com/science/article/pii/S0306261915004912.

[101] X. Gao and A. Malkawi. "A new methodology for building energy performance benchmarking: An approach based on intelligent clustering algorithm". In: *Energy and Buildings* 84 (2014), pp. 607–616. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2014.08.030. URL: http://www.sciencedirect.com/science/article/pii/S0378778814006720.

[102] R. Arambula Lara et al. "Energy audit of schools by means of cluster analysis". In: *Energy and Buildings* 95 (2015). Special Issue: Historic, historical and existing buildings: designing the retrofit. An overview from energy performances to indoor air quality, pp. 160–171. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2015.03.036. URL: http://www.sciencedirect.com/science/article/pii/S0378778815002455.

[103] F. Khayatian, L. Sarto, and G. Dall'O'. "Application of neural networks for evaluating energy performance certificates of residential buildings". In: *Energy and Buildings* 125 (2016), pp. 45–54. ISSN: 0378-7788. DOI: https://doi.org/10.1016/j.enbuild.2016.04.067. URL: http://www.sciencedirect.com/science/article/pii/S0378778816303322.

[104] G. K. F. Tso and K. K. W. Yau. "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks". In: *Energy* 32.9 (2007), pp. 1761–1768. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2006.11.010. URL: http://www.sciencedirect.com/science/article/pii/S0360544206003288.

[105] K. Fabbri, L. Tronchin, and V. Tarabusi. "Real Estate market, energy rating and cost. Reflections about an Italian case study". In: *Procedia Engineering* 21 (2011). 2011 International Conference on Green Buildings and Sustainable Cities, pp. 303–310. ISSN: 1877-7058. DOI: http://dx.doi.org/10.1016/j.proeng.

2011.11.2019. URL: http://www.sciencedirect.com/science/article/pii/S1877705811048533.

[106] C. Hjortling et al. "Energy mapping of existing building stock in Sweden – Analysis of data from Energy Performance Certificates". In: *Energy and Buildings* 153 (2017), pp. 341–355. ISSN: 0378-7788. DOI: http://dx.doi.org/10.1016/j.enbuild.2017.06.073. URL: http://www.sciencedirect.com/science/article/pii/S0378778817321850.

[107] H. Xiao, Q. Wei, and Y. Jiang. "The reality and statistical distribution of energy consumption in office buildings in China". In: *Energy and Buildings* 50 (2012), pp. 259–265. ISSN: 0378-7788. DOI: http://dx.doi.org/10.1016/j.enbuild.2012.03.048. URL: http://www.sciencedirect.com/science/article/pii/S0378778812001971.

[108] E. G. Dascalaki et al. "Energy certification of Hellenic buildings: First findings". In: *Energy and Buildings* 65 (2013), pp. 429–437. ISSN: 0378-7788. DOI: http://dx.doi.org/10.1016/j.enbuild.2013.06.025. URL: http://www.sciencedirect.com/science/article/pii/S0378778813003630.

[109] M. Gangolells et al. "Energy mapping of existing building stock in Spain". In: *Journal of Cleaner Production* 112 (2016), pp. 3895–3904. ISSN: 0959-6526. DOI: http://dx.doi.org/10.1016/j.jclepro.2015.05.105. URL: http://www.sciencedirect.com/science/article/pii/S0959652615006848.

[110] T. Hastie, R. Tibshirani, and J. Friedman. "Boosting and Additive Trees". In: *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer New York, 2009, pp. 337–387. ISBN: 978-0-387-84858-7. DOI: 10.1007/978-0-387-84858-7_10. URL: http://dx.doi.org/10.1007/978-0-387-84858-7_10.

[111] H. Peng, F. Long, and C. Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (Aug. 2005), pp. 1226–1238. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2005.159.

[112] K. Pearson F.R.S. "LIII. On lines and planes of closest fit to systems of points in space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572. DOI: 10.1080/14786440109462720. URL: https://doi.org/10.1080/14786440109462720.

[113] C. Ding and H. Peng. "Minimum Redundancy Feature Selection from Microarray Gene Expression Data". In: *Journal of Bioinformatics and Computational Biology* 03.02 (2005), pp. 185–205. DOI: 10.1142/S0219720005001004. URL: http://www.worldscientific.com/doi/abs/10.1142/S0219720005001004.

[114] R. Battiti. “Using mutual information for selecting features in supervised neural net learning”. In: *IEEE Transactions on Neural Networks* 5.4 (July 1994), pp. 537–550. ISSN: 1045-9227. DOI: 10.1109/72.298224.

[115] M. Hofmann and R. Klinkenberg. *RapidMiner: Data Mining Use Cases and Business Analytics Applications*. Chapman & Hall/CRC, 2013. ISBN: 1482205491, 9781482205497.

[116] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria, 2013. URL: http://www.R-project.org/.

[117] Pang-Ning, T. and Steinbach, M. and Kumar, V. *Introduction to Data Mining*. Addison-Wesley, 2006.

[118] C. Buratti, M. Barbanera, and D. Palladino. “An original tool for checking energy performance and certification of buildings by means of Artificial Neural Networks”. In: *Applied Energy* 120.Supplement C (2014), pp. 125–132. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2014.01.053. URL: http://www.sciencedirect.com/science/article/pii/S0306261914000828.

[119] M.Y Rafiq, G Bugmann, and D.J Easterbrook. “Neural network design for engineering applications”. In: *Computers & Structures* 79.17 (2001), pp. 1541–1552. ISSN: 0045-7949. DOI: https://doi.org/10.1016/S0045-7949(01)00039-6. URL: http://www.sciencedirect.com/science/article/pii/S0045794901000396.

[120] X. Xiao et al. “Twitter data laid almost bare: An insightful exploratory analyser”. In: *Expert Systems with Applications* 90 (2017), pp. 501–517. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2017.08.017. URL: http://www.sciencedirect.com/science/article/pii/S0957417417305559.

[121] S. Räbiger and M. Spiliopoulou. “A framework for validating the merit of properties that predict the influence of a twitter user”. In: *Expert Systems with Applications* 42.5 (2015), pp. 2824–2834. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2014.11.006. URL: http://www.sciencedirect.com/science/article/pii/S0957417414006915.

[122] K. Thomas et al. “Suspended Accounts In Retrospect: An Analysis of Twitter Spam”. In: *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM. 2011, pp. 243–258.

[123] E. Baralis et al. “Analysis of Twitter Data Using a Multiple-level Clustering Strategy”. In: *Model and Data Engineering*. Springer, 2013, pp. 13–24.

[124] C. Vicient and A. Moreno. “Unsupervised topic discovery in micro-blogging networks”. In: *Expert Systems with Applications* 42.17–18 (2015), pp. 6472–6485. ISSN: 0957-4174. DOI: http://dx.doi.org/10.1016/j.eswa.2015.04.014. URL: //www.sciencedirect.com/science/article/pii/S0957417415002444.

[125] M. Yang and H. Rim. “Identifying interesting Twitter contents using topical analysis”. In: *Expert Systems with Applications* 41.9 (2014), pp. 4330–4336. ISSN: 0957-4174. DOI: http://dx.doi.org/10.1016/j.eswa.2013.12.051. URL: //www.sciencedirect.com/science/article/pii/S0957417414000141.

[126] O. Phelan, K. Mccarthy, and B. Smyth. “Using twitter to recommend real-time topical news”. In: *Proceedings of the third ACM conference on Recommender systems*. ACM. 2009, pp. 385–388.

[127] T. Kim et al. “What’s Happening: Finding Spontaneous User Clusters Nearby Using Twitter.” In: *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*. IEEE. 2011, pp. 806–809.

[128] C. H. Lee. “Mining spatio-temporal information on microblogging streams using a density-based online clustering method”. In: *Expert Systems with Applications* 39.10 (2012), pp. 9623–9641.

[129] E. Steiger, B. Resch, and A. Zipf. “Exploration of Spatiotemporal and Semantic Clusters of Twitter Data Using Unsupervised Neural Networks”. In: *Int. J. Geogr. Inf. Sci.* 30.9 (Sept. 2016), pp. 1694–1716. ISSN: 1365-8816. DOI: 10.1080/13658816.2015.1099658. URL: http://dx.doi.org/10.1080/13658816.2015.1099658.

[130] J. Bernabe-Moreno et al. “A new model to quantify the impact of a topic in a location over time with Social Media”. In: *Expert Systems with Applications* 42.7 (2015), pp. 3381–3395. ISSN: 0957-4174. DOI: http://dx.doi.org/10.1016/j.eswa.2014.11.067. URL: //www.sciencedirect.com/science/article/pii/S0957417414007696.

[131] R. Lee, S. Wakamiya, and K. Sumiya. “Exploring Geospatial Cognition Based on Location-based Social Network Sites”. In: *World Wide Web* 18.4 (July 2015), pp. 845–870. ISSN: 1386-145X.

[132] K. Saito et al. “Change point detection for burst analysis from an observed information diffusion sequence of tweets”. In: *Journal of Intelligent Information Systems* 44.2 (2015), pp. 243–269.

[133] E. Lloret et al. “Towards a unified framework for opinion retrieval, mining and summarization”. In: *Journal of Intelligent Information Systems* 39.3 (2012), pp. 711–747.

[134] P. Arcaini et al. “User-driven Geo-temporal Density-based Exploration of Periodic and Not Periodic Events Reported in Social Networks”. In: *Inf. Sci.* 340.C (May 2016), pp. 122–143. ISSN: 0020-0255.

[135] T. Sakai et al. “Real-time local topic extraction using density-based adaptive spatiotemporal clustering for enhancing local situation awareness”. In: *2015 7th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management (IC3K)*. Vol. 01. Nov. 2015, pp. 203–210.

[136] T. Cunha, C. Soares, and E. Mendes Rodrigues. “TweeProfiles: Detection of Spatio-temporal Patterns on Twitter”. In: *Advanced Data Mining and Applications: 10th International Conference, ADMA 2014, Guilin, China, December 19-21, 2014. Proceedings*. Ed. by Xudong Luo, Jeffrey Xu Yu, and Zhi Li. Cham: Springer International Publishing, 2014, pp. 123–136. ISBN: 978-3-319-14717-8.

[137] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Vol. 1. 1. Cambridge university press Cambridge, 2008.

[138] G. Salton, A. Wong, and C. S. Yang. “A Vector Space Model for Automatic Indexing”. In: *Commun. ACM* 18.11 (Nov. 1975), pp. 613–620. ISSN: 0001-0782. URL: http://doi.acm.org/10.1145/361219.361220.

[139] J. MacQueen. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[140] M. Steinbach, G. Karypis, and V. Kumar. “A comparison of document clustering techniques”. In: *KDD Workshop on Text Mining*. 2000.

[141] R. García-Gavilanes et al. “Who Are My Audiences? A Study of the Evolution of Target Audiences in Microblogs”. In: *Social Informatics: 6th International Conference, SocInfo 2014, Barcelona, Spain, November 11-13, 2014. Proceedings*. Ed. by Luca Maria Aiello and Daniel McFarland. Springer International Publishing, 2014, pp. 561–572.

[142] R. Agrawal, T. Imielinski, and Swami. “Mining association rules between sets of items in large databases”. In: *ACM SIGMOD 1993*. 1993, pp. 207–216.

[143] J. Han, J. Pei, and Y. Yin. “Mining Frequent Patterns without Candidate Generation”. In: *In SIGMOD’00, Dallas, TX* (May 2000).

[144] M. Ester et al. “Incremental Clustering for Mining in a Data Warehousing Environment”. In: *Proceedings of the 24rd International Conference on Very Large Data Bases*. VLDB ’98. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1998, pp. 323–333. ISBN: 1-55860-566-5. URL: http://dl.acm.org/citation.cfm?id=645924.671201.

[145] M. Ester et al. “A Density-based Algorithm for Discovering Clusters a Density-based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”. In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD’96. Portland, Oregon: AAAI Press, 1996, pp. 226–231. URL: http://dl.acm.org/citation.cfm?id=3001460.3001507.

[146] J. Bouttier, P. Di Francesco, and E. Guitter. “Geodesic distance in planar graphs”. In: *Nuclear Physics B* 663.3 (2003), pp. 535–567. ISSN: 0550-3213. DOI: http://dx.doi.org/10.1016/S0550-3213(03)00355-9. URL: http://www.sciencedirect.com/science/article/pii/S0550321303003559.

[147] L. Hubert and P. Arabie. “Comparing partitions”. In: *Journal of Classification* 2.1 (Dec. 1985), pp. 193–218. ISSN: 1432-1343. DOI: 10.1007/BF01908075. URL: https://doi.org/10.1007/BF01908075.

[148] W. M. Rand. “Objective Criteria for the Evaluation of Clustering Methods”. In: *Journal of the American Statistical Association* 66.336 (1971), pp. 846–850. DOI: 10.1080/01621459.1971.10482356. URL: http://www.tandfonline.com/doi/abs/10.1080/01621459.1971.10482356.

[149] S. Deerwester et al. “Indexing by latent semantic analysis”. In: *Journal of the American Society for Information Science* 41.6 (), pp. 391–407. DOI: 10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/%28SICI%291097-4571%28199009%2941%3A6%3C391%3A%3AAID-ASI1%3E3.0.CO%3B2-9.

This Ph.D. thesis has been typeset by means of the TEX-system facilities. The typesetting engine was LuaLATEX. The document class was `toptesi`, by Claudio Beccari, with option `tipotesi =scudo`. This class is available in every up-to-date and complete TEX-system installation.