Doctoral Dissertation
Doctoral Program in Computer and Control and Engineering (31th cycle)

Exploring Data Hierarchies to Discover Knowledge in Different Domains

Giuseppe Ricupero
* * * * * *

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

Doctoral Examination Committee:
Prof. Damiano Carra, Referee, Università degli studi di Verona
Prof. Robert Wrembel, Referee, Poznan University of Technology
Prof. Maurizio Morisio, Politecnico di Torino
Prof. Rossano Schifanella, Università di Torino
Prof. Antonio Servetti, Politecnico di Torino

Politecnico di Torino
2019

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.

. . . .

Giuseppe Ricupero
Turin, 2019

Summary

Nowadays, in the era of smart technologies and accessibility of mobile and wearable devices, huge amounts of data are being produced every day. The possibilities for analysis of heterogeneous data coming from the Internet of Things (IoT) applied in different complex application domains are very vast. IoT systems generate and capture massive data collections describing human mobility, citizen's perception of provided services, and the overall urban environment as in terms of air quality and weather conditions. At the same time, in the business field, enterprises are continuously acquiring data with the aim of improving their processes and to guide their business decisions towards the right direction. To this end, these huge data collections have to be properly leveraged.

Data mining techniques are powerful instruments that can be effectively used to analyze data collections and extract hidden and useful knowledge otherwise unavailable. They allow extracting previously unknown interesting patterns such as dependencies among data objects (*association rule mining*), or a model describing data classes (*classification*).

However, the continuously increasing dimension and heterogeneousness characterizing this kind of data limits the feasibility of analysis by means of the data mining techniques currently available. Therefore, an important question is how these collections can be more efficiently transformed into exploitable knowledge.

This PhD thesis addresses the study and development of *novel data analysis frameworks* and *patterns* to extract useful insights from the targeted data collections. To this end, the exploration of data taxonomies built on top of the considered data is proposed. A *data taxonomy* is a set of is-a hierarchies each one referring to a specific data attribute. Each hierarchy aggregates all the values assumed by the corresponding attributes into higher level concepts in a tree-based structure.

The data taxonomy strategy has been applied on real datasets coming both from the urban and business contexts as reference case studies. The works in the urban context focused on the generation of highly interpretable descriptive analytics at multiple levels of abstraction in order to support municipalities to obtain services that are more convenient and a better environment for their citizens. Meanwhile, the reference case studies in the business context leveraged taxonomies to improve profits coming from sales, or with the objective of aligning knowledge ontologies with different levels of granularity provided by commercial partners who want to interoperate.

The results of the experimental studies have proven the effectiveness of the developed methodologies, thus, leading to the deployment in production of one of the proposed business-domain solutions.

Acknowledgements

I have started my Ph.D. journey when I was already a grown-up man with a job, a family and a lovely daughter. So, it has been with a different mindset that I have considered all the time, patience, and effort spent by my tutor, prof. Silvia Chiusano, to guide me and teach me what is worth in the world of Academia. Together with prof. Tania Cerquitelli, you have been a crucial source of advice and support all along these years. Thank you.

The same applies for the talented researchers with whom I shared the scientific work made till now, professors Paolo Garza and Luca Cagliero. You have taught me a lot.

A thank you goes also to my Lab5 colleagues who have shared joys and sorrows of this long journey and who have always welcomed me as if I would have always been there.

It cannot miss a note of gratitude for the company I work for, Italiaonline, that trusted me and allowed me to reach this aim. A company is made of people and many of them has helped me. In particular, I want to mention Pasquale Morra who indirectly led me on this road and believed in me from the beginning.

I want to dedicate these closing lines to my beloved family. Despite being scattered along different distant places, they have always taken the time to make me feel loved and appreciated. It really helped a lot both on sunny and rainy days.

Finally, I start to partially repay the profound debt I have with my loved sweetheart. You have shown me how strong a love bond could be sharing this burden together with me. Definitely, I have received from the life an invaluable gift.

This thesis is dedicated to my beloved daughter *May the love of knowledge make you explore the world with eyes full of curiosity and amazement*

Contents

List of Tables x

List of Figures xi

1 Introduction 1
- 1.1 Research topics description . . . 1
- 1.2 Thesis plan and research contribution . . . 2
  - 1.2.1 Data mining applications in urban context . . . 2
  - 1.2.2 Data mining applications in business context . . . 3

2 Monitoring and Analyzing Air Quality in Urban Areas 7
- 2.1 Related work . . . 8
- 2.2 Theoretical background . . . 9
  - 2.2.1 Frequent itemset mining . . . 10
  - 2.2.2 Association Rules . . . 12
  - 2.2.3 Weighted frequent itemset mining . . . 12
- 2.3 *GECKO*: modeling correlations among air pollution-related data . . . 15
  - 2.3.1 Data integration . . . 16
  - 2.3.2 Data representation . . . 17
  - 2.3.3 Taxonomy generation . . . 18
  - 2.3.4 Data analyses . . . 19
  - 2.3.5 Experimental validation . . . 21
- 2.4 *ARQUATA*: discovering air quality patterns in urban environments . . . 25
  - 2.4.1 Data integration and preparation . . . 25
  - 2.4.2 Air quality pattern mining . . . 25
  - 2.4.3 Reporting . . . 26
  - 2.4.4 Experimental results and discussion . . . 27
- 2.5 Summary . . . 29

3 Green Urban Mobility: Analyses of Bike Sharing Data 31
- 3.1 Related work . . . 33
- 3.2 Methodology . . . 34

3.2.1 Data collection, modeling and enrichment 34
3.2.2 Mining Occupancy Monitoring Patterns 37
3.2.3 Knowledge exploration 47
3.3 Experimental results 49
3.3.1 Reference use case datasets 49
3.3.2 OMP characterization 50
3.3.3 Parameter analysis 57
3.3.4 Algorithm performance 61
3.4 Summary 61

4 Exploring data hierarchies to mine high-utility itemsets 65
4.1 Theoretical background 67
4.2 Generalized High-Utility Itemsets 69
4.3 The ML-HUI Miner algorithm 71
4.4 Experiments 72
4.4.1 Performance analysis 73
4.4.2 Knowledge discovery 74
4.5 Summary 75

5 Integration of business activities between different web directories 77
5.1 Related work 78
5.2 Theoretical background 79
5.3 Motivation and an industry use case 80
5.4 Methodology 82
5.4.1 Data preparation 82
5.4.2 Dataset description 83
5.4.3 Classifier architecture 83
5.5 Experimental validation 85
5.6 Summary 88

6 Conclusions 89

Bibliography 93

List of Tables

2.1 An example transactional dataset T with 6 transactions. Each transaction $t_i \in T$ is a collection of items (i.e., $t_i \subseteq I$) and is identified by a transaction identifier (TID$_i$). 10

2.2 Example dataset 13

2.3 Item weights 13

2.4 Rule examples. 23

2.5 2013 seasonal analysis: weighted frequent itemsets. Critical gap thresh. 30%. 28

3.1 Notation. 35

3.2 Example of *Occupancy Level Dataset*. 36

3.3 Example of *Occupancy Level Dataset* in transactional format. 41

3.4 *Bicing* (Barcelona). Groups of stations with maximal intermittence in different hourly time slots. 53

3.5 *Citi Bike* (New York). Groups of stations with maximal intermittence in different hourly time slots. 53

3.6 *Bicing* (Barcelona). Groups of stations with maximal criticality in different hourly time slots. 54

3.7 *Citi Bike* (New York). Groups of stations with maximal criticality in different hourly time slots. 54

4.1 Transactional dataset 67

4.2 External utilities of items in dataset 67

4.3 High-Utility Itemsets. minutil (itemsets) = 17 68

4.4 Generalized High-Utility Itemsets. minutil (generalized itemsets) = 30 . 68

4.5 Dataset characteristics 72

5.1 Example of a labeled entry from the training set (original data in Italian). 83

5.2 Training set statistics. 84

5.3 Performance indices of the *restaurants* and *restaurants.italian* classifiers. 86

5.4 Performance of a subset of classes of the level 1 *restaurants* classifier. . 86

5.5 Performance of a subset of classes of the level 2 *restaurants.italian* classifier. 87

List of Figures

2.1 The analyzed air pollutants collected through sensor networks. 8

2.2 Lattice representing the search space based on the items appearing in the example dataset D 11

2.3 The *GECKO* framework architecture. 15

2.4 An example of the generalized cross association rules extracted. 21

2.5 The *ARQUATA* engine. 26

2.6 Comparison between winters 2013, 2014, and 2015. 28

3.1 The *B*ike Station Ov*E*r*L*oad Ana*L*yzer (*BELL*) architecture. 35

3.2 The FP-Tree representing the example transactional Occupancy Level Dataset (Table 3.3). 46

3.3 O-itemset mining example. 47

3.4 *Bicing* (Barcelona). Hourly distribution of the number of OMPs and their corresponding levels of intermittence/criticality. *maxdist*=0.5 km. *time slot size*=1h. *full-th*=3. 55

3.5 *Citi Bike* (New York). Hourly distribution of the number of OMPs and their corresponding levels of intermittence and criticality. *maxdist*=0.5 km. *time slot size*=1h. *full-th*=3. 55

3.6 Heat maps representing intermittence and criticality values in Barcelona at the hourly time slot [11am-12am). *maxdist*=0.5 km, and *time slot size*=1h. *full-th*=3. 56

3.7 Heat maps representing intermittence and criticality values in New York at the hourly time slot [11am-12am). *maxdist*=0.5 km, and *time slot size*=1h. *full-th*=3. 56

3.8 Barcelona. Impact of the occupancy threshold on the characteristics of the mined OMPs. *maxdist*=0.5 km. *time slot size*=1h. 58

3.9 Barcelona. Impact of the maximum distance threshold on the characteristics of the mined OMPs. *full-th*=3. *time slot size*=1h. 59

3.10 *Bicing* (Barcelona). Distribution of the number of OMPs and their corresponding levels of intermittence/criticality with a time slot granularity of 4 hours. *maxdist*=0.5 km. *full-th*=3. *time slot size*=4h. 59

3.11 Barcelona. Characteristics of the mined OMPs related to weekdays and weekends. *full-th*=3. *time slot size*=1h. 60

4.1 Taxonomy on items in the dataset 68
4.2 Chess: number of mined patterns 73
4.3 Execution time (γ=0.2) 74
5.1 Mapping of categories at different abstraction levels. 81
5.2 The training set building process 82
5.3 The TACOMA classifier architecture. 85

Chapter 1

Introduction

In the last few years, the use of Information and Communication Technologies has made available a huge amount of heterogeneous data in various complex application domains. For example, in the urban scenario, Internet of Things (IoT) systems generate and capture massive data collections describing human mobility, citizen's perception of provided services, and the overall urban environment as in terms of air quality and weather conditions. These collections can be a valuable instrument to provide more convenient services and better environments.

Data mining techniques are powerful instruments that can be effectively used to analyze data collections and extract hidden and useful knowledge otherwise unavailable. They allow extracting previously unknown interesting patterns such as groups of similar data objects (*cluster analysis*), dependencies among data objects (*association rule mining*), or a model describing data classes (*classification*).

However, these kind of data collections are often characterized by a continuously increasing dimension and heterogeneousness, which limit the feasibility of analysis by means of data mining techniques currently available. Therefore, an important question that has been recently raised is how these collections can be transformed into exploitable knowledge.

1.1 Research topics description

This PhD thesis addresses the study and development of *novel data analysis frameworks* as well as *novel patterns* to extract useful insights from the targeted data collections. The proposed approaches rely on the design and development of proper techniques for the integration and analysis of huge volumes of heterogeneous data, covering such critical issues as the large dataset cardinality, dimensionality, and the variable data distribution.

To this end, the exploration of data taxonomies built on top of the considered data is proposed in this PhD thesis. A *data taxonomy* is a set of is-a hierarchies each one referring to a specific data attribute. Each hierarchy aggregates all the values assumed by the corresponding attributes into higher level concepts in a tree-based structure.

In the proposed frameworks, the original data are enriched with a data taxonomy with the aim of analysing data at different abstraction levels. Based on the characteristics of the explored data as well as the objective of the analysis in the targeted application domain, the most suitable data abstraction level can be selected for the data analysis task. Data analysis at the highest abstraction levels allows mining patterns that represent more succinct information, while data analysis on lower abstraction levels allows discovering patterns representing more detailed information. In this PhD thesis, cross-level patterns have also been investigated, in which data represented at different abstraction levels coexist.

1.2 Thesis plan and research contribution

The PhD activity has been conducted by considering two different application domains as reference case studies: (i) the *urban domain* and (ii) the *business domain*. Proper data analysis frameworks and patterns have been designed and developed to address some challenging research topics and peculiar data issues in the two domains.

1.2.1 Data mining applications in urban context

In the urban areas different geo-referenced IoT sensor networks are deployed to acquire a huge amount of heterogeneous data, for example, on the urban environment and the usage of services offered to citizens. The analysis of these data collections can provide useful insight to the municipality to improve the well-being of citizens. To address the critical issue of assisting municipality actors in the process of improving the air quality in smart cities, two novel data analysis frameworks (named *GECKO* [14] and *ARQUATA* [12]) have been designed and developed. Moreover, an additional data analysis framework has been developed, and it is aimed at improving the quality of a widespread green mobility system, i.e., bike sharing [13].

Analyzing air-pollution related data. The *GECKO* (GEneralized Correlation analyzer of pOllution data) framework [14], focusing on the analysis of air quality in the urban environment, takes into account not only heterogeneous air-pollution related data including the concentration of the air pollutants, but also traffic flow measurements and meteorological data. The knowledge extraction process is driven by taxonomies used to generalize low-level measurement values into corresponding high-level categories. The concept of taxonomies is exploited in the process of association rules mining to discover interesting and multiple-level correlations among the different types of collected data at different abstraction levels. To ease the manual inspection of the results, the extracted correlations are classified into a few classes based on the semantics of underlying data.

The data mining engine named *ARQUATA* (AiR QUAlity patTern Analyzer) [12] focuses on the analysis of pollutant concentration data aimed at discovering combinations of pollutant concentrations that are, on average, in a critical condition. An established type of pattern, namely the weighted frequent itemset pattern, is used to identify these air quality patterns. To offer different viewpoints of analysis to domain experts and municipality actors, these patterns are extracted from several aggregations of the raw data following specific temporal and spatial granularities. For instance, the data coming from the city center area can be monitored along different seasons, or with the heating systems off and on, to be examined by domain experts.

To demonstrate the effectiveness of the proposed analytics engines, both the *ARQUATA* and *GECKO* frameworks have been validated using real open data acquired from a major Italian Smart City (i.e., Milan).

Analyzing bike-sharing systems data. Bike-sharing systems are green mobility systems that help to enhance the quality of life in cities by reducing pollutant emissions and traffic congestion. The PhD activity addressed the design and development of the *BELL* (Bike Station OvErLoad AnaLyzer) system [13] with the aim of improving the user perception and ease of maintenance of bike-sharing systems. The proposed *BELL* methodology relies on a new pattern type called *OMP* (Occupancy Monitoring Pattern), which detects situations of dock overload in multiple stations.

The analysis of station occupancy levels at different time granularities allows system managers to investigate how overload conditions evolve over time, and to identify overload conditions that frequently occur in specific time periods. To leverage the concept of taxonomy, the occupancy level data have been enriched with temporal information with a coarser granularity. The granularity of the time period can be defined based on the target analysis. For instance, daily granularity allows spotting larger-scale phenomena, such as the most unbalanced days in terms of dock occupancy, while an hourly granularity can help determine at what hour to schedule the re-balancing of the bikes among the stations. The effectiveness of generating useful insights has been demonstrated by the results of the experimentation on real open data acquired in different Smart Cities (i.e., Barcelona, and New York).

1.2.2 Data mining applications in business context

Another direction of studies of the generalization concept covers business context. In this field, data coming from online retail [16], e.g., Amazon, and web directory, i.e., Pagine Gialle have been exploited as example case-studies. Specifically, one of the studies has leveraged generalization to improve planning advertising campaigns of retail products, while the other study is aimed at developing a classification model to support the integration of business activities among different web directories.

Discovering the most profitable sets of products. High-Utility Itemset Mining (HUIM) is an established data mining technique used to discover recurrent combinations of products (items) characterized by high profit from transactional datasets. Based on the observation that items can be clustered into domain-specific categories, the PhD activity proposed a new type of pattern, named *GHUI* (Generalized High-utility Itemset) [16] that entails generating correlations among data items at multiple abstraction levels. Specifically, GHUIs represent combinations of items at different granularity levels characterized by high profit (utility). While profitable combinations of item categories provide interesting high-level information, GHUIs at lower abstraction levels represent more specific correlations among profitable items. A single-phase algorithm is proposed to efficiently discover GHUIs. The experiments, which were performed on both real and synthetic data, demonstrate the effectiveness and usefulness of the proposed approach.

Taxonomy-based integration of business activities between different web directories. A pervasive problem on the web is represented by the integration of information coming from various sources and categorized according to significantly different taxonomies. In particular, it is still an open research issue, how to correctly categorize each instance of a certain concept of the source taxonomy when the corresponding instance of the target taxonomy has a finer granularity level. For example, the concept of the source taxonomy *furniture* should be mapped to the concepts of the target taxonomy such as *chair*, *table*, *bookcase*.

In this study, the issue has been formulated as a classification problem. Each target category is treated as a class, and all the relevant textual data of the specific instance are taken into account to build a classification model. Additionally, the target taxonomy is leveraged to generalize to a higher-level category when a coherent match is missing. The proposed approach, namely *TACOMA* (Text-bAsed CategOry MApping), has been validated using different classification algorithms (i.e., Artificial Neural Networks, Support Vector Machines, Random Forest) on real business activities data coming from a prominent web directory (i.e., Pagine Gialle) to be integrated with a widely used navigation system (i.e., Apple Maps). Due to the good level of accuracy, the results of this study have been integrated into the company production system and the paper about this research activity is in progress.

This thesis is organised as follows. Chapters 2 and 3 present the research activity conducted on the urban application domains. More specifically, Chapter 2 presents the *GECKO* and *ARQUATA* frameworks for the analysis of air pollution related data, while Chapter 3 presents the *BELL* engine for the analysis of bike-sharing systems data. Chapters 4 and 5 focus on the research activity performed on the business domain. Chapter 4 describes the *GHUI* pattern that entails generating correlations among data items at multiple abstraction levels. Chapter 5 presents the *TACOMA* approach for the taxonomy-based integration of business activities among different web directories. Finally, Chapter 6 summarizes the results presented in this PhD thesis and discusses future developments for the proposed approaches.

Chapter 2

Monitoring and Analyzing Air Quality in Urban Areas

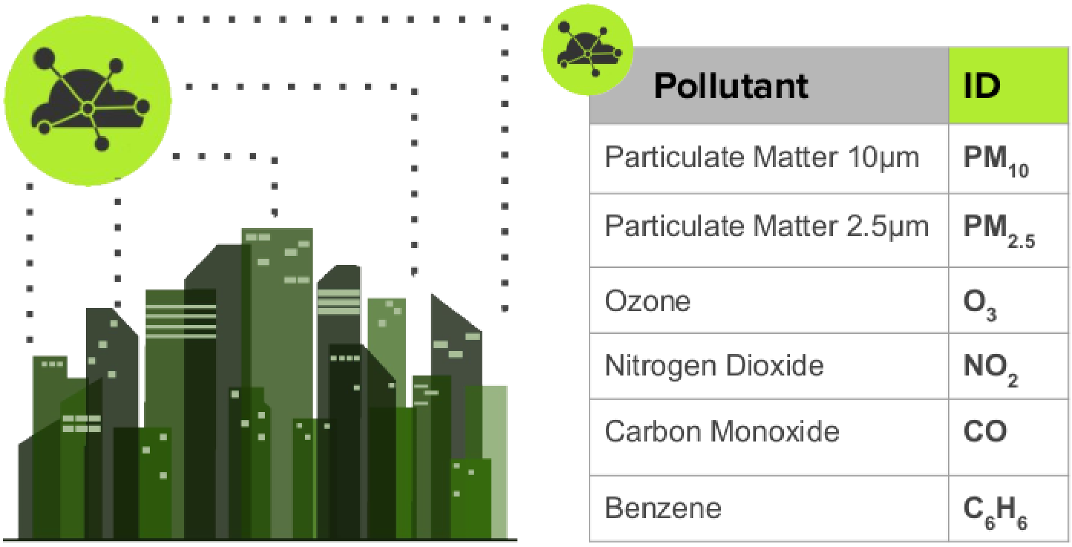
Nowadays Smart Cities are increasingly pervaded by sensors deployed in public areas, on vehicles, and on wearable devices. These sensor networks allow us to collect a variety of data useful for monitoring the factors influencing the quality of citizen's life from different viewpoints. Counteracting the presence of high levels of pollutants is crucial to ensure the livability of urban environments.

The quality of the air can vary over time and across different areas of the same city. Furthermore, it is influenced by different factors such as weather conditions (e.g., humidity, temperature and atmospheric pressure) and human activities (e.g., traffic flows, people's mobility). To monitor pollutant concentrations and their relationship with meteorological and traffic conditions, sensor networks are deployed by the public administration over the city area (see Figure 2.1). Air quality data acquired from sensor networks can be further enriched with information coming from wearable sensors, while climate data can be measured through personal weather stations. Analyzing the air quality levels acquired by sensors is particularly useful for characterizing pollutant concentrations. Sensor data are usually sampled at fairly high frequencies, for relatively long time periods, and across potentially large city areas.

In this chapter, two frameworks are presented to analyze air-pollution related data and calculate how the above mentioned factors impact on the air quality of the cities.

In Section 2.3, a novel data mining system, named *GECKO* (*GEneralized Correlation analyzer of pOllution data*), is presented. The *GECKO* system leverages the power and expressiveness of the generalized association rules to extract, by applying domain-expert provided taxonomies on top of the data, interpretable correlations at different abstraction levels among a large variety of data related to air quality (e.g., meteorological conditions, acquisition times, vehicular traffic measurements).

Differently, the *ARQUATA* (*AiR QUAlity patTern Analyzer*) engine, presented in Section 2.4, focuses on the air-pollution data to better characterize their concentrations



Pollutant	ID
Particulate Matter 10µm	PM_{10}
Particulate Matter 2.5µm	$PM_{2.5}$
Ozone	O_3
Nitrogen Dioxide	NO_2
Carbon Monoxide	CO
Benzene	C_6H_6

Figure 2.1: The analyzed air pollutants collected through sensor networks.

through a newly defined class of data mining patterns, hereafter denoted as *air quality patterns*, which extract combinations of pollutants averagely in a critical condition. Here, the concept of temporal and spatial aggregation is applied to offer different viewpoints of analysis to domain experts and municipality actors.

GECKO and *ARQUATA* engines were validated on real open data collected in a major Italian city (i.e., Milan). The discovered patterns demonstrate their effectiveness in extracting interesting knowledge that can be easily exploited by public administrators to monitor the air quality in urban environments through the reports automatically generated by both the illustrated systems.

2.1 Related work

Other authors have already studied the correlation between different pollutants through statistics-based methods such as one-way ANOVA analysis [5]. Furthermore, Principal Component and Canonical Correlation analyses [75] have been leveraged to analyze the correlation between pollutants and meteorological data [24]. A parallel effort has been devoted to make use of data mining techniques to analyze the air quality levels in urban environments [87, 88]. Classification algorithms have been used to predict the air quality level in areas not equipped with monitoring stations [87]. To train the classification model, historic and real-time measurements on air quality, weather conditions, traffic flows, and people’s mobility have jointly been analyzed. Similarly, in [88] air quality and meteorological data acquired in the past were analyzed to predict the level of the air quality in the near future. Association rule mining approaches have found application in various application domains (e.g., network traffic analysis [9], social data analysis [17]) to discover interesting correlations among data items. The exploitation of these approaches on air pollution-related data can support the discovery of interesting yet hidden knowledge. The extracted patterns are commonly managed by domain experts through manual inspection to support decision-making.

However, none of the above-mentioned approaches has leveraged the generalized association rules technique to analyze the correlations among air-pollution related data. The *GECKO* system allows to extract interpretable correlations, at different abstraction levels, among a large variety of data related to air quality. Pollutant measurements are first integrated with traffic and meteorological data and enriched with an analyst-provided taxonomy, which aggregates measurement values into the corresponding higher-level categories. Then, an established generalized association rule mining algorithm [9] is applied to the prepared dataset. The extracted rules, namely the generalized association rules, represent frequent co-occurrences between pollutant levels and environmental conditions at different abstraction levels. Finally, to ease the expert-driven rule inspection process, the rules are classified into few classes according to the semantics of the represented information.

The *ARQUATA* engine, unlike other approaches (e.g., [24], [87], [88]), proposes the use of a class of data mining patterns, hereafter denoted as air quality patterns, to discover combinations of pollutants whose concentration levels are averagely critical in a given spatio-temporal context (e.g., the sensor measurements acquired in a given city area during the last year). Among the patterns available in data mining literature, in this study, the weighted frequent itemsets were considered, because, unlike traditional pattern mining approaches, weighted itemset mining algorithms inherently handle numerical pollutant levels. This simplifies the preprocessing phase of the analyzed data and reduces the bias due to discretization. *ARQUATA* also supports the generation of automatic reports, which indicate the presence of critical conditions in specific contexts and their temporal evolution, by performing a comparison between the results of different mining sessions scheduled in consecutive time periods or in different city areas.

2.2 Theoretical background

This section contains the necessary theory regarding the techniques used in the research studies of this chapter: *frequent itemset* and *association rules* for *GECKO* and *weighted frequent itemset* for *ARQUATA*.

2.2.1 Frequent itemset mining

Itemset mining is an exploratory data mining technique which consists of discovering interesting and useful patterns in transactional databases [2]. More specifically, it entails discovering the groups of attribute values that frequently co-occur in the analyzed database. Itemset mining has been applied in various application domains such as market basket analysis, bio-informatics, text mining, product recommendation, and Web clickstream analysis.

In the context of relational data, an *itemset* is a set of items (*attribute*, *value*) all belonging to distinct attributes. For example, itemset {(PM$_{2.5}$, *red*),(*wind-direction*, *south-east*)} indicates that items (PM$_{2.5}$, *red*) and (*wind-direction*, *south-east*) co-occur in the analyzed data. A more formal definition follows:

Definition 2.2.1 (Itemset and Support Count)**.** Let $S = \{i_1, \ldots, i_n\}$ be the set of all items in a transactional dataset and $T = \{t_1, \ldots, t_N\}$ be the set of all the transactions whose number is N. Each transaction t_i contains a subset of items chosen from T. In association analysis, a collection of zero or more items is termed as *itemset.* If an itemset contains k items, it is called k-itemset. A transaction t_i is said to contain an itemset X, if X is a subset of t_i, i.e., $X \subseteq t_i$.

An important property of an itemset is its support count, which refers to the number of transactions that contain a particular itemset. The support count for an itemset X, written as $\sigma(X)$, can be stated as follows:

$$\sigma(X) = |\{t_i : X \subseteq t_i, t_i \in T\}|$$

In the dataset shown in Table 2.1 the support count for the itemset $\{A, B, C\}$ is equal to 2, because there are two transactions that contain all the three items (i.e., transaction with TID equal to 4 and 5). □

TID	Transaction
1	{ A, B, D, E }
2	{ B, C, E }
3	{ A, B, D, E }
4	{ A, B, C, E }
5	{ A, B, C, D, E }
6	{ B, C, D }

Table 2.1: An example transactional dataset T with 6 transactions. Each transaction $t_i \in T$ is a collection of items (i.e., $t_i \subseteq I$) and is identified by a transaction identifier (TID$_i$).

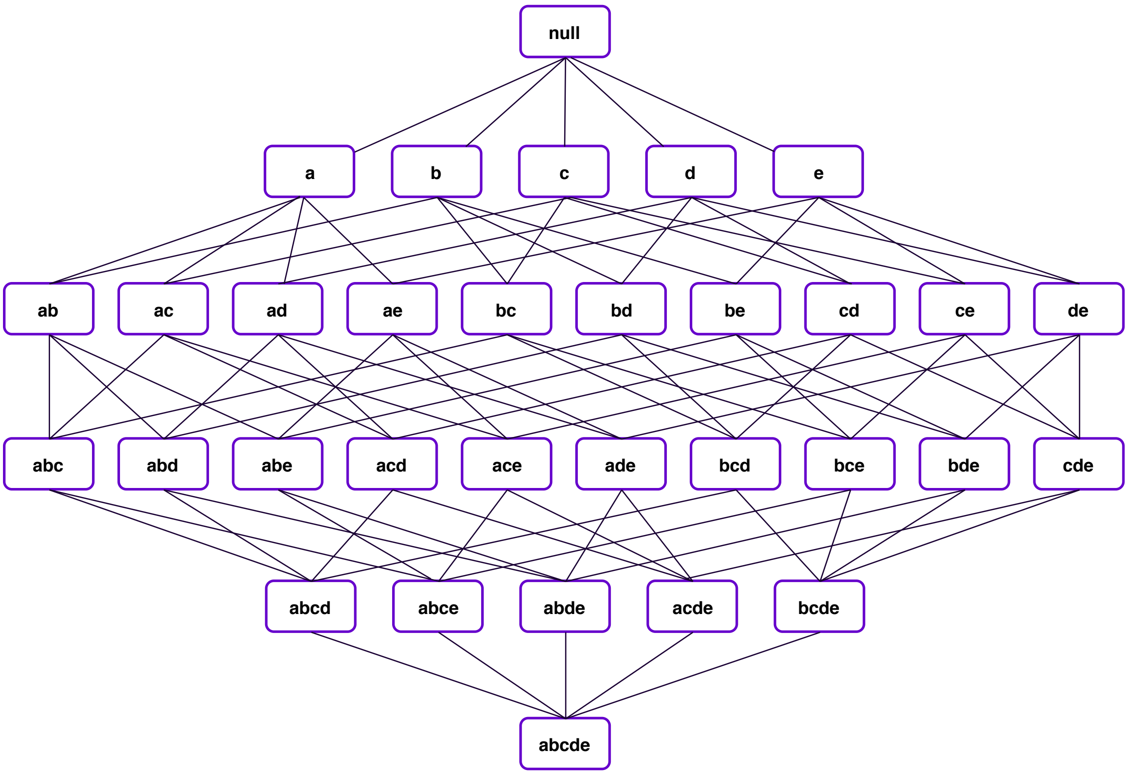


Figure 2.2: Lattice representing the search space based on the items appearing in the example dataset D

Support of an itemset Another way of characterizing an itemset X is through its support value, which is denoted by $sup(X)$ and defined as the ratio between the number of transactions in T containing X (i.e., $\sigma(X)$) and the total number of transactions in T. In the example dataset in Table 2.1, for example, the support of the itemset $\{A, B, D\}$ is 50% (3/6). This value represents the frequency of occurrence of the itemset in the dataset.

Frequent Itemset Mining (FIM) An itemset X is considered frequent if its support is greater than a user-provided minimum support threshold *minsup*. Given a transactional dataset T and a minimum support threshold *minsup*, the Frequent Itemset Mining problem consists in extracting the complete set of frequent itemsets from T.

The dimension of the search space can be represented as a lattice, whose top is an empty set. Its size increases exponentially with the number of items.

Due to the exponential growth of the lattice, data mining techniques, make often use of an approximate representation or a subset of the complete lattice, which is also difficult to store. In Figure 2.2, the lattice related to the example in Table 2.1 is shown.

2.2.2 Association Rules

Definition 2.2.2 (Association Rule)**.** An association rule is an implication expression in the form $X \implies Y$, where X and Y are disjoint itemsets (i.e., sets of data items, see 2.2.1) known respectively as *Antecedent* and *Consequent* of the rule. □

Definition 2.2.3 (Support and Confidence of an association rule)**.** The strength of an association rule $X \implies Y$ is measured by its *support* and *confidence*. Rule support determines how often a rule is applicable to a given data set, while confidence how frequently items in Y appear in transactions that contain also X. More formal definitions of *support* and *confidence* follow:

$$supp(X \implies Y) = \frac{\sigma(X \cup Y)}{N}$$

$$conf(X \implies Y) = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

where the definition of the support count function $\sigma(X)$ is given in Section 2.2.1 and N is the total number of transactions in the dataset. □

Definition 2.2.4 (Lift)**.** In some cases, measuring the strength of a rule in terms of support and confidence may be misleading. When the rule *consequent* is characterized by relatively high support value, the corresponding rule may be characterized by high confidence even if its actual strength is relatively low. To overcome this issue, the lift (or correlation) index may be used, to measure the (symmetric) correlation between sets X and Y. The lift index is defined as the ratio:

$$lift(X \implies Y) = \frac{conf(X \implies Y)}{sup(Y)}$$

Lift values below 1 show a negative correlation between sets X and Y, while values above 1 indicate a positive correlation. The interest of rules having a lift value close to 1 may be marginal. □

2.2.3 Weighted frequent itemset mining

Frequent Itemset Mining (FIM) is an important data mining task having plenty of real-world applications. However, an important limitation of FIM is that it gives the same importance to all the items in the dataset ignoring their weight, interest, risk, or profit. To address this issue, the problem of Weighted Frequent Itemset Mining (WFIM) was proposed by considering the importance of each item. Cai et al. first defined a weighted-support model by multiplying the support of each item by its average weight [19].

Preliminaries Let $S = \{i_1, i_2, ..., i_m\}$ be a finite set of m distinct items appearing in a transactional database $T = \{t_1, t_2, ..., t_n\}$, where each transaction $T_q \in T$ is a subset of S, and has a unique identifier called TID. A weight $w(i_j)$ is assigned to each item $i_j \in S$, which represents its importance (e.g., profit, interest, risk). Weights for all items are stored in a weight table $wtable = \{w(i_1), w(i_2), ..., w(i_m)\}$.

An itemset $I \in S$ with k distinct items $\{i_1, i_2, ..., i_k\}$ is of length k and is referred to as a k-itemset. An itemset I is said to be contained in a transaction t_q if $I \subseteq t_q$. Furthermore, for an itemset I, let the notation TIDs(I) denotes the TIDs of all transactions in T containing I.

As a running example, table 2.2 shows a transactional database containing 4 transations.

Definition 2.2.5 (Item weight)**.** The weight of an item $i_j \in T$ is denoted as $w(i_j)$, and represents the importance of this item to the user ($w(i_j) \in (0, 1]$).

TID	Items
1	a,b,c,d
2	b,d
3	a,b,c
4	c,d

Table 2.2: Example dataset

Item	weight
a	0.4
b	0.7
c	1.0
d	0.5

Table 2.3: Item weights

□

Definition 2.2.6 (Item weight in a transaction)**.** The weight of an item i_j in t_q is defined as the weight of i_j in T. Thus: $w(i_j, t_q) = w(i_j)$, $1 \leq q \leq |T|$.

For example, the weight of (b) in t_1 is $w(b, t_1) = w(b) = 0.7$. □

Definition 2.2.7 (Itemset weight)**.** The weight of an itemset I in T indicated as $w(I)$ is defined as the sum of the weights of all the items in I divided by the number of items in I, that is: $w(I) = \frac{\sum_{i_j \in I} w(i_j)}{|I|}$, where $|I|$ is the cardinality of I.

For example, the weight of (bce) is calculated as $w(abc) = (w(a) + w(b) + w(c))/3 = (0.4 + 0.7 + 1.0)/3 = 0.7$. □

Definition 2.2.8 (Itemset weight in a transaction)**.** The weight of an itemset I in t_q is defined as as the weight of the itemset I in T, that is: $w(I, t_q) = w(I)$.

For example, the weight of (abc) in t_1 is calculated as $w(abc, t_1) = w(abc) = 0.7$. □

Definition 2.2.9 (Weighted support for an itemset in T.)**.** The weighted support of an itemset I in T is denoted as $wsup(I)$, and is defined as:

$$wsup(I) = \sum_{I \subseteq t_q \wedge t_q \in T} w(I, t_q) = w(I) \times sup(I)$$

For example, the (abc) appears in transactions t_1, t_3. The weighted support of (abc) is calculated as $wsup(abc) = \{w(abc, t_1) + w(abc, t_3)\} = w(abc) \times 2 = 0.7 \times 2 = 1.4$. □

Definition 2.2.10 (Weighted Frequent Itemset, WFI)**.** Let α be a user-defined percentage value named the minimum weighted-support threshold. An itemset I in T is said to be a weighted frequent itemset (WFI) if its weighted support is no less than the minimum weighted-support threshold multiplied by the number of transactions in T, that is: WFI $\leftarrow \{I \mid wsup(I) \geq \alpha \times |T|\}$. □

2.3 *GECKO*: modeling correlations among air pollution-related data

The *GEneralized Correlation analyzer of pOllution data* (*GECKO*) system is a data mining engine that analyze the correlations between pollutants and different environmental factors, such as meteorological and traffic conditions, in a Smart City context. The main architectural blocks are:

(i) *Data integration*, in which pollutant and environmental data are acquired and integrated,

(ii) *Data representation*, in which data are tailored to a relational data format,

(iii) *Taxonomy generation*, in which a domain-expert taxonomy is applied on top of data to enrich them aggregating concepts into higher-level ones,

(iv) *Data analyses*, in which generalized association rules are extracted from the prepared data to support domain experts in performing advanced analyses.

A more detailed description of the architecture, which is illustrated in Figure 2.3, is given as follows.

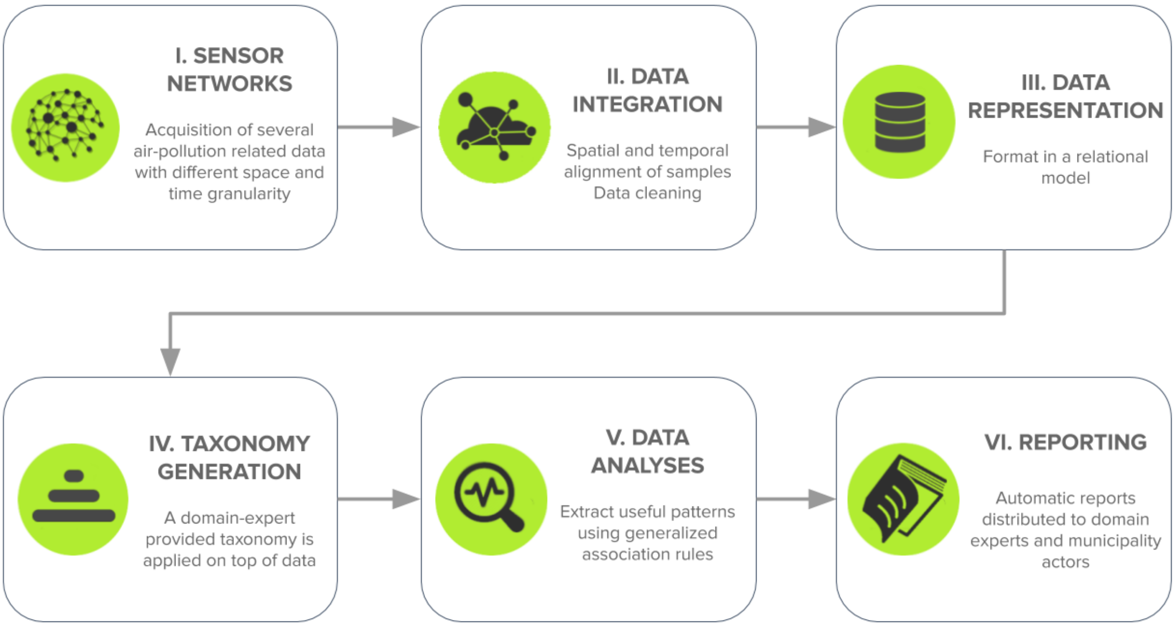

I. SENSOR NETWORKS
Acquisition of several air-pollution related data with different space and time granularity
II. DATA INTEGRATION
Spatial and temporal alignment of samples
Data cleaning
III. DATA REPRESENTATION
Format in a relational model
IV. TAXONOMY GENERATION
A domain-expert provided taxonomy is applied on top of data
V. DATA ANALYSES
Extract useful patterns using generalized association rules
VI. REPORTING
Automatic reports distributed to domain experts and municipality actors

Figure 2.3: The *GECKO* framework architecture.

2.3.1 Data integration

Since the concentrations of pollutants can be relevantly affected by both weather conditions (e.g., temperature, humidity) and type of traffic crossing the city area (e.g., how many gasoline engine vehicles crossed the area), different sensor networks should be exploited to periodically monitor values for different data types. Specifically, measurements for three main types of data should be acquired: pollutant data, meteorological data, and traffic data. In urban environments, a different geo-referenced sensor network is usually deployed for monitoring each of the above data types. An ad hoc integration strategy is applied since the considered sensor networks may adopt a different timeline in sampling values and be deployed in different city areas. In this section, first, the considered data types are described, and then the data integration strategy currently adopted in *GEneralized Correlation analyzer of pOllution data.*

Pollutant data. Concentration measurements for each pollutant were periodically collected through dedicated sensors deployed in pollution monitoring stations (PolMS). Each station is characterized by the geo-coordinates (i.e., latitude and longitude) of its location, and stations are located in different areas of the city. The most damaging pollutants are monitored, including particulate matters PM_{10} and $\text{PM}_{2.5}$, carbon monoxide (CO), and ozone (O_3). Each station monitors the concentrations of various pollutants at a fixed time granularity. Depending on the type of pollutant, the frequencies of data acquisition can be hourly or daily.

Meteorological data. To analyze the climate conditions of the urban area, the *GEneralized Correlation analyzer of pOllution data* collects the most common meteorological indicators (e.g., air temperature, relative humidity, precipitation level, wind speed, atmospheric pressure). Climate conditions are acquired through geo-referenced meteorological stations distributed throughout the urban territory.

Traffic data. The concentration of traffic is measured as the number of vehicles entering a city area at a given time granularity (e.g., hourly). Since vehicles equipped with different engines may affect the air quality differently, traffic data was considered separately for each category of vehicles. Specifically, vehicles are categorized based on their fuel type (e.g., gasoline, diesel, electric).

To allow the analysis of the correlations between pollutant levels and environmental factors (i.e., weather and traffic conditions), the three different types of data described above are integrated into a unique repository. Meteorological and traffic data are preprocessed before data integration to align the spatial and temporal granularity of the acquired data. Since the analysis is focused on pollutant data, the spatial-temporal granularity of the sensor network monitoring pollutant concentrations is considered as a reference for time and space alignment.

To effectively deal with alignment issues, for each Pollution Monitoring Station (PolMS) meteorological and traffic data are aligned to the closest timestamp available in pollutant data through an approximate join. Specifically, meteorological data associated with a given pollution station are computed as a distance-based weighted mean of the values provided by the three nearest meteorological stations monitoring climate data. The weight assigned to each value is inversely proportional to the distance from these three stations to the PolMS. Hence, three equally distant meteorological stations would have the same importance for determining the weather values of a given city area. For traffic data the number of vehicles entering each area is associated to all the sensors deployed in the area. Traffic data are timely integrated through an approximate join similar to that adopted for climate data integration.

2.3.2 Data representation

To perform association rule-based analyses, heterogeneous data acquired from sensors are tailored to a relational data format, prepared to the next mining step by means of established preprocessing techniques.

Relational data model A relational dataset is a set of records. Each record r_i corresponds to a given time period T_i and it collects pollutant, meteorological, and traffic data acquired in T_i. A record is a set of items, where an item is a pair (*attribute*, *value*). While *attribute* is the description of a data feature of interest in the context under analysis, *value* is the value assumed by the corresponding attribute. Each record contains at most one item per data attribute (i.e., multiple attribute values in the same record are not allowed).

Let's consider the following attributes in the context of analysis of this study. (i) Pollutants: particulate matters PM_{10} and $PM_{2.5}$, Ozone (O_3), Nitrogen dioxide (NO_2), Carbon Monoxide (CO), and Benzene C_6H_6. (ii) Meteorological factors: wind direction, wind speed, temperature, humidity, pressure, UV radiations, precipitations. (iii) Traffic conditions: numbers of gasoline engine, diesel engine, natural gas, electric, and hybrid vehicles.

Data discretization Continuous attributes are unsuitable for use in association rule-based analyses, because their values are very unlikely to frequently occur in the analyzed dataset. For this reason, a data discretization step is applied prior to running the association rule mining process.

Pollutant concentration levels are discretized into different categories named with colors from green to red according to the severity of the level range from the point of view of the citizen's health. Currently, categories have been defined based on the classification given the Italian ARPA Piemonte agency responsible for environment protection in the Piemonte region [7] (e.g., *blue* and *green* imply non-critical levels, while *orange* and *red* indicate highly critical levels).

The *traffic indicator* values are uniformly discretized by using the equal-width discretization algorithm available in the RapidMiner suite [64]. For example, the humidity values (expressed in $\frac{kg}{m^3}$) are discretized as *very low* between zero and 20, *low* between 20 and 40, *medium* between 40 and 60, *high* between 60 and 80, *very high* between 80 and 100, while for the UV radiations (expressed in $\frac{W}{m^2}$) the discretization levels are the following ones: *very low* between zero and 0.9, *low* between 0.9 and 2.9, *medium* between 2.9 and 5.9, *high* between 5.9 and 7.9, *very high* between 7.9 and 10.9, *extremely high* above 10.9.

Concerning the *meteorological attributes*, the wind speed is discretized, according to the Beaufort scale, in 13 different levels, from *Calm* (level 0) to *Hurricane force* (level 12), while the other attributes are discretized into standard value ranges. For example, the wind direction degrees are discretized based on the classical cardinal points (i.e., as *north-east*, *east*, *south-east*, *south*, *south-west*, *west*, *north-west*, and *north*).

2.3.3 Taxonomy generation

To analyze pollutant data at different abstraction levels a taxonomy is built on top of relational data. A taxonomy is a set of is-a hierarchies, each one referring to a specific data attribute. Each hierarchy aggregates all the values assumed by the corresponding attributes into higher-level concepts in a tree-based structure. For example, let us consider the wind direction attribute. Low-level (discrete) values *north-east*, *east*, and *south-east* are generalized as *east-side*, while values *south-west*, *west*, *north-west* are generalized as *west-side*. An item consisting of a pair (*attribute*, *generalized value*), where *generalized value* is an higher-level aggregation occurring in the input taxonomy, will be hereafter denoted as *generalized item*. For example, based on the hierarchy on the wind direction attribute, item (*wind direction*, *north-west*) can be generalized as the corresponding generalized (higher-level) item (*wind direction*, *west-side*).

Taxonomies are analyst-provided. They can be either given by the domain expert based on their common knowledge or generated semi-automatically by applying multiple discretization runs on the same attribute domain. To generate the taxonomy, further discretization runs on top of discretized record values are applied. Pollutant concentration level categories (e.g., *blue* and *green*) are further discretized as *non-critical*, *fairly-critical*, and *highly critical* according to the level of severity of the pollutant from the point of view of the citizen's health. Traffic levels are discretized as *low*, *medium*, and *high*. Meteorological values are further discretized into upper-level categories (e.g., *east-side*, *west-side*). Hourly timeslots are categorized as 4-hour, and 8-hour timeslots (e.g., early morning, evening), while dates are aggregated into the corresponding week of the month (e.g., 1st week of December) , month of the year (e.g., December), and season (e.g., winter).

Since the process of taxonomy generation is semi-automatic, the taxonomy may consist of hierarchies of different height. To avoid bias in the next association rule mining process, the hierarchies in the taxonomy are balanced by equalizing the corresponding heights. As discussed in [15], the aforementioned procedure is established in generalized pattern mining. To this aim, artificial root nodes are added to lower-height hierarchies until all their heights match those of the highest one.

2.3.4 Data analyses

This block aims at discovering interesting associations between pollutant levels and environmental factors (meteorological and traffic conditions), in the form of generalized association rules. Association rules are illustrated in the Section 2.2.2.

To analyze pollutant data at different granularity levels, the itemset definition, reported in the Section 2.2.1, can be straightforwardly extended to the case in which data are enriched with a taxonomy. A generalized itemset [74] is defined as a set of items and/or generalized items. Note that traditional (non-generalized) itemsets are special case of generalized itemset in which all items assume non-aggregated values according to the input taxonomy. For example, generalized itemset {($PM_{2.5}$, *highly critical*),(*wind direction*, *east-side*)} generalizes the former itemset by aggregating item values according to the hierarchies built on the $PM_{2.5}$ and *wind-direction* attributes (see Section 2.3.2).

A generalized item *matches* a given record if its value corresponds or is an aggregation of the value of any item of the record (at any abstraction level). For example, generalized item (*date*, *Winter*) matches a record containing item (*date*, *December 1st, 2013*). The support of a generalized itemset in a relational dataset is an established quality index which is computed as the percentage of dataset records matched by all of its items.

A *generalized association rule* [74] is an implication $A \rightarrow B$, where A and B are disjoint generalized itemsets, i.e., generalized itemsets having no attributes in common. Hereafter, A and B will be denoted as the antecedent and consequent of rule $A \rightarrow B$, respectively. Generalized rules are characterized by three main quality indeces, i.e., support, confidence, and lift, as illustrated for the regular association rules in the Section 2.2.2.

Generalized rule mining The *GECKO* system extracts from the prepared relational dataset all the generalized rules that satisfy a minimum support threshold *minsup* and a minimum confidence threshold *minconf*. Since both positively and negatively correlated rules are considered for in-depth analysis, no minimum/maximum lift threshold is enforced. While positively correlated rules represent strong correlations among data items, negatively correlated ones represent implications that hold less than expected.

The algorithms The generalized association rule mining task is accomplished as a two-step process: (i) Frequent generalized itemset mining, which extracts all the generalized itemsets whose support is above *minsup*. (ii) Generalized association rule mining, which extracts all the generalized rules whose support is above *minsup* and whose confidence is above *minconf*, starting from the previously mined set of frequent itemsets.

To accomplish Step (i), GenIO, an algorithm specialized in the generalized frequent itemsets extraction, is integrated in the *GECKO* system, while to perform Step (ii) the

RuleGen procedure integrated in the Apriori algorithm is adopted. To prevent generating all the possible item combinations, GenIO generates a subset of potentially interesting generalized itemsets covering, at a higher abstraction level, most of the information represented by infrequent itemsets. More details on the GenIO and Apriori algorithms are given in [9] and [3], respectively.

Regarding the complexity analysis of the various steps, the following statements apply: the steps of *Data Integration*, *Data Representation*, and *Taxonomy Generation* are all linear to the number of records times the average dimensionality of the dataset. The *Data analyses* phase instead, characterized by the generation of the generalized association rules through the above-mentioned GenIO and RuleGen algorithms, has a larger complexity.

The generalized frequent itemset mining step has a complexity that is combinatorial with the number of items $|S|$ plus the number of all their corresponding generalized versions $|G|$, i.e., $\mathcal{O}((|S| + |G|)(2^{|S|+|G|-1}))$ in the worst case. Nevertheless, the complexity is greatly reduced when a support threshold greater than one is chosen. Additionally, the optimizations already integrated into the GenIO algorithm further reduce the complexity applying an opportunistic strategy: a generalized item is added just when at least one of the lower level items is infrequent, and father-child combinations are avoided in the same itemset. The RuleGen algorithm complexity follows $\mathcal{O}(2^f - 2)$ in the worst case, where f represents the frequent itemsets generated by the previous step.

Rule categorization Exploring the results of the rule extraction process can be a challenging task, because the number of mined rules can be very high. To ease the manual exploration of the result, rules are categorized into a subset of classes according to the represented knowledge. Thus, experts can focus their attention on the subset of classes of interest.

Rule class Pollutant-Pollutant (PP). This class comprises all the rules that contain only items belonging to attributes related to pollutant concentration levels. Rules (PM$_{10}$, *red*) → (PM$_{2.5}$, *red*) and (PM$_{10}$, *yellow*) → (O_3, *non-critical*) are examples of rules of class PP. These rules can be useful for identifying correlations between the concentration levels of multiple pollutant and, thus, to plan targeted actions (e.g., planning air monitoring protocols, saving measurement costs).

Rule class Pollutant-Traffic (PT). This class comprises all the rules that contain items related to pollutant concentration levels and traffic conditions (e.g., number of gasoline engine vehicles). Rule (PM$_{10}$, *red*) → (*number of gasoline engine vehicles*, *high*) is an example of rules of class PT. These rules can be useful for correlating pollutant concentrations with the transit of different types of vehicles in the city. Based on these correlations, municipality managers may redesign traffic policies with the aim at reducing pollutant concentrations.

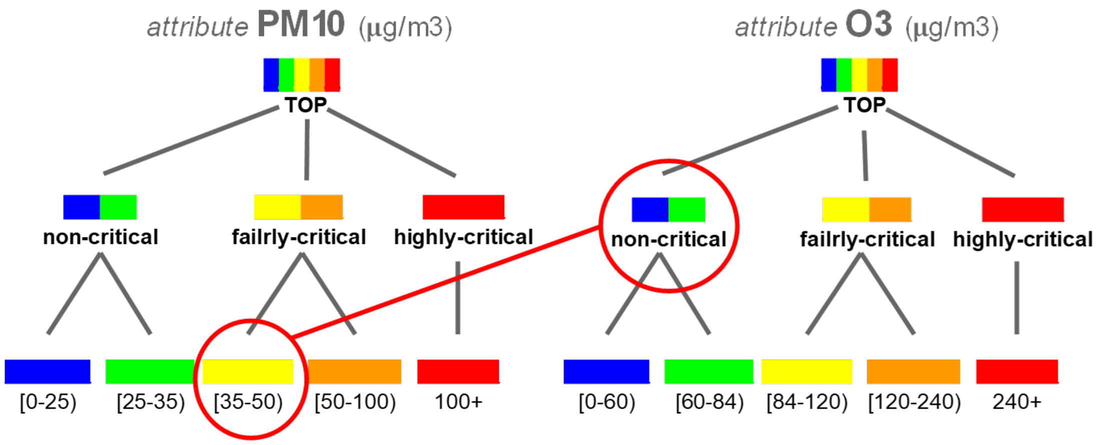


Figure 2.4: An example of the generalized cross association rules extracted.

Rule class Pollutant-Meteo (PM). This class comprises all the rules that contain items related to pollutant concentration levels and meteorological conditions (e.g., temperature, humidity). Rule (PM_{10}, *red*) → (*temperature*, *very cold*) is an example of rules of class PM. These rules can be useful for correlating pollutants with climate conditions. Hence, they can identify meteorological conditions in which specific pollutants should be carefully monitored to prevent unsafe air conditions.

Rule class Pollutant-Date (PTE). This class comprises all the rules that contain items related to pollutant concentration levels and temporal attributes (e.g., date, time). Rule (PM_{10}, *red*) → (*date*, *morning*) is an example of rules of class PTE. These rules can be useful for correlating the levels of pollutants with specific time periods or time slots. Based on these rules, in-site monitoring actions can be scheduled at the timeslots at which high pollutant concentrations are most likely.

More complex rules, e.g., class Pollutant-Meteo-Traffic (PMT), can be extracted as well. They represent implications between pollutant levels and a combination of environmental conditions (e.g., rule (PM_{10}, *red*) → {(*temperature*, *very cold*), (*number of gasoline engine vehicles*, *high*)}).

Classes are manually explored by domain expert to infer potentially interesting knowledge from the contained rules. To consider first the top correlated combinations of pollutant data, rules are sorted by decreasing lift. See Figure 2.4 for a graphical example of the association rules extracted.

2.3.5 Experimental validation

The proposed approach was validated on real data acquired in Milan, that is one of the largest and most important Italian Smart Cities. To perform the analyses, two open datasets collecting the sensor measurements were considered acquired over a 12-month time period (i.e., over year 2013). The generalized rules were extracted by using the Python implementation of the GenIO algorithm [9] provided by the respective authors.

Frequent and high-confidence rules were extracted, which represent recurrent and potentially reliable correlations among multiple data items. Whenever not otherwise specified, the following standard parameter setting will be considered: *minsup*=1% and *minconf*=20%. The experiments were performed on a quad-core 3.30 GHz Intel Xeon workstation with 16 GB of RAM, running Ubuntu Linux 12.04 LTS.

Datasets The analyzed datasets collect pollutant concentrations, climate conditions and traffic levels of different categories of vehicles acquired in the central area of Milan (zone C). The main dataset attributes and the taxonomy used to aggregate the data values at multiple abstraction levels are described in Section 2.3.2. The first dataset, hereafter denoted as *Daily*, collects the daily pollutant levels measured on a daily basis as well as the environmental information about meteorological and traffic conditions. The second dataset (*Hourly*) collects the hourly pollutants levels together with the corresponding environmental conditions.

Pollutant data were gathered by the ARPA Lombardia [7] through monitoring stations equipped with a set of sensors, each one measuring a different pollutant. Meteorological measurements were collected through the Weather Underground web service [86], which gathers data from a geo-referenced network of Personal Weather Stations (PWSs) registered by users. Three PWSs located in the city center were considered. Traffic data were provided by the Municipality of Milan[1]. They consist of the counts of the number of vehicles entering in the central area of Milan, separately for each category of vehicles.

Knowledge discovery The extracted rules were categorized, according to the type of item correlations they represent, into the classes described in Section 2.3.4. For each class, a subset of the most interesting rules extracted from both datasets is reported in Table 2.4. Some of the selected rules recall established correlations between pollutants and environmental factors, discussed by previous works on the topic (e.g. [5, 24, 75]). However, as discussed below, the mined generalized association rules provide more insightful information than the ground knowledge, because they indicate the levels at which pollutants, climate factors, and traffic conditions are actually influenced with each other.

Correlations between pollutant levels (Class PP). When particulate matters PM_{10} and $PM_{2.5}$ have the same criticality level (e.g., *yellow*, *green*), a strongly positive pairwise item correlation appears (see Rules R_1-R_3). The positive rule lift values confirm that

[1] http://dati.comune.milano.it/

Table 2.4: Rule examples.

ID	Dataset	Rules	Sup (%)	Conf (%)	Lift
Class PP					
R_1	Daily	(PM_{10},*yellow*) → ($\mathrm{PM}_{2.5}$,*yellow*)	9.7	72.9	5.4
R_2	Daily	(PM_{10},*green*) → ($\mathrm{PM}_{2.5}$,*green*)	27.0	66.7	2.2
R_3	Daily	(PM_{10},*blue*) → ($\mathrm{PM}_{2.5}$,*blue*)	37.78	95.78	2.0
R_4	Daily	(PM_{10},*green*) → ($\mathrm{PM}_{2.5}$,*blue*)	10.3	25.2	0.52
R_5	Daily	(PM_{10},*yellow*) → {(O_3,*non-critical*), (CO,*highly critical*)}	7.0	52.1	5.1
R_6	Daily	(PM_{10},*yellow*) → {(O_3,*non-critical*), (CO,*highly critical*)}	5.6	41.7	5.0
R_7	Hourly	(O_3,*highly critical*) → (NO_2,*non-critical*)	5.9	51.6	1.5
R_8	Hourly	(O_3,*non-critical*) → (NO_2,*highly critical*)	11.3	24.1	1.7
Class PM					
R_9	Daily	{(*precip.*,*drizzling*), (PM_{10},*orange*), ($\mathrm{PM}_{2.5}$,*red*)} → {(*temp.*,*very cold*), (*CO*,*fairly high*)}	1.1	40.0	20.1
R_{10}	Daily	{(*temp.*,*very cold*), (*CO*,*fairly high*) → {(O_3,*blue*), ($\mathrm{PM}_{2.5}$,*red*)}	1.4	55.6	2.2
R_{11}	Daily	{(*precip.*,*no rain*), (*temp.*,*hot*), (C_6H_6,*non-critical*) } → ($\mathrm{PM}_{2.5}$,*green*), (O_3,*non-critical*)}	1.1	66.7	20.7
Class PTE					
R_{12}	Daily	(PM_{10},*green*) → (*date*,*weekday*)	30.3	74.2	1.1
R_{13}	Daily	(PM_{10},*green*) → (*date*,*weekend*)	10.6	25.9	0.9
R_{14}	Daily	(*date*,*spring*) → (PM_{10},*green*)	5.6	55.6	1.4
R_{15}	Daily	(*date*,*winter*) → (NO_2,*blue*)	52.8	54.3	1.2
R_{16}	Hourly	(*hourly time period*,*late afternoon*) → (NO_2,*fairly critical*)	6.5	39.1	1.4
R_{17}	Hourly	(*hourly time period*,*night*) → (NO_2,*fairly critical*)	9.7	29.1	0.9
R_{18}	Hourly	(*hourly time period*,*late morning*) → (NO_2,*fairly critical*)	6.5	39.9	1.4
R_{19}	Hourly	{(*date*,*winter*), (O_3,*non-critical*)} → (NO_2,*highly critical*)	5.9	26.9	1.9
Class PTR					
R_{20}	Daily	(*num. diesel engine vehicles*,*medium*) → (PM_{10},*fairly high*)	9.7	70.0	1.8
R_{21}	Daily	(*num. diesel engine vehicles*,*high*) → (PM_{10},*green*)	14.7	34.4	0.9
R_{22}	Daily	(*num. gasoline engine vehicles*,*high*) → (CO,*low*)	9.2	73.3	1.4

the pollutant levels co-occur more than expected. On the other hand, opposite pollutant levels (e.g., in Rule R_4 *green* for PM_{10} and *blue* for $\mathrm{PM}_{2.5}$) show a negative correlation, meaning that the occurrence of a pollutant level implies the absence of the other one. Beyond pointing out the established correlation between the concentrations of particulate matters PM_{10} and $\mathrm{PM}_{2.5}$, rules R_1-R_3 provide additional and potentially useful information, because they indicate the levels at which the two pollutants are most likely to be correlated with each other. The confidence of the aforesaid rules indicates the probability of occurrence of a pollutant level given the level of another pollutant. For example, according to the confidence value of Rule R_1, the probability of having level *yellow* for $\mathrm{PM}_{2.5}$ given level *yellow* of PM_{10} is approximately 73%. These probabilities can be useful for planning air quality monitoring activities. For example, if two pollutants have a high probability of sharing levels *orange* and *red*, a critical concentration of one pollutant should trigger prompt monitoring actions targeted to the other pollutant as well.

Rules R_5 and R_6 show the correlation between PM_{10} and the pair Ozone (O_3) and carbon monoxide (CO). For example, a fairly critical level of PM_{10} (*yellow*) is often related to a *non-critical* level of Ozone. Rules R_5 and R_6 contain two generalized items each, i.e., (O_3, *non-critical*) and (CO, *highly critical*), which aggregate the information provided by their corresponding lower-level items (O_3, *blue*) and (CO, *orange*) at a higher abstraction level.

Rules R_7 and R_8 show the inverse relationship between the levels of Nitrogen dioxide (NO_2) and Ozone (O_3). The oxidation in atmosphere of Nitrogen oxide, Ozone, and other pollutants produces Nitrogen dioxide. Hence, a high concentration of Nitrogen dioxide is often associated with a low concentration of Ozone (and vice versa).

Correlations between pollutant levels and meteorological factors (Class PM). Rules R_9-R_{11} show a positive correlation between the external temperature values and the concentration of particulate matters PM_{10} and $PM_{2.5}$, Carbon Monoxide, and Ozone (O_3). For example, according to Rule R_{11}, when the temperature is cold and the precipitations are too weak to disperse the pollutants in the air, the concentrations of the aforesaid pollutants are likely to be fairly critical (i.e., levels *fairly high* for *CO* and *red* for $PM_{2.5}$, respectively). Conversely, for pollutant NO_2 an opposite trend comes out. In fact, based on generalized rule R_{15} (reported in the following for rule class PTE) the concentrations of NO_2 is low (level *blue*) in winter. On the other hand, when the temperature is hot or very_hot, the pollutant levels are likely to *non-critical* (i.e., *green* or *blue*).

Correlations between pollutant levels and time (Class PTE). Based on the lift value of Rules R_{12} and R_{13} (approximately one), pollutant levels and day of the week categories (i.e., weekday, weekend) seem to be statistically independent with each other. On the other hand, a correlation between pollutant levels and seasons holds. This effect seems to be an indirect consequence of the strong correlation holding between pollutant levels and temperature values.

Correlations between pollutant levels and traffic conditions (Class PTR). The effect of traffic flows on the air quality can be investigated by analyzing the rules involving pollutant levels and traffic conditions. For example, rules R_{20}-R_{22} show the correlation between the presence of many diesel engine vehicles in the city area and the concentration of PM_{10}. According to these rules, the presence of a medium/high number of vehicles is negatively correlated with a low concentration of PM_{10} and positively correlated with a fairly high concentration of the same pollutant. Conversely, a high number of gasoline engine vehicles is positively correlated with a low concentration of Carbon Monoxide. The latter rule indicates that the presence of diesel engine vehicles is critical for PM_{10} emissions, whereas gasoline engine vehicles does emit a significant amount of Carbon Monoxide.

2.4 *ARQUATA*: discovering air quality patterns in urban environments

Figure 2.5 summarizes the main blocks of the proposed *ARQUATA* engine, which relies on three main components: (i) Data integration and preparation. (ii) Air quality pattern mining. (iii) Reporting.

2.4.1 Data integration and preparation

Different geo-referenced sensor networks are exploited to periodically monitor the concentration levels of the main air pollutants in the urban environment. Since each network may adopt a different timeline in sampling pollutant concentrations in different city areas, the acquired measures are then integrated by considering a common time granularity. For the sake of simplicity, hereafter daily time granularities will be considered. Each pollutant level is characterized by (i) a sampling timestamp and (ii) a set of geo-coordinates of acquisition (i.e., latitude and longitude). Pollutant concentrations are first cleaned to remove missing values and incorrect readings, normalized to make the pollutant concentrations distributions uniform with each other, and then integrated into different contextual datasets, one for each context under analysis. A *contextual dataset* collects sensor measurements acquired from a subset of sensors corresponding to a specific spatial or functional domain (e.g., the sensors belonging to the same district, industrial area, or residential zone). Each contextual dataset consists of a set of records, one for each sampling timestamp. Each record comprises all the pollutant concentrations acquired at the corresponding timestamp.

2.4.2 Air quality pattern mining

This step focuses on extracting patterns characterizing the air quality. Air quality pattern extraction relies on itemset mining, an established data mining technique to discover significant yet hidden correlations among large datasets [2]. Using itemset mining, several types of air quality patterns can be extracted (e.g., closed itemsets, generalized itemsets, emerging patterns). In this study, the focus is on a specific type of itemset, namely the weighted frequent itemsets (see Section 2.2.3), to discover combinations of pollutants whose concentration levels are all averagely critical in the considered time period. For each pollutant a reference *critical level* is given by the domain expert (e.g., the critical level specified by law). *Weighted itemsets* are sets of items, each one representing a distinct pollutant (e.g., {$PM_{2.5}$, PM_{10}}), and extracted from contextual datasets. To this aim, for each record a weight is associated with each pollutant occurring in the itemset. Weights are computed as the percentage variation of the corresponding pollutant concentration with respect to the critical level. For example, if the critical level of $PM_{2.5}$ is 10 and the $PM_{2.5}$ level at the considered timestamp is 15 then the item weight

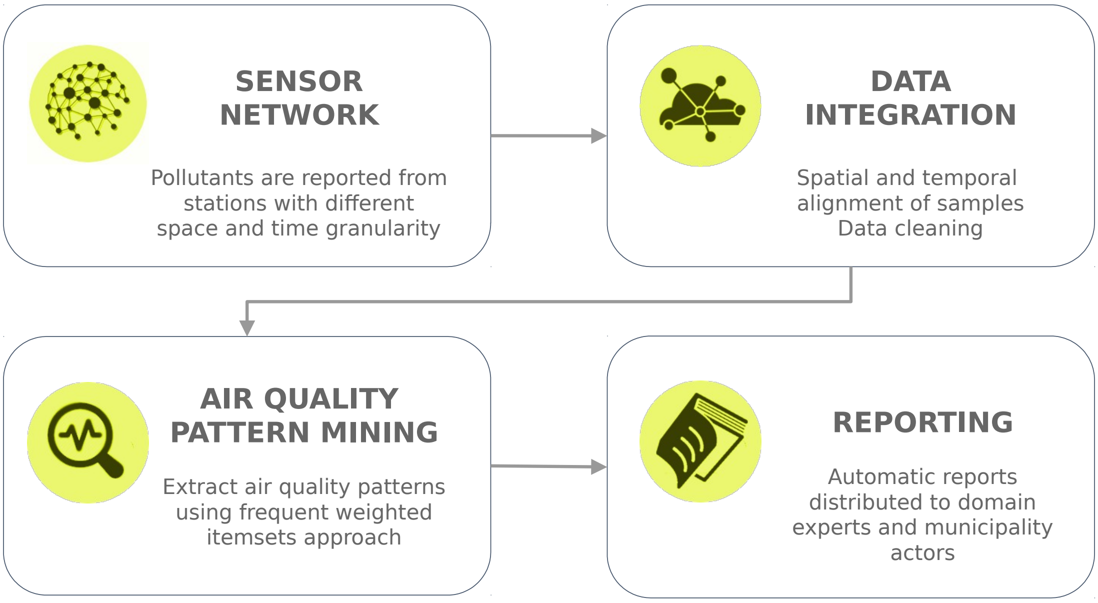

SENSOR NETWORK
Pollutants are reported from stations with different space and time granularity
DATA INTEGRATION
Spatial and temporal alignment of samples Data cleaning
AIR QUALITY PATTERN MINING
Extract air quality patterns using frequent weighted itemsets approach
REPORTING
Automatic reports distributed to domain experts and municipality actors

Figure 2.5: The *ARQUATA* engine.

is 50%, because the level exceeds the critical level by 50%. Weighted itemsets are characterized by a notable quality measure, denoted *critical gap*. It indicates the average percentage variation of the least pollutant weight (i.e., the minimal percentage variation of the corresponding pollutants). Using the algorithm proposed in [18], from each contextual dataset all the weighted frequent itemsets are extracted, which are combinations of pollutants whose critical gap is above a given (user-specified) threshold. For example, if the critical gap threshold is set to 10%, itemset $\{PM_{2.5}, PM_{10}\}$ is extracted if both $PM_{2.5}$ and PM_{10} have an average percentage variation above 10%.

2.4.3 Reporting

Since the correlations among pollutant levels can vary over time and space, users are commonly interested in monitoring their temporal and spatial evolution. To effectively characterize the underlying trends in air pollution-related data, the results of different mining sessions on data acquired from the same context are compared with each other and reported to citizens and municipality actors. Technical reports directed to domain experts include (i) Periodic summaries on the latest mining results (e.g. the top-10 combinations of pollutants in order of decreasing critical gap). (ii) A comparison between the mining results achieved in different time periods or in different city areas (e.g., the combinations of pollutants in common for all years/areas, the combinations appearing in just one year/area). Based on the above results, experts may process the extracted patterns and schedule the periodic forwarding of higher-level reports, directed to either citizens or municipality actors, discussing (i) the currently critical levels of pollutants for each city area/district (e.g., in the last winter the $PM_{2.5}$, PM_{10} concentrations were averagely critical at the same time with a critical gap higher than 30%), (ii) the most significant temporal trends in pollutant level variations (e.g., the criticality of the levels of $PM_{2.5}$, PM_{10} is constantly decreasing from year 2014 on), (iii) the most significant spatial trends (e.g., the criticality of the levels of $PM_{2.5}$, PM_{10} is more significant in the city center). Reports are tailored to end user roles (e.g., city major, assessor, citizen) and the corresponding authorities.

2.4.4 Experimental results and discussion

The proposed approach was validated on real pollutant data acquired on a daily basis by the ARPA Lombardia in the central area of Milan (Italy). The seasonal trends in pollutant data acquired in the year 2013 were analyzed.

The data of this particular year has been chosen as a reference case study for a number of reasons. First, years 2013-2015 were selected due to the higher number of sensors in comparison with the previous years. Then, each yearly dataset was evaluated and the one of 2013 was selected as a sample with the lowest percentage of missing values, which typically occur, for example, due to malfunction of the sensors or interrupted communication with the monitoring station.

For each pollutant critical level, the highest safe concentration (i.e., the upper border of the green zone available on the ARPA website) was considered as a reference.

To characterize seasonal trends in pollutant data, all the weighted patterns whose critical gap is higher than 30% (see Table 2.5) were extracted.

Autumn and winter are seasons characterized by similar trends. In both seasons the levels of $PM_{2.5}$ and PM_{10} are significantly higher than the corresponding critical levels (e.g., the level of $PM_{2.5}$ in autumn and winter are on average 90.62% and 120.72% higher than the critical level, respectively).

The combination of pollutants that simultaneously exceed the critical level in most cases are particularly interesting. For instance, based on the critical gap of pattern {PM_{10},$PM_{2.5}$}, in the winter season pollutants PM_{10} and $PM_{2.5}$ simultaneously exceed their critical level by 93.49%. Since the two critical conditions appeared to be strongly correlated with each other, targeted actions may be performed to counteract them. A fair correlation between the triple of pollutants NO_2, PM_{10}, and $PM_{2.5}$ appeared as well. However, the critical gap of patterns {NO_2,PM_{10}} (53.56%) and {NO_2,$PM_{2.5}$} (50.51%) are significantly lower than those of {PM_{10},$PM_{2.5}$} (93.49%).

The above results are consistent with the expectation, because the association between PM_{10} and $PM_{2.5}$ is established. Note that pattern {PM_{10},$PM_{2.5}$} is extracted only in autumn and winter, because the pollutant concentrations are probably related to the use of heating systems. The municipality may foster the use of new generation heating systems, characterized by lower pollutant emissions through targeted actions and then use the framework to analyze the effect of the performed actions on the air quality.

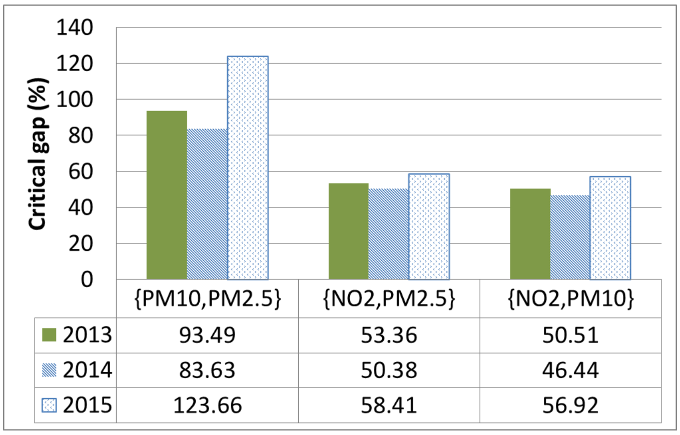


Figure 2.6: Comparison between winters 2013, 2014, and 2015.

For example, a yearly comparison between the critical gaps of patterns {PM_{10},$PM_{2.5}$}, {NO_2,$PM_{2.5}$}, and {NO_2,PM_{10}} among years 2013, 2014 and 2015 is reported in Figure 2.6. The comparison showed a slight decrease from 2013 to 2014, but a slight increase from 2014 to 2015.

Autumn

itemset	*Critical gap (%)*
{$PM_{2.5}$}	90.62
{PM_{10}}	88.09
{PM_{10},$PM_{2.5}$}	77.64
{NO_2}	54.64
{NO_2,PM_{10}}	38.37
{NO_2,$PM_{2.5}$}	35.51
{NO_2,$PM_{2.5}$,PM_{10}}	34.12

Winter

itemset	*Critical gap (%)*
{$PM_{2.5}$}	120.72
{PM_{10}}	99.49
{PM_{10},$PM_{2.5}$}	93.49
{NO_2}	76.87
{NO_2,$PM_{2.5}$}	53.36
{NO_2,PM_{10}}	50.51
{NO_2,$PM_{2.5}$,PM_{10}}	45.92

Spring

itemset	*Critical gap (%)*
{NO_2}	30.07

Summer

itemset	*Critical gap (%)*
no patterns satisfying the threshold	

Table 2.5: 2013 seasonal analysis: weighted frequent itemsets. Critical gap thresh. 30%.

Computational Complexity To analyze the computational complexity of the *ARQUATA* framework, it is useful to consider two different steps: *pre-processing phase*, which has a linear complexity, i.e. $\mathcal{O}(n\bar{t})$, where n is the number of transactions and $\bar{t}$ is the average length of the transactions in the dataset, and the *pattern extraction phase*. The latter, as stated in [2], is based on FP-growth [39], the state-of-the-art algorithm used for frequent itemset mining. The complexity of this FP-growth-like algorithm is linear when considering the number of mined itemsets, which in its turn could grow in the worst case exponentially according to the formula $\mathcal{O}(2^{|S|})$, where $|S|$ is the number of distinct items.

2.5 Summary

In this chapter two novel data mining systems, *ARQUATA* and *GECKO*, designed to analyze air pollution-related data, have been described. Their aim is to report interpretable descriptive analytics in terms of critical conditions to citizens and municipality actors.

The *ARQUATA* system, is focused on mining air quality patterns from air pollution data. In particular, a specific type of patterns, namely the weighted frequent itemsets, has been used to identify combinations of pollutants that are, on average, in a critical condition.

In its turn, *GECKO* leverages the expressiveness of generalization combined with association rules analysis to discover interesting and multiple-level correlations among a large variety of open air pollution-related data. Specifically, correlations among pollutant levels, traffic, and climate conditions are discovered and analyzed at different abstraction levels. The knowledge extraction process is driven by a taxonomy to generalize low-level measurement values as the corresponding categories.

Both systems, *ARQUATA* and *GECKO*, have been validated using real datasets acquired in a major Italian Smart City (i.e., Milan). The results of the experimentation demonstrate the potential of the proposed methodologies in modeling interesting correlations at different abstraction levels.

However, there is still room for improvements. For example, *GECKO* may be enriched with (i) other kinds of interesting data affecting air quality such as people's mobility and private/public transport data, the actual distance travelled by cars in the city, and fuel consumption depending on the car model and on different segments of the route (i.e., urban, extra-urban, or combined), and (ii) data mining algorithms to discover correlations among weighted air pollution-related data.

In perspective, a useful extension would be to integrate a predictive analytics module into *GECKO* that would leverage classification techniques to forecast air-quality levels in city areas in which the coverage of the pollution monitoring stations is poor or totally absent. In particular, an associative classifier – a supervised technique known for producing accurate predictive models – could be a good choice for the possibility to partially reuse the work already done.

Chapter 3

Green Urban Mobility: Analyses of Bike Sharing Data

In recent years municipalities have fostered alternative ways of public transportation in order to reduce pollution and traffic congestion [70, 53, 76, 58, 37] . Bicycle sharing systems [85, 69] are a notable example of eco-friendly transportation systems, where citizens can rent bicycles on a short-term basis. Bikes are retrieved from stations spread throughout the city and each station has a maximum capacity as it is equipped with a fixed number of docks. Citizens can rent a bicycle parked at any station and return it to any other station with free docks. However, to achieve a satisfactory user experience, system managers should carefully monitor the level of occupancy of the stations. For example, if a station is frequently overloaded at peak hours then a rebalancing action should be scheduled in order to move some of the parked bicycles to any station located in the neighborhood. In case the problem is more severe, managers may decide to expand the station to fit the increasing demand.

Stations are geo-referenced and equipped with sensors to constantly monitor their level of occupancy. Each station tracks the occupancy levels of its docks, thus providing geo-referenced time series data. These data acquired from stations can be collected and stored in a unique repository and analyzed by means of machine learning techniques.

This chapter presents a novel exploratory data-driven methodology, named *B*ike Station Ov*E*r*L*oad Ana*L*yzer (*BELL*), which analyzes the occupancy levels of the stations of a bicycle sharing system. The aim is to identify situations of dock overload in multiple stations which could lead to either service disruption or low customer satisfaction. For example, when all the docks in a station are occupied, users have to move to a nearby station to park their bike. By gathering insightful information regarding occupancy levels of multiple stations, domain experts can effectively apply targeted actions in order to avoid and/or limit the unpleasant situations described above. For instance, the mobile application of the system may recommend alternative nearby stations with free docks. Furthermore, the maintenance service may rebalance the number of bikes in each station thus avoiding overloaded conditions.

In the *BELL* methodology occupancy level data acquired from the geo-referenced stations are analyzed to discover a new type of pattern, called *Occupancy Monitoring Pattern* (OMP). OMPs describe in a concise way situations of imbalance in the occupancy levels of spatially correlated stations. Specifically, OMPs model two complementary dock overload situations: (i) Situations in which a set of stations are overloaded in an alternate fashion (hereafter denoted as *intermittent situations*), and (ii) Situations in which the docks of a set of stations are frequently overloaded at the same time (hereafter denoted as *critical situations*). To consider the spatial correlation between the occupancy level of different stations, spatial constraints can be enforced to represent groups of nearby stations in OMPs (i.e., stations within a limited geographical distance).

Intermittent and critical situations cause disservices with varying degrees of severity for end users. Intermittent situations indicate an imbalance in station usage which could be addressed by proposing alternative nearby stations to end users or by periodically re-positioning the bicycles in the neighborhood. Conversely, critical situations indicate that a given area is temporarily inaccessible for parking bikes because all the stations in the area are in a dock overload situation. The latter (more severe) situation can be addressed, for example, by increasing the number of available docks in the stations, or by moving bikes to the not fully occupied stations located in other city areas.

The generated OMPs are explored to discover significant intermittent and critical situations. The exploration is driven by two ad hoc quality indices introduced in this study, namely the *intermittence* and the *criticality* indices, which allow domain experts to focus on the most severe warnings.

OMPs permit a spatio-temporal exploration of critical and intermittent situations. Since stations are geo-referenced, OMPs display the city areas *where* disservices are likely to occur. Moreover, since OMPs can be related to specific time periods, they allow experts to identify *when* these disservices are likely to occur. Building spatial and temporal hierarchies on top of the data allow exploring the OMPs at different abstraction levels. For instance, a daily time granularity allows spotting larger-scale phenomena, such as the most unbalanced days in terms of dock occupancy, while an hourly granularity can help determine at which hour to schedule the re-balancing of the bikes among the stations.

BELL has been thoroughly evaluated using a real dataset acquired from the bicycle sharing systems of two important cities, i.e., Barcelona (Spain) and New York (USA). The experimental results demonstrated the effectiveness of *BELL* in identifying useful knowledge regarding the spatio-temporal distribution of possible service disruptions for end users of bicycle sharing systems. Possible scenarios of usage of the extracted patterns aimed at supporting maintenance activities and improving user experience were considered.

This chapter is organized as follows. Section 3.1 overviews the literature. Section 3.2 presents and thoroughly describes the proposed approach. Section 3.3 experimentally evaluates the performance of the implemented *BELL* methodology on data acquired in real urban environments. Section 3.4 discusses the policy implications of the presented results, and 3.4 summarizes this work and presents future developments.

3.1 Related work

The analysis of urban data related to bicycle sharing systems has already been addressed in previous studies. Specifically, in this field the main branches of research can be categorized as follows. (i) Grouping stations based on their usage profile [22, 66, 21, 57]. (ii) Predicting future station occupancy levels [42, 34, 36, 40, 41]. (iii) Repositioning bicycles between the stations [84, 65, 82, 68, 73, 61].

Branch (i) focuses on identifying groups of stations with different usage profiles by applying unsupervised machine learning techniques (e.g., clustering [22]). To characterize station usage, temporal features [22], spatial features [66], or a mix of the above [21] are considered. Instead of partitioning the set of stations into disjointed groups according to their common usage pattern, the methodology proposed in this study focuses on locating sets of nearby stations showing a critical or alternate usage profile (e.g., a station is overloaded whereas the nearby station is almost empty). It is worth to mention that the information provided by OMPs, which is the core of the *BELL* methodology, cannot be obtained by any of the existing approaches.

Branch (ii) aims at forecasting the occupancy level of a station in the near future (i.e., with a time horizon between 30 minutes and 2 hours ahead) by applying supervised machine learning techniques (e.g., regression [42, 34, 36, 52], classification [40, 41]). Based on these predictions, a recommender system can be integrated into the mobile application of the provider to suggest the stations close to the user-specified point of interest with a sufficient number of free docks/available bicycles. Predictions are based not only on past occupancy levels but also on contextual information (e.g., meteorological data [52]). The main differences between the aforesaid works and the proposed approach are enumerated below. (a) Unlike the aforesaid approaches, this work does not address the problem of forecasting the station occupancy levels using supervised machine learning techniques. Conversely, it presents a methodology based on an unsupervised technique. (b) In the prediction task the aim is to forecast short-term variations in occupancy level (typically, between 30 minutes and 2 hours ahead). This study aims at identifying recurrent situations of imbalance in dock occupancy, which policy-makers may consider for scheduling medium- and long-term maintenance actions (e.g., rebalance the number of bicycles in the stations, resize the existing stations, place new stations).

Branch (iii) focuses on planning the rebalance of the bicycles in stations according to actual user demands (e.g., more bicycles close to parking areas and business centers or more free docks close to restaurants at lunchtime). The aim of these study is to support providers in improving user experience by means of rebalancing bicycles among the stations.

According to Li et al. [46] there are three ways of repositioning bicycles in a bike sharing system: static, dynamic, and user-based. The static and dynamic repositioning differs one from another with respect to the state of the bike sharing system during the operations. In the static method the rebalancing is performed when the system does not operate (e.g. night hours), while the dynamic method is applied when the system is in active use.

An example of strategy for the static rebalancing has been presented by Liu et al. [49]. The study uses an optimization model to rebalance the bikes while minimizing the distance covered and, thus, transportation costs.

In its turn, Nair et al. [57] analyzed a case of dynamic rebalancing. The authors performed a stochastic characterization of demand to design fleet-management strategies dealing with flow asymmetries. Other studies that fall into the same category employed a prediction model based on time series analysis taking into account meteorological data [10, 83].

The other way to rebalance the quantity of bicycles in stations is by encouraging a user to collaborate, i.e. to leave a bicycle at a certain station in exchange for a reward. This reward could largely vary but, for example, time slots of free usage have been evaluated by Fricker et al. [33].

The study presented in this chapter, which allows spotting overloaded or imbalanced docks occupancy situations at the desired time granularity, could be leveraged both for static and dynamic rebalancing by the bike sharing service providers. In particular, the problem addressed in this study is complementary to the optimization of the flow asymmetries [57], because detecting dock overload situations could trigger rebalance actions driven by optimization-based strategies [84, 73, 61].

3.2 Methodology

BELL is a new data mining methodology aimed at monitoring the occupancy levels of the stations in a bicycle sharing system. The main architecture blocks, depicted in Figure 3.1, are (i) *Data collection, modeling and enrichment*, (ii) *Mining Occupancy Monitoring Patterns (OMP)*, which entails discovering OMP patterns from the prepared data, and (iii) *Knowledge exploration*, which consists of exploring the extracted OMPs to discover actionable knowledge. A more thorough description of each step is given in the following sections. Table 3.1 summarizes the notation used throughout the sections.

3.2.1 Data collection, modeling and enrichment

To monitor the usage of the bicycle sharing system, the occupancy levels of all the stations are acquired at different points of time and stored into an Occupancy level dataset. Collected data are then enriched with additional spatial and temporal information needed to support the subsequent data analysis phase.

Data collection and modeling. Given a time window TW and a set TS={t_1, …, t_n} of

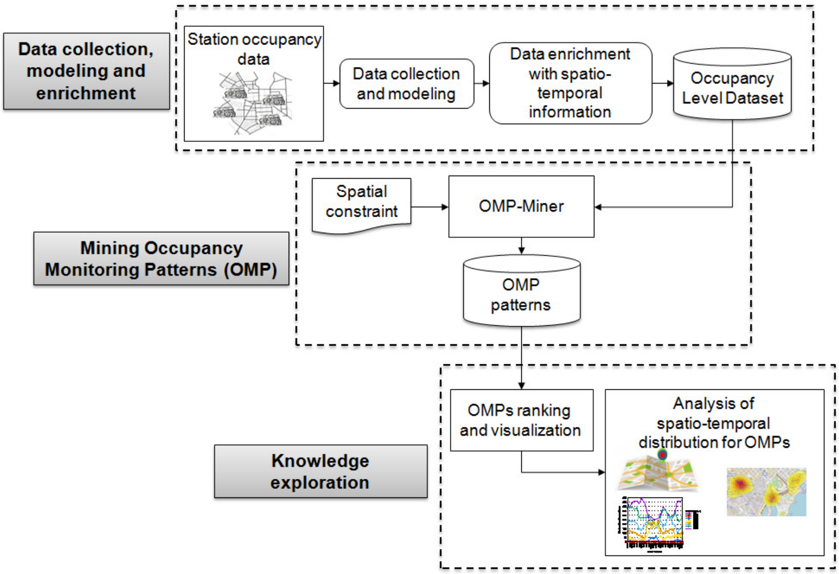

Data collection, modeling and enrichment
Station occupancy data
Data collection and modeling
Data enrichment with spatio-temporal information
Occupancy Level Dataset
Spatial constraint
OMP-Miner
Mining Occupancy Monitoring Patterns (OMP)
OMP patterns
OMPs ranking and visualization
Analysis of spatio-temporal distribution for OMPs
Knowledge exploration

Figure 3.1: The *B*ike Station Ov*Er*Load Ana*L*yzer (*BELL*) architecture.

Symbol	Description
TW	Reference time window
TS	Set of points of time in TW
s_i	Station of the bicycle sharing system
o_i^j	occupancy level of station s_j at any timestamp t_i
S	Set of stations
$\mathcal{D}$	Occupancy level dataset in relational format
R_i	Dataset record corresponding to timestamp t_i
$\mathcal{T}$	Occupancy level dataset in transactional format
RID	Record identifier
TID	Transaction identifier
P	Occupancy Monitoring Pattern
$maxdist$	Spatial constraint

Table 3.1: Notation.

points of time in TW, for each station s_i in the system the number of free parkings at each time $t_i \in TW$ is acquired and collected in a unique repository named *Occupancy level dataset* ($\mathcal{D}$). $\mathcal{D}$ is modeled as a relational dataset [77]. A more formal definition follows.

Definition 3.2.1 (Occupancy level dataset)**.** Let TW be an arbitrary time window and let TS be a set of sampling time points in TW. Let S be a set of attributes, where each attribute $s_j \in S$ represents a different station in the bicycle sharing system. Let (s_i, o_i^j) be an arbitrary pair denoting the occupancy level o_i^j of station $s_j \in S$ at a given

	Stations			Time	
Record IDentifier (RID)	s_1	s_2	s_3	**Timestamp**	**Time period**
RID_1	Overloaded	Overloaded	Overloaded	t_1	TP_1
RID_2	Overloaded	Normal	Overloaded	t_2	TP_1
RID_3	Overloaded	Overloaded	Normal	t_3	TP_1
RID_4	Overloaded	Normal	Normal	t_4	TP_1
RID_5	Normal	Overloaded	Normal	t_5	TP_2
RID_6	Normal	Overloaded	Normal	t_6	TP_2
RID_7	Normal	Normal	Normal	t_7	TP_3

Table 3.2: Example of *Occupancy Level Dataset*.

timestamp $t_i \in TS$. The record R_i indicates the occupancy levels of all the stations in S at time t_i, i.e., it is a set of pairs $\{(s_j, o_i^j)\}, \forall\, j \mid s_j \in S$. Each record is logically identified by a Record IDentifier (RID). An occupancy level dataset $\mathscr{D}$ associated with time period TW is defined as $\bigcup_{i \mid t_i \in TS} R_i$. □

Station occupancy values are categorized into two different classes to indicate the occupancy level of the station. Specifically, the measurements indicating the number of free parkings at a station are labeled as follows: (i) *Overloaded*, if the number of freely available parkings is below a given occupancy threshold *full-th*, or (ii) *Normal*, if the number of freely available parkings is equal to or above *full-th*. The occupancy level threshold *full-th* is an absolute value specified by the domain expert. Label *Overloaded* is used to denote stations with a critical occupancy level, such that end users may not find free docks for parking. Instead, label *Normal* is used to denote station conditions that should not cause a disservice to end users.

Table 3.2 shows an example of an occupancy level dataset. The dataset stores the occupancy levels of three arbitrary stations (s_1, s_2, s_3) at seven points of time (t_1-t_7). The dataset contains seven records logically identified with a RID (RID$_1$-RID$_7$). Each record includes the occupancy levels of the three stations at a given point of time.

Notice that this study will not address the complementary problem of detecting sets of underutilized stations. However, since the proposed methodology is general, it can be straightforwardly adapted to deal with this complementary problem.

Data enrichment with temporal information. The analysis of station occupancy levels at different time granularities allows system managers to investigate how overload conditions evolve over time, and to identify overload conditions that frequently happen in specific time periods. To support this analysis, a hierarchy is built on the time dimension enriching the occupancy level data with a temporal information with a coarser granularity.

In dataset $\mathscr{D}$ each record includes the occupancy levels of all the stations acquired at a different point of time $t_i \in TS$. Each record is enriched with an additional attribute specifying the corresponding *time period* TP for the point of time t_i. In the example dataset in Table 3.2, records are associated with three different time periods denoted as TP_1, TP_2, and TP_3. The granularity of the time period can be defined based on the target analysis. For example, hourly or daily time slots can be selected as reference time periods to monitor dock overload situations during the day.

Data enrichment with spatial information. To detect dock overload situations restricted to a given area, the occupancy level data is enriched with spatial information. Since all the stations in the system are geo-referenced, the geographical coordinates of all the stations in the system is collected. This information is used in the proposed approach to compute the pairwise distances between stations.

3.2.2 Mining Occupancy Monitoring Patterns

To automatically detect recurrent dock overload conditions in multiple stations, a new type of pattern is proposed, called the *Occupancy Monitoring Pattern* (OMP). OMPs represent sets of stations showing a dock overload condition which may cause a disservice to the end users of the bicycle sharing system. An algorithm is proposed in this study to efficiently extract all the OMPs of nearby stations and to compute their quality measures from a given occupancy level dataset.

The subsequent sections are organized as follows. The main properties of OMPs are presented in Section 3.2.2.1. In Section 3.2.2.2 the OMP mining problem has been addressed as an itemset mining problem, while the proposed algorithm for OMP extraction is described in Section 3.2.2.3.

3.2.2.1 OMP characterization

OMPs allow to detect dock overload conditions in multiple stations. More specifically, OMPs represent the following situations.

- *Critical situation.* The occupancy levels of a group of stations are frequently overloaded at the same time. In this case, simultaneously, *all* the stations in the group are fully occupied.
- *Intermittent situation.* The occupancy levels of a group of stations are frequently overloaded in an alternate fashion. At a given point of time, *some* stations are fully occupied whereas the other ones are almost empty. At another point of time, the occupancy level of the same stations could be opposite.

To consider only sets of nearby stations, i.e., stations with a limited geographical distance in the city area, a *spatial constraint* can be enforced. Enforcing such a constraint implies that the OMPs consist of stations with maximal geographical distance below a given (analyst-provided) threshold.

Critical situations are potentially harmful because when all the stations in the group are overloaded users cannot return the rented bicycles. In particular, the discovery of a group of overloaded stations implies that a specific city area is temporarily inaccessible. To quantitatively evaluate the severity of this issue, the measure denoted as *criticality* is introduced. This measure counts the number of recorded timestamps (i.e., the number of dataset records) at which *all* the stations of the considered OMP have a critical level of occupancy.

Intermittent situations are potentially harmful as well, because the stations in the group are overloaded in an alternate fashion. While considering nearby stations, some free docks are available in the corresponding area, but a potential service disruption may occur when a user arrives at an overloaded station. Still, the user could reach any of the close stations, since some of them are underutilized. To quantitatively estimate the severity of an intermittent situation, the *intermittence* measure is introduced. Intermittence counts the number of points of time at which *at least one* station (but not all of them) of the considered OMP has an occupancy level above a given threshold. The higher the intermittence, the more severe the imbalance situation.

More formal definitions of the OMP and its quality measures follow.

Definition 3.2.2 (Occupancy Monitoring Pattern)**.** Let $\mathcal{D}$ be an occupancy level dataset and let S be its attribute set. An Occupancy Monitoring Pattern (OMP) P in $\mathcal{D}$ is a set of k distinct stations in S, i.e., P={s_1, …, s_k}, $s_i \in S$. □

Definition 3.2.3 (Criticality measure)**.** The criticality of an OMP P in dataset $\mathcal{D}$ indicates the number of records R_i in $\mathcal{D}$ for which all the stations in P take value *Overloaded*. It is defined as the number of R_i in $\mathcal{D}$ such that $\forall\ (s_j, o_i^j) \in R_i$ the following conditions hold: (i) $s_j \in P$; (ii) $o_i^j = Overloaded$. □

The criticality values of similar OMPs are correlated with each other. Specifically, if an OMP P is a subset of another OMP P' (i.e., $P \subset P'$) then the criticality of P is above or equal to those of P'. Such a notable property, called *anti-monotonicity*, will be exploited to efficiently mine OMPs (see Section 3.2.2.2).

Definition 3.2.4 (Intermittence measure)**.** The intermittence of an OMP P in dataset $\mathcal{D}$ indicates the number of records R_i in $\mathcal{D}$ for which at least one station, but not all of them at the same time, takes value *Overloaded*. It is defined as the number of R_i in $\mathcal{D}$ for which the following conditions hold: (i) $\exists(s_j, o_i^j) \in R_i$ such that $s_j \in P$ and $o_i^j = Overloaded$; (ii) $\exists(s_q, o_i^q) \in R_i$ such that $s_q \in P$ and $o_i^q = Normal$. □

Criticality and intermittence values can be normalized by the number of records in $\mathcal{D}$. Their normalized values are usually denoted as *relative* criticality/intermittence values.

Example. P={s_2, s_3} is an OMP consisting of two stations (i.e., s_2 and s_3). In Table 3.2, to compute the criticality and intermittence values of P in dataset $\mathcal{D}$, the occupancy levels of stations s_2 and s_3 were evaluated at different timestamps. Since they are overloaded at the same time only in one timestamp (see record with identified RID$_1$

associated with timestamp t_1), the relative criticality value of P is $\frac{1}{7}$ (14.28%). In four timestamps (i.e., t_2, t_3, t_5, t_6 corresponding to records with RIDs equal to RID$_2$, RID$_3$, RID$_5$, RID$_6$) one station is overloaded whereas the other is normal. Therefore, the relative intermittence value of P is $\frac{4}{7}$ (57.14%).

To analyze how the occupancy level of stations evolves over time as well as detect dock overload situations happening within limited time ranges, the criticality and intermittence measures of an OMP can be reformulated by considering only the records related to a specific time period. This allows us to discover interesting patterns at a finer granularity level. Based on the target application, the time period with a suitable time granularity can be selected for monitoring the usage of stations. Given an OMP P, its criticality and intermittence value in a time period TP_k are computed considering only the subset of records with time period equal to TP_K.

Definition 3.2.5 (Criticality and Intermittence measures in time period TP_k)**.** Let TP_k be an arbitrary time period in dataset $\mathscr{D}$. Let $\mathscr{R}(TP_k)$ be the subset of records R_i in $\mathscr{D}$ that are associated with timestamps in TP_k. The criticality of an OMP P in TP_k is defined as the number of R_i in $\mathscr{R}(TP_k)$ such that $\forall\,(s_j, o_i^j) \in R_i$ the following conditions hold: (i) $s_j \in P$; (ii) $o_i^j = Overloaded$. The intermittence of an OMP P in TP_k is defined as the number of R_i in $\mathscr{R}(TP_k)$ for which the following conditions hold: (i) $\exists(s_j, o_i^j) \in R_i$ such that $s_j \in P$ and $o_i^j = Overloaded$; (ii) $\exists(s_q, o_i^q) \in R_i$ such that $s_q \in P$ and $o_i^q = Normal$. □

OMPs can be filtered based on the spatial distance between the corresponding stations. For this purpose, a spatial constraint *maxdist* was introduced on OMPs. This constraint specifies the maximum geographical distance (denoted *maxdist*) between stations in each OMP. OMPs satisfying the spatial constraint represent sets of nearby stations showing an overload situation. The higher is *maxdist*, the larger is the area including stations with critical/intermittent levels of dock occupancy.

Definition 3.2.6 (Spatial constraint)**.** Let $maxdist$ be a positive number. An OMP P satisfies the spatial constraint if for every pair of stations $s_j, s_k \in P$, $j \neq k$, their geographical distance d(s_j, s_k) is below *maxdist*. □

Given an OMP P=$\{s_1, \ldots, s_k\}$ that satisfies the spatial constraint, every subset $P' \subset P$ satisfies it as well. In fact, if for all pairs of stations $s_j, s_k \in P$ the condition d$(s_j, s_k) <$ *maxdist* is verified, it easily follows that the condition is also verified for all pairs of stations in $P' \subset P$. Such a property, called *anti-monotonicity* property, will be particularly useful for efficiently generating all the OMPs of interest (see Section 3.2.2.2).

In the implementation of the proposed methodology, geographical distances between stations were approximated with the Euclidean measure [77] thus disregarding the road network, the presence of obstacles, bridges, or underpasses. As discussed in [62], it can be deemed as a justifiable simplification since (i) Cities generally act to maximize the permeability of movement for pedestrians and cyclists, (ii) Network distances for cycling journeys are not significantly longer than Euclidean distances, especially in the city center. Similar approximations were made in other studies focused on bike and car sharing system data analyzes as well (e.g., [28, 68]).

3.2.2.2 Designed approach for OMP mining

The problem of generating OMPs has been addressed as an itemset mining problem (see Section 2.2.1).

To enable the itemset mining process in the target context, the records contained in $\mathscr{D}$ are tailored to a transactional data format. The transactional data format is required as input for the data mining algorithm exploited in this study. Intuitively, data generated with the same timestamp t_i are collected under the same RID_i (e.g., for $i = 1$ see the first row of Table 3.2), which is later converted to the corresponding TID_i (e.g., for $i = 1$ see the first row of Table 3.3).

More precisely, for each timestamp $t_i \in \mathscr{D}$ and its corresponding $\boldsymbol{R}_i$ identified by RID_i, a TID_i is generated as follows. First, the concept of *occupancy item* (o-item, in short) was introduced; next, each record $\boldsymbol{R}_i \in \mathscr{D}$ is represented in a transactional data format as a set of o-items, that is an o-itemset.

Example. The record $\boldsymbol{R}_3$ in Table 3.2, identified by the RID_3 and associated with the timestamp t_3 is represented in Table 3.3 with a transactional format as the o-itemset $\{\langle s_1, Overloaded, TP_1\rangle, \langle s_2, Overloaded, TP_1\rangle, \langle s_3, Normal, TP_1\rangle\}$ identified by TID_3.

An *o-item* represents a dock occupancy measurement acquired within a given time period and associated with a given station. More formally, an o-item is modeled as a triple $\langle s_j, o_i^j, TP_i\rangle$, where s_j is an arbitrary station, o_i^j is the occupancy level of station s_j at any timestamp $t_i \in TP_i$, and TP_i is a time period. Note that the exact timestamp at which the measurement was acquired is not explicitly reported in the o-item, because the goal is to identify the stations that have acquired critical dock occupancy levels within each time period.

In the transactional dataset $\mathscr{T}$ each transaction is logically identified by a Transaction IDentifier (TID). Each record contained in $\mathscr{D}$ is represented as a transaction in $\mathscr{T}$ characterized by the same identification value (i.e., a record with RID equal to RID_x is mapped to a transaction with TID equal to TID_x).

Example. Table 3.3 reports the transactional representation of dataset $\mathscr{D}$ on Table 3.2. Records RID_1-RID_7 are mapped to transactions TID_1-TID_7.

An *occupancy itemset* (o-itemset, in short) is a set of o-items (of arbitrary size) such that all the contained o-items correspond to the same time period. The *frequency of an o-itemset* is the number of transactions including it.

Example. $\{\langle s_1, Overloaded, TP_1\rangle, \langle s_3, Overloaded, TP_1\rangle\}$ is an o-itemset with frequency equal to 2 in the transactional dataset in Table 3.3, because it occurs in transactions with TID equal to TID_1 and TID_2. This o-itemset indicates that stations s_1 and s_3 were temporarily overloaded in two different measurements acquired in period TP_1.

Transaction IDentifier (TID)	Transaction
TID_1	$\langle s_1, Overloaded, TP_1\rangle$, $\langle s_2, Overloaded, TP_1\rangle$, $\langle s_3, Overloaded, TP_1\rangle$
TID_2	$\langle s_1, Overloaded, TP_1\rangle$, $\langle s_2, Normal, TP_1\rangle$, $\langle s_3, Overloaded, TP_1\rangle$
TID_3	$\langle s_1, Overloaded, TP_1\rangle$, $\langle s_2, Overloaded, TP_1\rangle$, $\langle s_3, Normal, TP_1\rangle$
TID_4	$\langle s_1, Overloaded, TP_1\rangle$, $\langle s_2, Normal, TP_1\rangle$, $\langle s_3, Normal, TP_1\rangle$
TID_5	$\langle s_1, Normal, TP_2\rangle$, $\langle s_2, Overloaded, TP_2\rangle$, $\langle s_3, Normal, TP_2\rangle$
TID_6	$\langle s_1, Normal, TP_2\rangle$, $\langle s_2, Overloaded, TP_2\rangle$, $\langle s_3, Normal, TP_2\rangle$
TID_7	$\langle s_1, Normal, TP_3\rangle$, $\langle s_2, Normal, TP_3\rangle$, $\langle s_3, Normal, TP_3\rangle$

Table 3.3: Example of *Occupancy Level Dataset* in transactional format.

OMPs and their criticality and intermittence values can be derived from the mined o-itemsets. Therefore, the proposed methodology for OMP mining is based on the following two steps. First, o-itemsets are mined. Then, OMPs are generated on top of the mined o-itemsets and their criticality and intermittence values are computed. In the following the two steps are separately described.

Step 1: O-itemset mining. A set of o-itemsets is extracted from the transactional representation of the occupancy level dataset. Each of the mined o-itemsets satisfies the following conditions. (i) All the contained o-items have the same occupancy level (i.e., all *normal* or all *overloaded*). (ii) All the stations contained in the o-itemset satisfy the spatial constraint *maxdist*. Thus, for every pair of stations appearing in the o-itemset, their geographical distance is below *maxdist*.

Condition (i) allows us to extract two different types of o-itemsets: the *critical o-itemsets*, which include only the o-items with occupancy level *overloaded*, and the *normal o-itemsets*, which include only the o-items with occupancy level *normal*. These o-itemsets combine the stations having all the same occupancy level in a given time period. As discussed below, these two o-itemset types will be useful at the next step to compute the OMP intermittence value. Condition (ii) allows us to filter out the combinations of o-items related to faraway stations. This will allow us to generate only OMPs including nearby stations in Step 2.

Step 2. OMPs generation. The output of Step 1 is processed at Step 2 to generate the set of OMPs. An OMP P is generated from a pair of critical and normal o-itemsets that include (i) the same stations and (ii) the same time period. The frequency values of these two o-itemsets are used to compute the criticality and intermittence values of P.

The OMP generation process is here detailed using an example case. Let us consider a pair of critical (denoted I_C) and normal (denoted I_N) o-itemsets, having both the same stations and the same time period. Consider for instance the critical o-itemset I_C= $\{\langle s_i, Overloaded, TP_k\rangle$, $\langle s_j, Overloaded, TP_k\rangle\}$ and the normal o-itemset I_N=$\{\langle s_i, Normal, TP_k\rangle$, $\langle s_j, Normal, TP_k\rangle\}$. Let us denote as *freq_value(critical)* and *freq_value(normal)* their respective frequency in time period TP_k in the analyzed dataset. Let P be the OMP generated from these two o-itemsets. The following statements hold.

(i) Pattern P contains all the stations appearing in the critical o-itemset I_C (or equivalently in the normal o-itemset I_N), i.e., P={s_i, s_j}.

(ii) According to Definition 3.2.5, the criticality of pattern P in time period TP_k is the number of times all the stations in P are overloaded in TP_K. It follows that criticality of P in period TP_k is equal to the number of transactions in TP_k including the o-itemset I_C. Thus,

$$criticality = freq\_value(critical) \tag{3.1}$$

(iii) According to Definition 3.2.5, the intermittence of pattern P in a time period TP_k is the number of times at least one station in P (but not all stations at the same time) is overloaded in TP_k. It follows that the intermittence of P in period TP_k is equal to the total frequency of all o-itemsets with the same stations as P, such that at least one station (but not all them at the same time) is overloaded in TP_k. For the sake of efficiency, the proposed approach avoids generating all these o-itemsets, but instead it proceeds as follows. Let us denote as *card_value* the total number of transactions in period TP_k in the analyzed dataset. It easily follows that *card_value* is equal to the sum of the following three terms: the frequency of the critical o-itemset I_C ($freq\_value(critical)$), the frequency of the normal o-itemset I_N ($freq\_value(normal)$) and the total frequency of all o-itemsets with the same stations as P, such as at least one station (but not all them at the same time) is overloaded at time TP_k. Therefore, the intermittence of P in period TP_k is computed as

$$intermittence = card\_value - (freq\_value(critical) + freq\_value(normal)) \tag{3.2}$$

Example. P={s_2, s_3} is an OMP with criticality equal to 1 and intermittence equal to 2 in time period TP_1. These measures are computed based on the frequencies of the critical o-itemset $\{\langle s_2, Overloaded, TP_1\rangle, \langle s_3, Overloaded, TP_1\rangle\}$ and of the normal o-itemset $\{\langle s_2, Normal, TP_1\rangle, \langle s_3, Normal, TP_1\rangle\}$. The critical o-itemset has frequency equal to 1 being contained in the transaction with TID equal to TID$_1$. Thus, the criticality of P is equal to freq_value(critical) =1. The normal o-itemset has frequency equal to 1 since it is included in the transaction with TID equal to TID$_4$ (i.e., freq_value(normal) =1). card_value is equal to 4 because four transactions refer to time period TP_1. Based on Equation 3.2, it follows that the intermittence of P is computed as intermittence = 4 - (1+1) = 2. This intermittence value corresponds to the total frequency of the o-itemsets $\{\langle s_2, Normal, TP_1\rangle, \langle s_3, Overloaded, TP_1\rangle\}$ and $\{\langle s_2, Overloaded, TP_1\rangle,$ $\langle s_3, Normal, TP_1\rangle\}$, respectively contained in the transactions with TIDs equal to TID$_2$ and TID$_3$.

Section 3.2.2.3 describes the algorithm used in the *BELL* framework to mine the OMPs including nearby stations according to the spatial constraint *maxdist* as well their criticality and intermittence values.

3.2.2.3 The OMP-Miner algorithm

Algorithms 1 and 2 report the pseudo-code of the algorithm designed to extract OMPs. It consists of the following three main phases:

- Phase 1: Creation of a compact in-memory representation of the occupancy level transactional dataset (Algorithm 1, line 1).

- Phase 2: Mining of all the critical and normal o-itemsets including nearby stations according to the spatial constraint *maxdist* (Algorithm 1, line 2).

- Phase 3: Generation of the OMPs on top of the mined o-itemsets and computation of their criticality and intermittence levels (Algorithm 1, lines 3-7).

As stated in 3.2.2.2, the issue of identifying clusters of docking stations in an overloaded or intermittent condition can be naturally revolved to the problem of discover frequent itemsets, i.e., recurrent combinations of items possessing certain characteristics that are valuable for the user. Thus, in this study one of the most prominent algorithm proposed in the literature, FP-growth [39], has been implemented as an established and efficient foundation for our methodology to extract the o-itemsets. The main advantage of the FP-growth based approach is the selective generation of the candidate o-itemsets, which prevents the time- and memory-consuming candidate generation phase adopted by the Apriori strategy [3]. Moreover, lately FP-growth has largely increased its scalability due to the fact that an equivalent distributed version of the algorithm has become available on big data frameworks such as Spark [6] making it possible and relatively easy to adapt *BELL* to perform on even larger datasets efficiently.

Phase 1 entails storing the measurements reported in the transactional representation $\mathscr{T}$ of the original dataset into a compact tree-based structure. To accomplish this task, the prefix-tree data structure adopted by FP-Growth, namely the FP-Tree, is exploited to store the transactional dataset $\mathscr{T}$.

In the context of this study, each node of the tree contains an o-item together with the frequency of the o-item in the path. A transaction in $\mathscr{T}$ is stored in the FP-tree as a path connecting o-items corresponding to the same time period. Figure 3.2 reports the FP-tree that represents the transactional dataset $\mathscr{T}$ in Table 3.3. For the sake of compactness and readability *Overloaded* and *Normal* conditions in o-items are denoted as O and N, respectively. The key advantage of scanning the FP-tree index instead of the original dataset in the o-itemset mining process is that in the FP-tree multiple dataset transactions containing the same o-items are stored in the same path. For example, the FP-tree path $[\langle s_1, O, TP_1\rangle, \langle s_2, O, TP_1\rangle, \langle s_3, O, TP_1\rangle]$ represent transaction with TID equal to TID$_1$, but subpath $[\langle s_1, O, TP_1\rangle, \langle s_2, O, TP_1\rangle]$ represents a common part in transactions with TIDs equal to TID$_1$ and TID$_2$.

The FP-tree is built as follows (Algorithm 1, line 1). For each o-item in $\mathscr{T}$ its frequency is computed and stored in a data structure called Header Table. O-items are ordered in the Header Table by decreasing value of their frequency, and they are linked to the FP-Tree nodes including them. For the sake of compactness in Figure 3.2 only a portion of the whole Header Table is shown. Transactions in $\mathscr{T}$ are then considered one at time. First the o-items in the transaction are ordered according to the o-item order in the Header Table; then the ordered transaction is inserted in the FP-tree using the same approach described in [39].

Phase 2 entails generating all the critical and normal o-itemsets including only nearby stations by recursively visiting the FP-Tree (Algorithm 1, line 2). The O-ITEMSETMining algorithm relies on the recursive FP-tree visit adopted by FP-Growth. However, in the proposed approach the anti-monotonicity property of the spatial constraint (see Section 3.2.2.1) is exploited to reduce the number of generated combinations. The O-ITEMSETMining algorithm considers one at a time the o-items in the Header Table and generates the o-itemsets including the targeted o-item and a combination of the other o-items in the dataset. For instance, consider the FP-Tree in Figure 3.2. First the o-item $i^* = \langle s_3, O, TP_1 \rangle$ is selected to generate the o-itemsets including it (Algorithm 2, line 3). At this first step the o-itemset I=$\{\langle s_3, O, TP_1 \rangle\}$ with frequency equal to 2 is extracted.

To generate further extensions of the current o-itemset I, the dataset transactions including all o-items in I should be analyzed (Algorithm 2, line 4). These transactions are represented in the FP-tree paths containing all o-items in I. For instance, when I= $\{\langle s_3, O, TP_1 \rangle\}$, two FP-tree paths, highlighted in Figure 3.3(a), are selected. These paths represent transactions with TIDs TID$_1$ and TID$_2$. To avoid the generation of useless new o-itemsets, nodes from each selected path are filtered as follows (Algorithm 2, line 5). (i) To guarantee the compliance with the spatial constraint, nodes containing o-items that do not satisfy the maximal distance constraint with o-items in I are discarded. (ii) To guarantee that the o-itemsets are homogeneous in the occupancy level (i.e, all o-items have level Normal or Overloaded) nodes with an occupancy level different from the o-items in I are pruned.

In the example in Figure 3.3(b) two nodes are pruned from the selected paths. (i) Let's suppose that stations s_3 and s_1 do not verify the spatial constraint, i.e., $d(s_3, s_1)$>$maxdist$ while $d(s_3, s_2)$<$maxdist$. Since the mined o-itemsets cannot contains both stations s_3 and s_1, the node with o-item $\langle s_1, O, TP_1 \rangle$ is pruned from the selected paths. (ii) Node with o-item $\langle s_2, N, TP_1 \rangle$ is pruned because its occupancy level is different from the occupancy level in I= $\{\langle s_3, O, TP_1 \rangle\}$.

When the pruning phase is concluded, a conditional FP-tree, including only the selected paths is created (using the same approach used in Algorithm 1, line 1) and the O-ITEMSETMining algorithm is recursively invoked on it (Algorithm 2, line 8). This new invocation iterates over the conditional FP-Tree with the aim of extending the o-itemset I with the o-items in the conditional FP-tree. A stop condition for the recursive invocation is reached when the conditional FP-tree is empty. In this case the algorithm backtracks to the previous invocation of the O-ITEMSETMining function;

Algorithm 1 OMP-Miner($\mathscr{T}$, *maxdist*, TP)

```
Require: $\mathscr{T}$: occupancy level dataset in transactional format
Require: maxdist: maximum distance between two stations in the same OMP
Require: $TP$: set of time periods $TP_1, \ldots, TP_q$
Ensure: $\mathscr{P}$: the set of OMPs for each time period in $TP$
1: $FPTree \leftarrow$ FP-tree($\mathscr{T}$) { Create the initial FP-tree from $\mathscr{T}$ }
2: $\mathscr{F} \leftarrow$ O-ITEMSETMining($FPTree$, maxdist, $\emptyset$) { Recursive projection-based o-itemset mining function} { Generate OMPs on top of the mined o-itemsets in $\mathscr{F}$}
3: $\mathscr{F}_{normal}$: normal o-itemsets $I_N$ in $\mathscr{F}$
4: $\mathscr{F}_{critical}$: critical o-itemsets $I_C$ in $\mathscr{F}$
5: $H$: Hash map with keys $\langle I_N, TP_k \rangle$ storing the criticality values of each normal o-itemset $I_N \in \mathscr{F}_{normal}$ for each period $TP_k$
6: card_value[]: vector storing in the $k$-th element the number of transactions in $\mathscr{T}$ associated with period $TP_k$
7: $\mathscr{P}$= ComputeOMPintermittence($\mathscr{F}_{critical}$,$H$,card_value)
8: return $\mathscr{P}$
```

then it restarts the mining process from there by considering a different o-item in the local FP-tree.

In the running example, the conditional FP-Tree associated with the second algorithm invocation contains only o-item $\langle s_2, Overloaded, TP_1 \rangle$. Thus, the o-itemset $\{\langle s_3, O, TP_1 \rangle, \langle s_2, O, TP_1 \rangle\}$ with frequency equal to 1 is generated. At this point, a stop condition for the recursive invocation has been reached since the conditional FP-tree with respect to the o-itemset $\{\langle s_3, O, TP_1 \rangle, \langle s_2, O, TP_1 \rangle\}$ is empty. The algorithm backtracks to FP-tree represented in Figure 3.2 to target the extraction of the o-itemsets including the o-item which precedes item $\langle s_3, O, TP_1 \rangle$ in the Header Table.

Phase 3 aims at generating OMPs by properly combining the critical and normal o-itemsets mined at Phase 2 and stored in sets $\mathscr{F}_{critical}$ and $\mathscr{F}_{normal}$, respectively (Algorithm 1, lines 3 and 4).

For each critical o-itemset $I_C \in \mathscr{F}_{critical}$, an OMP P is generated with criticality and intermittence value computed according to Equation 3.1 and Equation 3.2, respectively. For instance, the critical o-itemset $\{\langle s_3, O, TP_1 \rangle, \langle s_2, O, TP_1 \rangle\}$ with frequency equal to 1 and the normal o-itemset $\{\langle s_3, N, TP_1 \rangle, \langle s_2, N, TP_1 \rangle\}$ with frequency equal to 1 are mined during Phase 2 from the running example dataset in Table 3.3. Those two o-itemsets are related to time period TP_1, which is associated with 4 transactions in the running example dataset. Given those two o-itemsets and the number of transactions associated with TP_1, the OMP-Miner algorithm extracts the OMP $\{s_3, s_2\}$ associated with TP_1 with criticality equal to 1 and intermittence equal to 2.

To efficiently compute the pattern intermittence value, the normal o-itemsets and their corresponding frequency values are stored in a hash map data structure. Given a critical o-itemset I_C, the frequency of the corresponding normal o-itemset I_N including the same stations is returned by the hash map given the key $\langle I_N, TP_k \rangle$ (Algorithm 1, line 7).

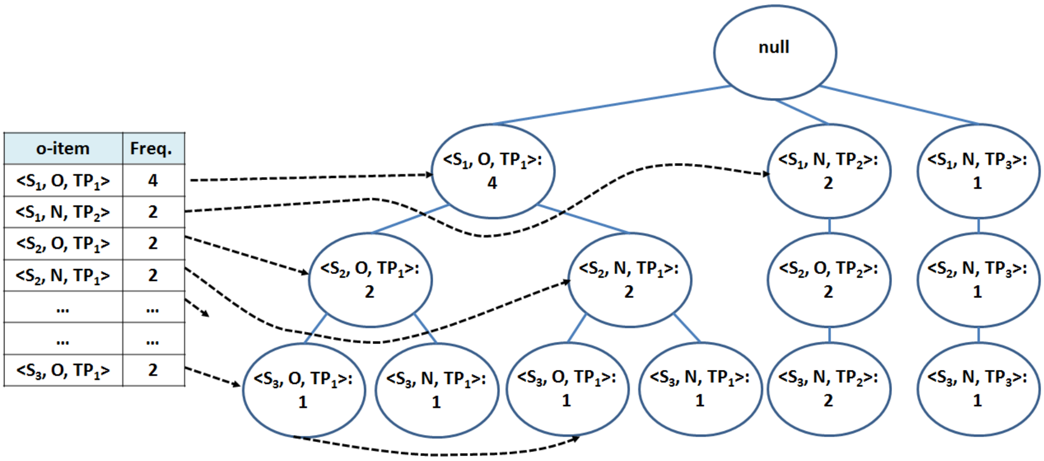
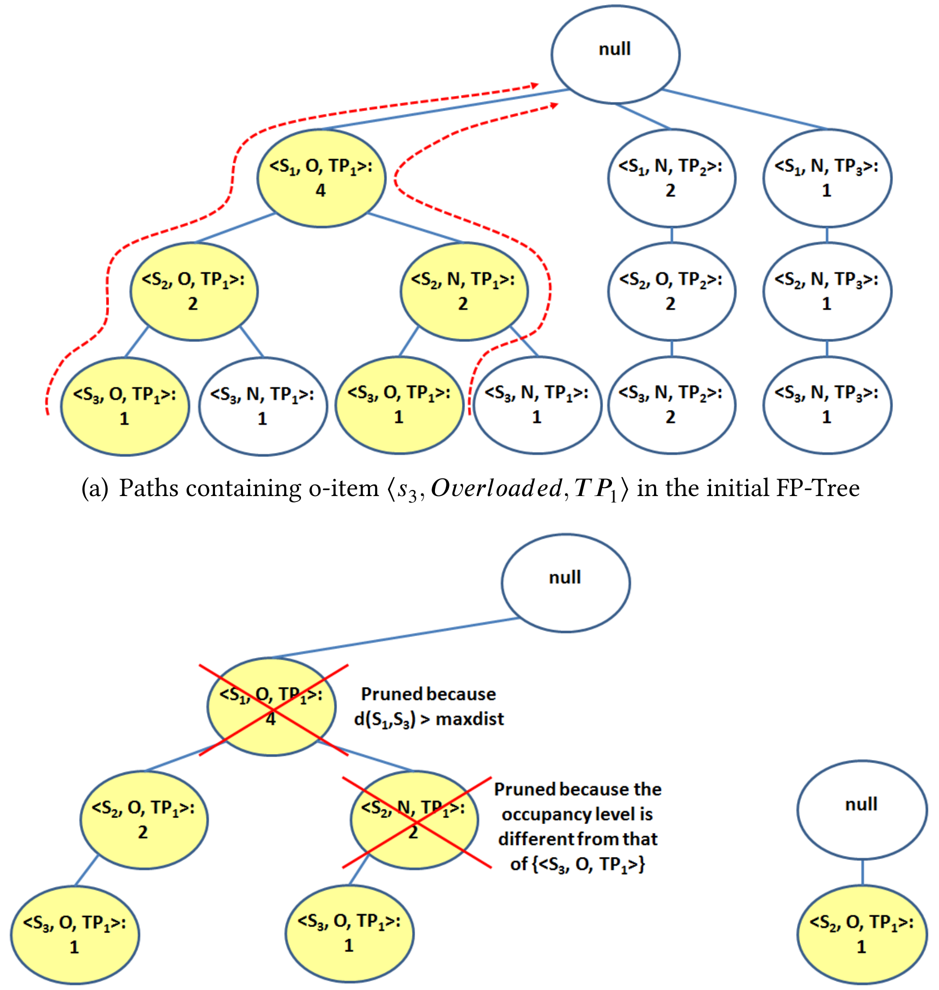


Figure 3.2: The FP-Tree representing the example transactional Occupancy Level Dataset (Table 3.3).

Algorithm 2 O-ITEMSETMining($FPTree$, $maxdist$, I^*)

```
Require: $FPTree$, an FP-tree
Require: $maxdist$: maximum distance between two stations in the same o-itemset
Require: $I^*$, the set of o-items with respect to which FPTree has been generated
Ensure: $\mathscr{F}$, the set of o-itemsets extending $I^*$
$\mathscr{F} \leftarrow \emptyset$
for all o-item $i^*=\langle s_j, o_i^j, TP_i\rangle$ in the header table of $FPTree$ do
    $I \leftarrow I^* \cup \{i^*\}$ {Generate a new o-itemset $I$ by joining o-itemset $I^*$ and o-item $i^*$ }
    $\mathscr{F} \leftarrow \mathscr{F} \cup \{I\}$
    STATE $CondPaths_I \leftarrow$ selectConditionalPaths($FPTree$, $I$) { Select $I$'s conditional paths}
    $PrunedCondPaths_I \leftarrow$ applyConstraints($CondPaths_I$, $I$) {Prune o-items $k^*=\langle s_k, o_i^k, TP_i\rangle$ such that $\exists\langle s_x, o_i^x, TP_i\rangle \in I \mid$ distance($s_k$, $s_x$)>$maxdist$ or $o_i^k \neq o_i^x$}
    $FPTree_I \leftarrow$ createFP-tree($FPTree$, $I$) {Build $I$'s conditional FP-tree}
    if $FPTree_I \neq \emptyset$ then
        $\mathscr{F} \leftarrow \mathscr{F} \cup$ O-ITEMSETMining($FPTree_I$, $maxdist$, $I$) { Recursive mining}
    end if
end for
return $\mathscr{F}$
```

Complexity analysis Phases 1 and 2 of OMP-Miner are based on an FP-growth-like mining algorithm. Similar to FP-growth [39], its complexity is linear with respect to the number of mined o-itemsets, which is combinatorial with the number of items, i.e., $2^{\#items}$ in the worst case. However, enforcing the spatial constraint allows us to significantly reduce the number of generated itemsets (see Algorithm 2). Finally, the extracted o-itemsets are combined to mine OMPs and compute their quality measures. Also, this final phase is linear with respect to the number of mined o-itemsets.



(a) Paths containing o-item $\langle s_3, Overloaded, TP_1\rangle$ in the initial FP-Tree

(b) Node pruning based on maximal distance constraint and occupancy level with respect to $\{\langle s_3, O, TP_1\rangle\}$

(c) I's conditional FP-tree with respect to $\{\langle s_3, O, TP_1\rangle\}$

Figure 3.3: O-itemset mining example.

3.2.3 Knowledge exploration

The OMPs extracted with the OMP-Miner algorithm can be explored by system managers to gain insight into system usage. This exploratory analysis allows domain experts to focus their attention on a limited number of stations on given areas and in specific time periods. Based on the mined knowledge, domain experts may recommend targeted maintenance actions with the aims of reducing disruption to end users. To effectively explore the mining result a list of recommendations is given below.

Exploration of intermittent situations. To detect significant intermittent situations, OMPs should be ranked by decreasing intermittence value. To ease the exploration process, the OMPs with very low intermittence value can be discarded. OMPs with maximal intermittence value indicate groups of stations that are frequently fully occupied in an alternate fashion. These OMPs represent station occupancy level conditions that could result in a limited disservice to the end user. If the stations in the OMP are located in the same area, then an alternative arrival station can be recommended to users who reach an occupied station. The severity of the possible disservices for end users can vary based on the criticality value of the OMP. When the pattern criticality level increases, the stations indicated by the OMP are more frequently fully occupied at the same time; thus, end users are unlikely to find a free dock at nearby stations.

To avoid disservices, system managers can suggest an alternative nearby station with free docks for parking; in case of OMPs with high intermittence but low criticality values, bicycles may be re-positioned in nearby stations because they are rarely fully occupied at the same time.

Exploration of critical situations. In order to detect significant critical situations which could lead to serious disservice for end users, OMPs should be ranked by decreasing criticality value. To ease the exploration process, the OMPs with very low criticality value can be discarded. OMPs with maximal criticality value indicate groups of nearby stations that are frequently fully occupied at the same time. Thus, end users are unlikely to find free docks for their bikes in this area.

Since nearby stations are all fully occupied, maintenance actions such as bicycle repositioning should be carried out considering stations that are further away or located in other areas of the city. Therefore to address these issues, maintenance actions could be much more expensive or even inapplicable. Alternative actions could be considered such as planning station resizing or system enlargement.

Exploration of the spatio-temporal distribution of intermittent and critical situations. To support management of the bicycle sharing system, the mined OMPs can be visualized on a map of the city area. Since each station in the OMP is characterized by a geographical position, OMPs can be represented as restricted city areas including the corresponding stations. This representation is intuitive and effective for highlighting the areas which could lead to disservices for end users. OMP representations can be differentiated based on the type of imbalance in station occupancy (i.e., critical, intermittent) and the degree of severity of the discovered pattern. Domain experts can also analyze intermittent and critical situations for different values of time periods to identify the time frames associated with more serious disruptions. For example, they can consider 1-hour time slot as time period to analyze the number and significance of intermittent and critical situations for each hour in a day. Alternatively, they can adopt a courser time granularity, as a larger time slot size (e.g., morning, afternoon, evening, night), to gather a more high-level view of the dock overload conditions in the bicycle sharing system.

Domain experts are recommended to adhere to the following guideline in order to properly set up the OMP-Miner algorithm. The spatial constraints *maxdist* should be set according to the geographical distribution of the stations in the city area. For example, stations located at walking distance can be considered as *near* while stations located in different districts can be classified as *distant*. To ensure that the extracted OMPs include only close stations, the user should set *maxdist* as the largest distance between a pair of *nearby* stations.

Some examples OMPs representing significant intermittent and critical situations in real data collections, and the analysis of their spatio-temporal distribution, are reported in Section 3.3.

3.3 Experimental results

The efficiency and usability of the *BELL* system on real data acquired from bicycle sharing systems were validated in two important cities: Barcelona, the capital city of the autonomous community of Catalonia and Spain's second most populated city and New York, the most populated city in the United States of America.

The experimental evaluation addresses the following aspects. Some examples of interesting OMPs representing significant intermittent and critical situations, extracted from the analyzed data collections, are presented in Section 3.3.2. Section 3.3.3 evaluates the impact of the system configuration parameters on the number of mined OMPs and on their corresponding intermittence and criticality values, while Section 3.3.4 reports performance evaluation in terms of execution time for the OMP-Miner algorithm. The main characteristics of the analyzed datasets are summarized in Section 3.3.1.

The OMP-Miner algorithm was implemented by using the C language. The experiments were performed on a 2.67 GHz six-core Intel(R) Xeon(R) X5650 machine with 32 Gbyte of main memory running Ubuntu 18.04 server with the 3.5.0-23-generic kernel.

3.3.1 Reference use case datasets

This section briefly presents the main characteristics of the two bike sharing systems considered as reference use case in this study and describes data that has been considered on the system usage.

The *Bicing* system in Barcelona. *Bicing* is the bicycle sharing system in Barcelona which consists of 377 stations distributed all over the city area. Stations have a fixed number of parkings, which vary from 15 to 39. A description of the service is given in [42]. Data from the *Bicing* website[1] can be crawled through the Google Maps APIs. To perform the analyses, the collection of measurements described in [42] have been

[1] www.bicing.com/localizaciones/localizaciones.php

taken into account. The acquired data (from a single operator) include 30 million records from the *Bicing* stations over a period of approximately a semester of service (i.e., between May 15th and November 30th, 2008). Occupancy values were acquired every 5 minutes.

The *Citi Bike* system in New York. *Citi Bike* is the bicycle sharing system in New York which features thousands of bikes at 528 stations across New York and Jersey City. Bicycles are available 24/7, 365 days a year. More information about the system is available at https://member.citibikenyc.com/. The *Citi Bike* system provides open data in the JSON format through the *Citi Bike* station feed service[2]. To perform the analyses, an ad hoc Web crawler was developed which downloaded and parsed the JSON data from the *Citi Bike* system feed to retrieve the historical occupancy data. Occupancy values were acquired every 5 minutes over a time period of approximately 13 months (i.e., between October 23th 2014 and November 17th, 2015).

Characteristics of the collected data on the system usage. In both bicycle sharing systems, each station is characterized by the information on its *name* and *geographic coordinates* (latitude and longitude). *Historical data* on station occupancy can be collected by submitting periodical requests to the stations in the system and storing the corresponding responses. Specifically, for each station the information on the *number of free* and *occupied slots in different time instants* was acquired within a given time window.

3.3.2 OMP characterization

Following, some OMPs are discussed as representative examples of the insights mined through the framework. Specifically, some top ranked OMPs with maximal intermittence and criticality values are discussed as reference cases. These OMPs represent dock overload conditions that could yield to disservices for end users in the usage of the bicycle sharing system.

OMPs were extracted from the *Bicing* and *Citi Bike* datasets using a standard system configuration with *maxdist* = 0.5 km, *full-th*=3, and *time period* equal to *time slot size* of 1 hour. This configuration pinpoints a time-space granularity suitable to provide useful information to end users and system managers. For example, let's set *maxdist* = 0.5 km because bikers are (usually) more willing to move to physically closer stations if the expected destination is fully occupied. Let's set the time period equal to *time slot size*=1h to determine more precisely sets of nearby stations that could lead to service disruption. Parameter *full-th* has been set to 3 to represent situations when the station is (almost) full. The impact of the system parameters on the characteristics of the extracted OMPs is discussed in Section 3.3.3.

[2] http://www.citibikenyc.com/system-data

Example OMPs with maximal intermittence. The OMP-Miner algorithm generates as output a set of OMPs with various intermittence values. The intermittence measure of an OMP is computed to measure the presence of a dock overload condition from the occupancy levels of the corresponding stations (see Algorithm 1, line 7). The higher the intermittence value, the more severe the imbalance condition. Hence, OMPs with highest intermittence values should be considered first in the result exploration.

Tables 3.4 and 3.5 report some examples of top ranked OMPs with maximal intermittence value extracted from the *Bicing* and *Citi Bike* datasets, respectively. In both tables OMPs are sorted by decreasing intermittence value. The example OMPs from the *Bicing* dataset (Tables 3.4) are characterized as follows.

OMPs with identifiers (IDs) 5-7 represent dock overload conditions that could yield a *limited disservice* for end users. Each of these OMPs represents a group of stations that the end user is likely to find fully occupied in alternate fashion (in about 62-63% of the recorded timestamps according to the intermittence value). However, the low criticality values of these OMPs point out that the stations in each OMP are rarely fully occupied at the same time (in about 0.13%-1.56% of the cases). It follows that, in case the user is unable to park in one station she/he can move to another nearby station where free parking docks will be available with a high probability. For example, OMP with ID 5 indicates that the usage levels of stations *Carrer de Bonavista* and *Pl. del Poble Romaní* are critical in an alternate fashion from 7am to 8am in 63% of the cases, but they are fully occupied at the same time only in 1.56% of the cases.

On the other hand, OMPs with IDs 1-2 represent dock overload conditions that could result in *a more serious disservice* for end users. Each of these OMPs models a group of stations having both intermittence and criticality values higher than OMPs with IDs 5-7. For each OMP, at least one station has a high probability of being occupied (intermittence value higher than 71%), and all stations have a not negligible probability of being fully occupied at the same time (criticality about 8%). Therefore, in case the user cannot park in one station, she/he might not find a free dock at a nearby station approximately 8% of the time. As an example, OMP with ID 1 shows that, from 4am to 5am , stations *Vilamara davant*, *Mallorca* and *Calabria* have a critical usage level in an alternate fashion in 73.84% of the recorded timestamps, and they are simultaneously fully occupied in 8.29% of the cases.

OMPs with IDs 3-4 represent an intermediate condition between the two above. These OMPs have intermittence and criticality values higher than OMPs with IDs 5-7 (intermittence 70%-71% instead of 63% and criticality 1.86-4% instead of 0.13%-1.56%), but lower than OMPs with IDs 1-2 (intermittence 70%-71% instead of 73% and criticality 1.86%-4% instead of 8%).

Based on the mined knowledge, domain experts may recommend an alternative nearby station for parking and/or targeted maintenance actions. For instance they may decide to relocate bicycles at the beginning of the time slot, moving them from stations with critical levels to non-critical stations.

Compared to the OMPs extracted from the *Bicing* dataset, the top ranked OMPs mined from the *Citi Bike* dataset have very high intermittence values (between 90% and 100%) and criticality equal to 0% (Table 3.5). For example, OMP with ID 2 consists of four nearby stations (*{W 33 St & 8 Ave, W 29 St & 9 Ave, W 31 St & 8 Ave, Penn Station Valet}*) with 100% intermittence and 0% criticality from 8pm to 9pm. These stations are close to Madison Square Garden Stadium and Pennsylvania Station, which are big subway and train hubs. These OMPs indicate conditions which could lead to a *limited disservice* for the end users. On the one hand, since the OMP intermittence value is very high, at least one of the stations in the OMP is likely to be fully occupied. While on the other hand, since the criticality value is 0%, at least one station has a free dock in all the recorded timestamps. Consequently, the user will probably find a free dock among nearby stations.

Example OMPs with maximal criticality. The OMP-Miner algorithm computes the criticality of each of the mined OMPs (see Algorithm 1, line 4). The criticality measure indicates the unavailability of most of the docks in a set of stations. The higher the criticality, the more critical the situation of imbalance that need to be faced.

Tables 3.6 and 3.7 report the top ranked OMPs with maximal criticality value mined from the *Bicing* and the *Citi Bike* dataset, respectively. OMPs in Tables 3.6 and 3.7 represent potentially *severe disservices* for the end users of the system, because they identify groups of nearby stations whose levels of usage are frequently *all* critical at the same time.

For example, for the *Bicing* in Table 3.6, OMP with ID 1 indicates that from 10am to 11am stations *Marquas de l'Argentera* and *Avinguda del Marques Argentera* (approximated distance 300m) both have critical usage levels in approximately 38% of the recorded timestamps. Thus, one third of the time the parking is unavailable in this time slot in the mentioned areas. If the problem persists, users working or living in the neighborhood are strongly discouraged from using the service. Since nearby stations are all fully occupied, maintenance actions such as bicycle repositioning should be carried out considering stations that are further away or located in other areas of the city. Therefore, in order to address these issues, maintenance actions could be much more expensive or even not feasible.

Results in Table 3.7 report even more critical situations for some groups of stations in the *Citi Bike* dataset. For instance, OMP with ID 1 representing the nearby stations *E 85 St & 3 Ave* and *E 84 St & 1 Ave* has a criticality equal to 51%. Hence, in half of the cases both stations are fully occupied.

Hourly distribution of intermittent/critical OMPs. The OMP-Miner algorithm allows us to extract OMPs and store their criticality/intermittence values in different time slots (see Algorithm 1, Line 5). Analyzing the quality measures in different time slots allows domain experts to detect time-constrained imbalance situations (e.g., situations arising in specific hourly time slots).

Figures 3.4 and 3.5 show the hourly distribution of the number of OMPs and their

ID	OMP	Time slot	Crit. %	Interm. %
1	{Vilamara- davant, Mallorca, Calabria}	[4am,5am]	8.29	73.84
2	{Vilamara- davant, Mallorca, Calabria}	[2am,3am]	8.58	73.53
3	{Sant Pere Mas Alt, Pl. Carles Sunyer, Pl. Catalunya, Pl. Urquinaona}	[10am,11am]	1.86	71.28
4	{Pl. Catalunya A, Pl. Catalunya B, Pl. Catalunya C, Pl. Urquinaona}	[11am,12am]	4.31	70.72
5	{Carrer de Bonavista, Pl. del Poble Romaní}	[7am,8am]	1.56	63.05
6	{Carrer del Cana, Pl. del Poble Romaní}	[5am,6am]	0.13	62.69
7	{Pl. del Poble Romaní, Montmany}	[6am,7am]	0.13	62.41

Table 3.4: *Bicing* (Barcelona). Groups of stations with maximal intermittence in different hourly time slots.

ID	OMP	Time slot	Crit. %	Interm. %
1	{W 42 St & 8 Ave, PABT Valet} PABT Valet}	[7pm,8pm]	0	100
2	{W 33 St & 8 Ave, W 29 St & 9 Ave, W 31 St & 8 Ave, Penn Station Valet}	[8pm,9pm]	0	100
3	{W 41 St & 8 Ave, W 45 St & 9 Ave, W 42 St & 8 Ave, PABT Valet}	[7pm,8pm]	0	100
4	{W 42 St & 8 Ave, PABT Valet}	[6pm,7pm]	0	93.7
5	{E 22 St & Broadway, E 24 St & Park Ave}	[11am,12am]	0	90

Table 3.5: *Citi Bike* (New York). Groups of stations with maximal intermittence in different hourly time slots.

corresponding levels of intermittence and criticality. The two figures report, for each hourly time slot, the *total number* of mined OMPs characterized by different ranges of intermittence and criticality values. In order to identify OMPs that could lead to a disservice for end users, OMPs with an intermittence/criticality value greater than or equal to 20% have been taken into consideration.

In the *Bicing* dataset (Figure 3.4) a significant number of OMPs with intermittence/-criticality values greater than or equal to 20% occurs in all hourly time slots. However, OMPs with higher values of intermittence/criticality mainly occur between 1am-2am, 7am-1pm and 4pm-11pm.

OMPs mined from the *City Bike* dataset (Figure 3.5) show a similar hourly distribution to OMPs from the *Bicing* dataset. However, a lower number of OMPs with high intermittence/criticality values comes from the *City Bike* dataset, probably because the

ID	OMP	Time slot	Crit. %	Interm. %
1	{Marquas de l'Argentera, Avinguda del Marques Argentera}	[10am,11am]	37.96	19.23
2	{Gran Via, Rocafort}	[11am,12am]	35.94	19.91
3	{Gran Via, Rocafort}	[10am,11am]	34.48	19.84
4	{Marquas de l'Argentera Avinguda del Marques Argentera}	[11am,12am]	33.52	21.15
5	{Paralà lel, Pl. Jean Genet}	[1am,2am]	32.64	25.42
6	{Paralà lel, Sant Oleguer, Pl. Jean Genet}	[1am,2am]	23.41	41.91
7	{Marquas de l'Argentera, Avinguda del Marques Argentera, Pl. Comercial}	[10pm,11pm]	22.99	37.16
8	{Marquas de l'Argentera Avinguda del Marques Argentera, Pl. Comercial}	[12pm,1am]	22.48	32.55

Table 3.6: *Bicing* (Barcelona). Groups of stations with maximal criticality in different hourly time slots.

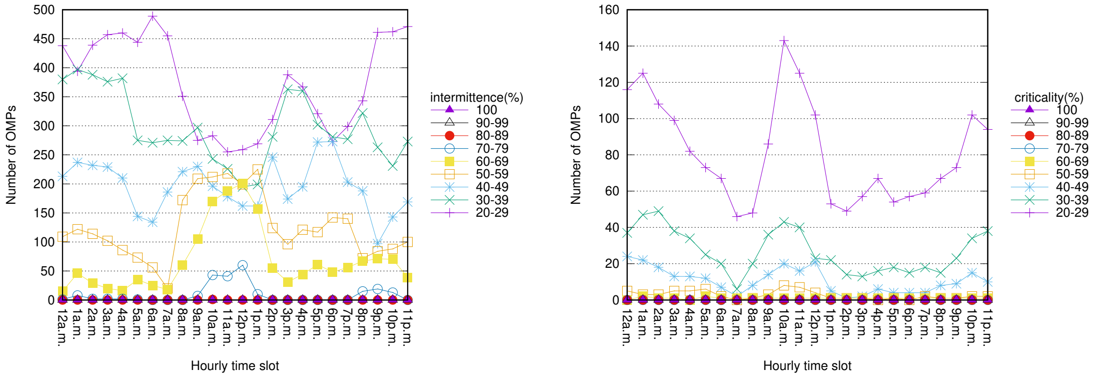
ID	OMP	Time slot	Crit. %	Interm. %
1	{E 85 St & 3 Ave, E 84 St & 1 Ave}	[8pm,9pm]	51.15	29.01
2	{E 53 St & Madison Ave, E 48 St & 5 Ave}	[9am,10am]	49.76	20.53
3	{E 84 St & 1 Ave, E 82 st & 2 Ave}	[9pm,10pm]	49.26	27.53
4	{E 85 St & 3 Ave, E 84 St & 1 Ave}	[7pm,8pm]	45.01	31.13
5	{W 51 St & 6 Ave, E 48 St & 5 Ave}	[9am,10am]	44.93	16.91

Table 3.7: *Citi Bike* (New York). Groups of stations with maximal criticality in different hourly time slots.

stations in New York are more widespread than those in Barcelona.

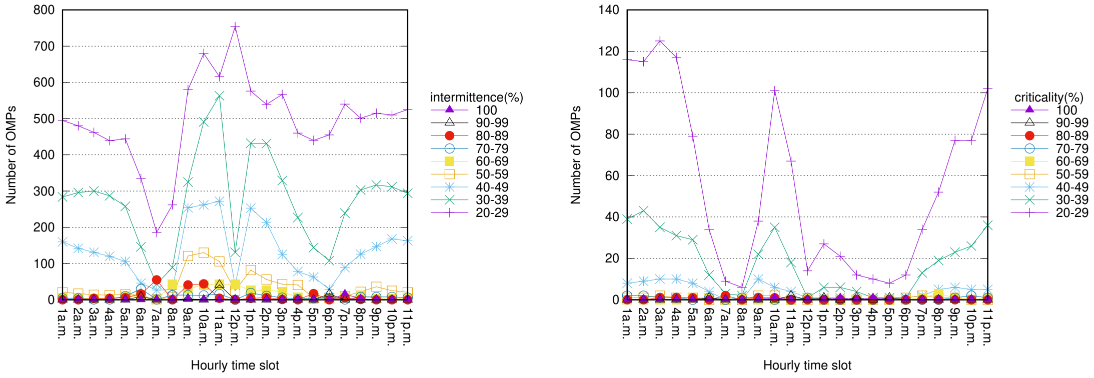
Domain experts can thus gather useful insights on the usage of the bicycle sharing system. On the one hand, they can identify daily time periods in which service disruptions may occur, and on the other hand they can also identify the set of nearby stations which are involved in these disservices.

Geographical distribution of significant intermittent and critical OMPs. Each OMP represents a group of geo-referenced stations. To support the management of the bicycle sharing system, maps can be used to highlight the city areas associated with OMPs (i.e., groups of stations) with high intermittence and criticality values. Notice that OMPs can be easily visualized on a map because they represent groups of



(a) Intermittence (b) Criticality

Figure 3.4: *Bicing* (Barcelona). Hourly distribution of the number of OMPs and their corresponding levels of intermittence/criticality. *maxdist*=0.5 km. *time slot size*=1h. *full-th*=3.



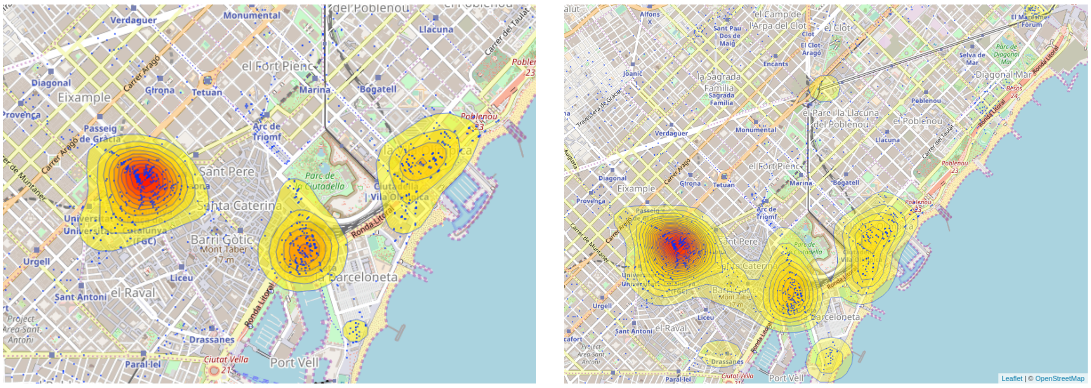
(a) Intermittence (b) Criticality

Figure 3.5: *Citi Bike* (New York). Hourly distribution of the number of OMPs and their corresponding levels of intermittence and criticality. *maxdist*=0.5 km. *time slot size*=1h. *full-th*=3.

nearby stations. The extraction and visualization of OMPs including distant stations is prevented by enforcing the spatial constraint in the OMP-Miner algorithm (see Algorithm 2, line 5).

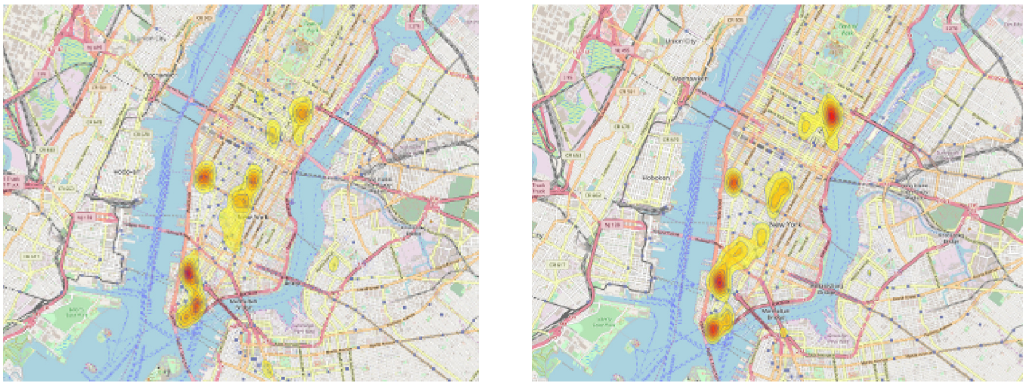
For example, Figures 3.6(a) and 3.6(b) show two heat maps[3] of the areas of Barcelona identified by the OMPs in hourly time slot [11am-12am). OMPs in this time slot represent significant intermittent and critical situations according to the results in Figure 3.4.

[3] The heat maps have been generated by using the service provided by Babicki et al. [8].



(a) Intermittence (b) Criticality

Figure 3.6: Heat maps representing intermittence and criticality values in Barcelona at the hourly time slot [11am-12am). *maxdist*=0.5 km, and *time slot size*=1h. *full-th*=3.



(a) Intermittence (b) Criticality

Figure 3.7: Heat maps representing intermittence and criticality values in New York at the hourly time slot [11am-12am). *maxdist*=0.5 km, and *time slot size*=1h. *full-th*=3.

In Figures 3.6(a) and 3.6(b) the color intensity of areas increases with the density of occurrence of OMPs and their intermittence and criticality values, respectively. The higher the color intensity, the most severe the disservice to end users.

Figure 3.6(a) shows that intermittent situations are mainly localized in the city center in four distinct areas. The area with the highest intensity is centered in *Placa Catalunya*, while the other two large areas are centered in *History Museum of Catalonia* and *La Vila Olimpica del Poblenou* and a small area is in *Pla de Miquel Tarradell*.

Instead, based on Figure 3.6(b), critical situations are more spread over the geographical areas. The larger area in Figure 3.6(b) covers all the three main areas in Figure 3.6(a). Moreover, three additional areas show up, two of them located on the top of the map (in the *Torre Glories* and *El Maresme Forum* areas) and one on the bottom

(*Drassanes* area).

Heat maps were exploited to analyze the geographical distribution of OMPs mined in hourly time slot [11am-12am) in New York (see Figures 3.7(a)-3.7(b)). Compared to Barcelona, more areas in New York are characterized by OMPs with high intermittence and criticality values. The areas with the highest intensity for intermittence situations are mainly located in the World Trade Center (on the bottom of the map). While the highest intensity for critical situations is located both in the areas of the World Trade Center and of the Museum Of Modern Art (on the top of the map).

3.3.3 Parameter analysis

The main parameters of the OMP-Miner algorithm are as follows. (i) The threshold used to discriminate station occupancy levels into *Normal* and *Overloaded*, i.e., the occupancy threshold *full-th*. (ii) The threshold used to decide whether two stations are located nearby or not, i.e., the maximum distance threshold *maxdist*. (iii) The time granularity used to analyze the evolution of imbalance situations over time, i.e., *time slot size*.

The analysis was performed to detect the impact of parameters *full-th*, *maxdist*, and *time slot size* on (i) The cardinality of the mined OMPs (i.e., the number of OMPs per time slot), (ii) The distribution of the intermittence values of the mined OMPs, and (iii) The distribution of the criticality values of the mined OMPs. Moreover, the impact of the day category on the hourly distribution of the intermittence and criticality values for the mined OMPs was analyzed.

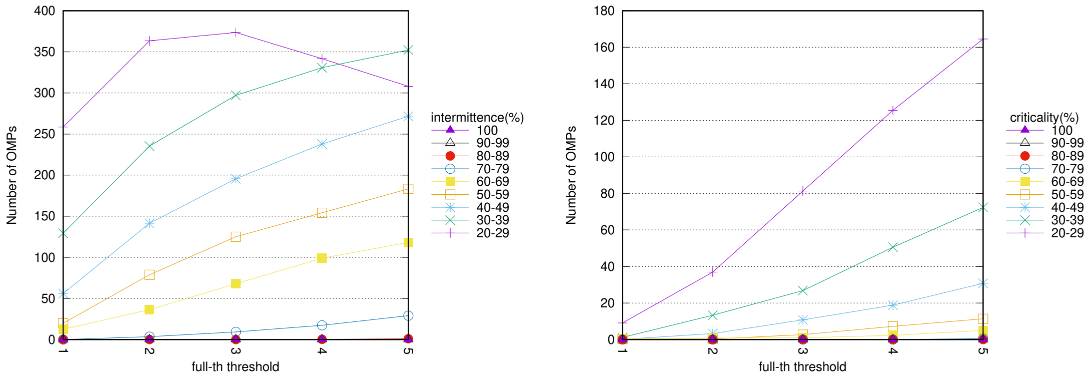
In the experimental evaluation one parameter was varied at a time, and the standard configuration was set for the remaining parameters. The standard configuration was introduced in Section 3.3.2 as *maxdist* = 0.5 km, *full-th*=3, *time slot size*=1h.

The results achieved on the *Bicing* dataset (Barcelona) reported hereafter are considered as reference example study. Similar results have been obtained from the *Citi Bike* dataset.

Occupancy threshold (*full-th*). Figures 3.8(a)-3.8(b) show the impact of the *full-th* parameter on the mined OMPs. The two figures report the total number of mined OMPs for each range of intermittence and criticality value when increasing *full-th*.

A station is in overloaded condition when less than *full-th* free docks are available. Therefore, the higher occupancy threshold value is set, the more OMPs with high intermittence/criticality value could be extracted. The results reported in Figures 3.8(a)-3.8(b) show this trend. The number of OMPs for each intermittence and criticality range increases almost linearly when increasing the *full-th* value. This increase is higher for the intermittence index.

Maximum distance threshold (*maxdist*). Figures 3.9(a)-3.9(b) show the impact of the *maxdist* parameter on the number of mined OMPs. The two figures report the total number of mined OMPs for each range of intermittence and criticality value when increasing



(a) Intermittence

(b) Criticality

Figure 3.8: Barcelona. Impact of the occupancy threshold on the characteristics of the mined OMPs. *maxdist*=0.5 km. *time slot size*=1h.

maxdist.

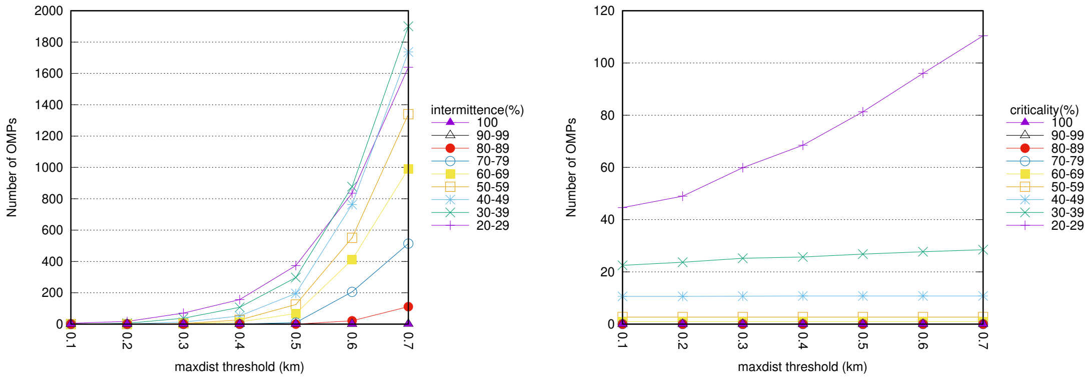
When the *maxdist* value is increased, the number of nearby stations also increases. Consequently, the number of mined OMPs increases because larger patterns including more stations are also generated. Results show that when increasing *maxdist* the number of OMPs increases almost exponentially for each intermittence range and almost linearly for each criticality range.

However, the number of OMPs that are worth considering for manual inspection (i.e., those with high intermittence/criticality values) remains roughly stable even while enforcing *maxdist* values higher than 0.5 km. Setting *maxdist* values higher or equal to 0.6 km is less interesting in the context of analysis of this study, because the end users are willing to move to physically closer stations if the expected destination is fully occupied.

Time slot size. The distribution of the number of extracted OMPs for each intermittence and criticality range when varying the time slot size were also analyzed. Experiments were performed for time slots ranging from 2 to 8 hours; as a representative example, Figure 3.10 reports the results achieved on the *Bicing* dataset with the 4-hours time slot.

Considering a courser time granularity to analyze collected data as, for example, a larger time slot size, can provide a high-level view of the station overload conditions in the bicycle sharing system. This view can be useful for end users but expecially for system managers to identify the time frames when usage conditions are critical. For instance, results in Figure 3.10(a) point out that the number of OMPs with high intermittence value (between 50%-59%) is significantly higher between 8.00am-12:00pm.

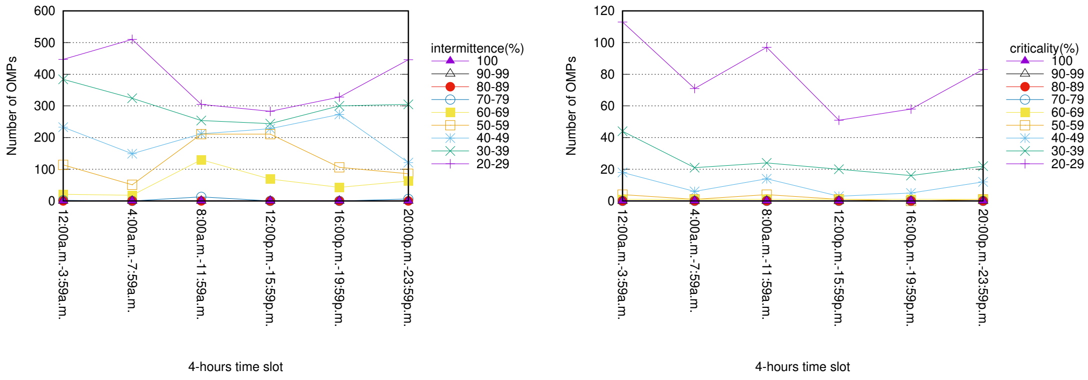
Domain-experts can then focus on each selected time frame to locally analyze collected data with a finer time granularity (i.e.,a time slot with lower size). This latter



(a) Intermittence

(b) Criticality

Figure 3.9: Barcelona. Impact of the maximum distance threshold on the characteristics of the mined OMPs. *full-th*=3. *time slot size*=1h.



(a) Intermittence

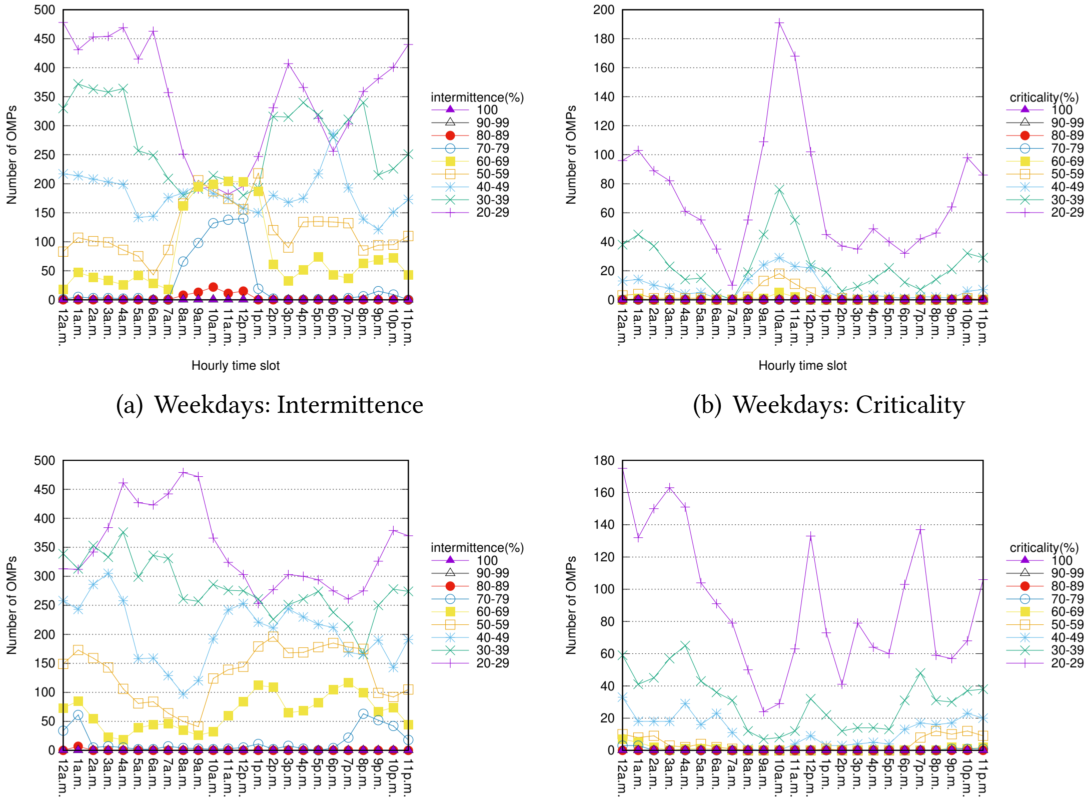
(b) Criticality

Figure 3.10: *Bicing* (Barcelona). Distribution of the number of OMPs and their corresponding levels of intermittence/criticality with a time slot granularity of 4 hours. *maxdist*=0.5 km. *full-th*=3. *time slot size*=4h.

analysis can provide more detailed information on dock overload conditions on each selected time frame.

In some cases, using time slots with a larger size could smooth local intermittence and criticality peaks of potential interest. For instance, few OMPs with intermittence in the range 70%-79% are mined with a 4-hour time slot (see Figure 3.10(a)). Instead, when considering 1-hour time slots, around 50 patterns with intermittence between 70%-79% are generated in the 10am, 11am, 12pm time slots (see Figure 3.4(a)).

Day category. Experiments have been performed to analyze the impact of the day category on the hourly distribution of intermittence and criticality. The OMPs extracted



(a) Weekdays: Intermittence

(b) Weekdays: Criticality

(c) Weekends: Intermittence

(d) Weekends: Criticality

Figure 3.11: Barcelona. Characteristics of the mined OMPs related to weekdays and weekends. *full-th*=3. *time slot size*=1h.

by considering the station occupancy log data related to workdays were compared to those mined by considering the weekends. Results are shown in Figures 3.11(a)-3.11(d).

Extracted OMPs show a significantly different trend in weekdays and weekends. More OMPs with higher criticality and intermittence values are mined in weekdays. These OMPs are mainly located in the time period from 7am to 2pm. In weekends, OMPs with high intermittence and criticality values (about 70%-79%) are mainly related to the period from 12am and 1 am and from 7pm to 11pm. Moreover, OMPs with high intermittence values are also mined for 2pm time slot.

These results highlight different usage of the bike sharing system of Barcelona during the days of the week. They support the need for different actions (such as bike re-balancing actions) depending on the type of day of the week that is considered. For example, bike re-balancing actions may be more relevant in weekdays than in weekends, and they must be scheduled in different time periods based on the day category.

3.3.4 Algorithm performance

The analysis of the performance of the OMP-Miner algorithm in terms of execution time was done. OMP-Miner requires time both for (critical and normal) o-itemset extraction and for the consequent generation of OMPs on top of the mined o-itemsets. The o-itemsets extraction is the most computationally expensive step. With the default parameter setting, the extraction time of o-itemsets is approximately 454s for *Bicing* (Barcelona) and 825s for *Citi Bike* (New York), while the time for OMP generation is a few milliseconds in both cases.

Also, the analysis of how the system parameters impact on the execution time was performed. Specifically, the analysis was focused on the maximum distance threshold *maxdist* which can impact significantly on the number of mined OMPs, and thus on the execution time. Experiments were run by varying the *maxdist* value while the standard configuration was adopted for the other parameters. The execution time, similarly to the number of mined OMPs, increases more than linearly with respect to the maximum distance threshold value. The time ranges from 3 minutes when *maxdist*=0.1 km up to 42 minutes when *maxdist*=0.6 km. The execution time increases to more than one hour when values of *maxdist* greater than 0.6 km are used, i.e., when *maxdist* is set to values that are considered not interesting in the considered application domain. Most of the execution time is spent on o-itemset generation, while even in the worst case the OMP generation requires a few seconds.

3.4 Summary

The *BELL* methodology analyzes historical occupancy data acquired from bicycle sharing systems with the aim of identifying situations of imbalance in dock occupancy levels of bike stations. The proposed methodology relies on an itemset-based approach, which extracts recurrent patterns from historical data and provides domain experts with a set of interpretable patterns to explore. The extracted OMPs describe the context (i.e., city area and time slot) in which a set of stations is in a critical/intermittent dock overload condition. The discovered patterns represent (i) groups of nearby stations whose slots are almost *all* occupied at most points of time, and (ii) groups of nearby stations among which *at least one of them* (but not all of them) has a high level of occupancy at most points of time (possibly in an alternate fashion).

The position of this study differs to a large extent from previous works in the literature. Specifically, (i) Previous works on clustering of the stations based on their usage profiles have been unable to identify intermittent dock overload situations. (ii) Studies on forecasting future occupancy levels of the stations have applied supervised techniques, while the methodology presented relies on an unsupervised technique (i.e., itemset mining). (iii) Previous approaches aimed at planning re-balancing actions are complementary to the proposed work because they can be applied to a subset of stations with intermittent dock occupancy levels.

As a general recommendation, it is advised to apply the *BELL* methodology to temporal intervals which are as homogeneous as possible with respect to any change that can affect the bicycle flow balance inside the system (e.g., rebalancing policy changes, additional docks in stations, or even additional stations). In any case, it is interesting to notice how the *BELL* methodology behaves when it is not possible to filter out these spurious data. Two kinds of anomalies can possibly emerge: false positive and false negative. The former is caused by imbalanced situations already solved by the policy modification but yet frequent enough in the data that precede the change itself (e.g., an overloaded condition in the city center solved by a scheduled bicycle rebalancing). The latter is a specular circumstance, critical/intermittent situations have started to occur after a change in the system but they are not enough represented in the dataset to be caught (e.g., possibly recent failures in some of the stations). Anyway, since the analysis is applied on a medium-term horizon at least, the above-mentioned anomalies are expected to be very limited and to be not significant when the *BELL*methodology is applied periodically.

The results achieved by the *BELL* methodology on real bicycle sharing system data have shown potentially harmful dock overload situations in the stations of bike sharing systems. Specifically, the applicability of the *BELL* methodology in two real case studies, the Barcelona and New York bicycle sharing systems, was explored. Notably, the achieved results show behaviors peculiar to each use case. For example, in New York the mined OMPs highlight situations of imbalance mainly due to intermittent occupancy levels (i.e., intermittence value=100%, criticality value=0%). This implies that although some areas were characterized by a strongly imbalanced bike distribution among stations in certain time slots, at least one station per area had a non-critical dock occupancy in the analyzed period. Hence, planning re-balancing actions could be sufficient to counteract situations of imbalance. Conversely, in Barcelona situations of imbalance were usually characterized by a mix of critical and intermittent conditions. Hence, rebalancing actions may be not sufficient and long-term maintenance actions (e.g., station resizing) need to be put in place to counteract the issue.

The takeaways from this study can be summarized as follows:

- The use of data mining tools to analyze bicycle sharing system data has become more and more attractive.

- Unsupervised approaches, like the *BELL* methodology presented in this study, characterize system usage in the medium- and long-term. They identify contexts in which user experience could worsen due to recurrent system inefficiencies.

- System users may take advantage of the data-driven approaches to system monitoring, because potentially critical situations can be automatically detected and managed without the need for explicit notification.

- Urban policymakers can exploit the *BELL* methodology to periodically monitor the dock overload situations detected in specific city areas at different time slots.

- Based on the knowledge extracted by the *BELL* methodology, policymakers could put in place *medium-term actions*, such as re-balancing actions triggered by the extraction of OMPs with high intermittence value, and *long-term actions*, such as station resizing or new station placement triggered by the extraction of OMPs with high criticality value.

- The results in the real case studies demonstrated the quality of the proposed methodology in supporting system managers under various aspects.

As future work, other data sources could be integrated to enrich the quality of the generated model. Variables such as the presence of environmental pollution, road network features, vehicular traffic, meteorological conditions, and the presence of cycling lanes as indicators of favorable/unfavorable conditions for bike sharing system usage could also be taken into consideration. The other interesting problem is to investigate the portability of the proposed methodology for different mobility services offered in urban contexts. For example, applying the proposed approach to charging stations of electric cars and to indoor car parks.

Additionally, the recent new bike-sharing mode appeared in many cities around the world, i.e. free-floating station-less bike sharing, could be the subject for a future extension. Splitting the city area homogeneously and applying the *BELL* methodology in a complementary way to measure the absence of bikes, it would be possible to analyze critical situations of clusters of adjacent city sectors, which completely or intermittently lack free-floating bikes. Also, in this case, the methodology can be applied to rebalance the bikes and/or increase the fleet to improve the users' perception of the overall service quality.

Chapter 4

Exploring data hierarchies to mine high-utility itemsets

Frequent itemset mining is an exploratory data mining technique which focuses on discovering recurrent combinations of items (of arbitrary size) that occur in potentially large transactional data [2]. Frequent itemsets have been used in many research contexts, among which market basket analysis [2], service profiling [11], to discover correlations between multiple data items. For example, in the context of market basket analysis, each row of the dataset (transaction) represents a different market basket. Transactions contain the subsets of purchased items. Frequent itemsets represent sets of items that customers frequently purchased together. For instance, itemset *{Coke, bread}* indicates that customers who purchased coke frequently purchased bread as well. Since generating all the possible combinations of items in a transactional dataset is computationally intractable [2], frequent itemset mining entails discovering only the combinations of items whose frequency of occurrence (support) is above a given threshold. However, the traditional itemset mining problem relies on three (potentially unreliable) assumptions:
(A) Items appear at most once in each transaction (e.g., disregarding the amounts of purchased items within each basket).
(B) Items have all the same importance in the analyzed data (e.g., the unit profit is assumed to be the same for all the items in the market).
(C) The semantic relationships between items are ignored (e.g., the co-occurrences of items belonging to the same product group within the same basket are considered as uncorrelated with each other).
To overcome limitations (A) and (B), the concept of High-Utility Itemset (HUI) has been proposed [51]. HUIs represent sets of frequently co-occurring items that are characterized by averagely high utility within the analyzed data. To mine HUIs, items in the transactional dataset are enriched with both per-transaction weights (hereafter denoted as *internal utilities*) and global weights (denoted as *external utility*). For example, in the context of market basket analysis internal utilities represent per-item amounts (e.g. the

customer purchased 3 bottles of coke), while external utilities indicate unit profits (bottles of coke cost 5 USD each). Utility itemsets represent sets of items whose total yield is above a given (user-specified) threshold. This knowledge may be exploited to perform cross-selling, to plan promotions, or to effectively arrange items on the shelves.

To overcome limitation (C), correlations between items at higher abstraction levels can be analyzed [74]. Based on a taxonomy built on top of the analyzed data, items are aggregated into semantically related groups (e.g. items *Coke* and *Water* into group *Beverage*). Then, generalized itemsets, which represent correlations among data items at different abstraction levels (e.g., not only {*Coke, bread*} but also {*Beverage, Food*}), can be extracted.

Many efforts have been devoted to efficiently address High-Utility Itemset Mining (e.g., [30, 31, 35, 47, 89]). To extract HUIs, several strategies have been proposed in the literature. The first attempts (e.g., [71, 20]) used a horizontal approach based on Apriori algorithm [3]. Among them is the well-known work of Liu et al.[51] based on a two-phase strategy, which consists of generating an overestimated solution according to the Transaction-Weighted-Downward closure model as a first step, and then refining it to discard non-profitable itemsets.

Another branch of studies (e.g., [81, 25]) followed a vertical approach inspired by the tree-based depth-first model of the FP-growth algorithm [39]. An example of this group is the algorithm UP-Growth of Tseng et al.[81] which proposed a compact data structure to store the pattern called utility-pattern tree (UP-tree) to mine HUIs more efficiently.

The most recent and advanced algorithm addressed either HUI mining in a single phase (e.g., [48, 45, 50]) or the generation of a compact HUI subset, e.g., the closed HUIs [32] and the top-k HUIs [80].

Although the existing solutions in the literature review are efficient in terms of temporal and spatial scalability, they are unable to cope with multiple-level data (limitation (C)). On the other hand, parallel works addressed the generalized itemset mining problem by performing bottom-up [11, 74] or top-down [38] taxonomy visits during candidate itemset generation. However, since all the mentioned studies do not consider item utilities, they still suffer from limitations (A) and (B).

This study aims at bridging the gap between HUI mining and generalized itemset mining. To this purpose, it proposes a new type of pattern, namely the Generalized High-utility Itemsets (GHUIs).

The proposed approach allows both multiple appearances of the same item within each transaction and different per-item profits. Unlike traditional HUIs, GHUIs are extracted from a transactional dataset enriched with a taxonomy, which describes the semantic is-a relationships between data items. These relationships are exploited to drive the process of knowledge generalization thus generating profitable combinations of items at multiple abstraction levels. To extract GHUIs the ML-HUI Miner is proposed, which extends a state-of-the-art HUI mining algorithm to cope with data enriched with taxonomies. The newly proposed algorithm integrates taxonomy information into the

Table 4.1: Transactional dataset

Transaction id	Items and internal utility
TID_1	(Coke, 2), (Bread, 2), (Steak, 1)
TID_2	(Water, 3), (Pasta, 2), (Steak, 1)
TID_3	(Water, 2), (Bread, 2)
TID_4	(Coke, 1), (Bread, 2)

Table 4.2: External utilities of items in dataset

Item	External utility
Water	1
Coke	5
Bread	1
Pasta	2
Steak	10

utility itemset mining process to mine GHUIs in a single-phase mining session.

Preliminary experiments performed on both real retail data and benchmark datasets show the efficiency and effectiveness of the proposed approach.

The chapter is organized as follows. Sections 4.1 and 4.2 introduce preliminary concepts and formalizes the newly proposed pattern, respectively. Section 4.3 presents the mining algorithm used to discover the newly proposed pattern. In Section 4.4 the experiments performed on real datasets have been summarized, while Section 4.5 draws conclusions and discusses future works.

4.1 Theoretical background

A transactional dataset is a set of transactions [2], where each transactions is a set of data items (i.e., objects identified by literals). Hereafter, let us denote as I the set of all possible items and as $t_j \in I$ the j-th transaction of a transactional dataset D. Items are characterized by (i) Internal Utility, denoted as iu(i,t_j) which indicates the relative importance of item $i \in I$ in transaction t_j, and (ii) External Utility, denoted as eu(i), which indicates the relative importance of item i in D with respect to all the other items in the dataset.

Table 4.1 reports an example of market basket dataset consisting of 4 transactions (identified by TID_i where $i \in [1,4]$)).

The utility of item i in transaction t_j, hereafter denoted as u(i,t_j), is computed as eu(i)· iu(i,t_j). In the running example, it indicates the total income related to an item

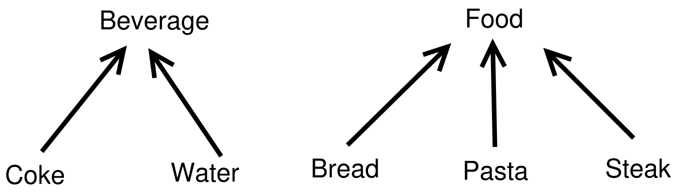


Figure 4.1: Taxonomy on items in the dataset

Table 4.3: High-Utility Itemsets. minutil (itemsets) = 17

Itemset	Utility
{ Steak }	20
{ Steak, Coke }	20
{ Coke, Bread }	19
{ Steak, Coke, Bread }	22

Table 4.4: Generalized High-Utility Itemsets. minutil (generalized itemsets) = 30

Generalized Itemset	Utility
{ Food }	30
{ Food, Beverage }	50

appearing in the market basket (e.g., the price of all the bottles of coke in a given market basket).

Itemsets are sets of items of arbitrary size. They will be denoted as k-itemset a set of k items. The utility of itemset $\boldsymbol{I}$ in transaction t_j is the sum of the utilities of all the corresponding items, i.e., u($\boldsymbol{I}$,t_j)=$\sum_{i \in I} u(i, t_j)$. The utility of itemset $\boldsymbol{I}$ in the transactional dataset $\boldsymbol{D}$ is obtained by summing the utilities of the itemset in all the dataset transactions, i.e., u($\boldsymbol{I}$,$\boldsymbol{D}$)=$\sum_{t_j \in D} u(I, t_j)$, where the condition u($\boldsymbol{I}$,$t_j$)=0 if $\boldsymbol{I} \notin t_j$ is assumed.

A notable type of itemset is the High-Utility Itemset. Given user-specified minimum utility threshold *minutil*, an itemset mined from dataset $\boldsymbol{D}$ is an High-Utility Itemset (HUI) if and only if u($\boldsymbol{I}$,$\boldsymbol{D}$)>*minutil*. Given a transactional dataset $\boldsymbol{D}$ and a minimum utility threshold *minutil*, the High-Utility Itemset Mining (HUIM) problem entails discovering all the HUIs in $\boldsymbol{D}$.

The HUIs extracted from the dataset in Table 4.1 by enforcing a minimum utility threshold *minutil*=20 are enumerated in Table 4.3, where the corresponding utility value is given too.

Example. {Coke, Bread} is HUI because the utility values of the 2-itemset for each transaction are: 12 (5x2+1x2) in TID$_1$, 0 in TID$_2$ and TID$_3$ (no matches), and 7 (5x1+1x2) in TID$_4$.

To efficiently extract HUIs, the Transaction-Weighted Utilization (TWU) has been introduced [79]. It is an over-estimate of the utility of the itemset, which can be exploited to prune the search space because it satisfies the following downward closure property: given two itemsets I_1 and I_2 such that $I_1 \subset I_2$, if the TWU of I_1 is below the utility threshold *minutil* even the TWU of I_2 does. The TWU of an itemset I, denoted as twu(I), is defined as the sum of the transaction utilities of all the transactions containing I, where the transaction utility of a transaction is the sum of the utility values of all its items. A formal definition of the TWU measure of itemset I follows: twu(I)= $\sum_{t_j \in \mathrm{D}|\mathrm{I} \subseteq \mathrm{t_j}} \sum_{i \in t_j} u(i, t_j)$.

4.2 Generalized High-Utility Itemsets

The goal of this study is to discover HUIs that incorporate knowledge at multiple granularity levels. To this aim, items are generalized at different abstraction levels.

Let T be a taxonomy (i.e., a is-a hierarchy), which aggregates items in I into higher-level concepts, hereafter denoted as *generalized items*. Generalized items represent higher-level categories which group individual items based on their semantic meaning. Let G be the set of generalized items in T. For the sake of simplicity, hereafter it is assumed that in the given taxonomy T each item $i \in I$ is aggregated into exactly one generalized item $g \in G$ (i.e., each item belongs to a specific higher-level category). Generalized items can be further generalized as other generalized items at higher granularity levels.

For each generalized item $g \in G$, Desc(g,T) $\in I$ denote the subset of descendant items of g according to the given taxonomy. In this study, the concept of *level* of a generalized item $g \in G$ in the taxonomy, hereafter denoted as l(g,T), is formalized as the length of the shortest path between g and any leaf node in the taxonomy. Note that, by construction, the level of non-generalized itemsets is zero, while the maximum level of an item corresponds to the taxonomy height (i.e., the length of the longest path from any node in T to a leaf node).

Example. Figure 4.1 depicts an example of taxonomy built on items occurring in the running example dataset (see Table 4.1). For instance, items *Coke* and *Water* are aggregated into the generalized item *Beverage*. The level of *Beverage* and is one, whereas the level of *Coke*, and *Water* is zero.

A generalized itemset is a set of generalized items in G. Similar to [11, 38, 74] the analysis is focused on the combinations of generalized items having the same level, because they compactly represent information at a given abstraction level. Hereafter the *level* of a generalized itemset is denoted as the level of its items.

In this study, the concept of utility has been extended to generalized items and itemsets. Specifically, the utility of a generalized item g in a transaction t_j, hereafter denoted as u(g,t_j), is the sum of the utility values of all the descendant items, while the utility of g in a transactional dataset is the sum of all the per-transaction utilities. More formal definitions follow.

Definition 4.2.1 (Utility of a generalized item)**.** Let g be a generalized item, D a transactional dataset, and T be a taxonomy. The utility of g in a transaction $t_j \in D$ is defined as u(g,t_j)=$\sum_{i \in Desc(g,T)}$eu(i)· iu(i,t_j). The utility of generalized item g in D is calculated as u(g,D)=$\sum_{i \in Desc(g,T)} u(i, D)$. □

Similar definitions hold on itemsets (i.e., sets of items). The utility of a generalized itemset in a transactional dataset indicates the overall profit of a combination of item categories.

Definition 4.2.2 (Utility of a generalized itemset)**.** Let GI be a generalized itemset and let D a transactional dataset, and T be a taxonomy. The utility of a generalized itemset GI in transaction t_j is defined as u(GI,t_j)=$\sum_{g \in GI} u(g, t_j)$. The utility of a generalized itemset GI is in the transactional dataset D is defined as u(GI,D)=$\sum_{t_j \in D} u(GI, t_j)$, where u($GI$,$t_j$)=0 if $GI \notin t_j$. □

Example. Let us consider again the market basket dataset reported in Table 4.1 and the taxonomy in Figure 4.1. The per-transaction utility of generalized item *Beverage* is 10 in transaction with TID_1, 3 in transaction with TID_2, 2 in transaction with TID_3, and 5 in transaction with TID_4. Hence, the utility of *Beverage* in the dataset is 20 (10+3+2+5).

This work is dedicated to the extraction of a selection of generalized itemsets, called Generalized High-Utility Itemsets (GHUIs).

Definition 4.2.3 (Generalized High-Utility Itemset)**.** Let *minutil* be a (user-specified) minimum utility threshold, let D be a transactional dataset, let T be a taxonomy, and let GI be a generalized itemset. A generalized itemset GI is a Generalized High-Utility Itemset (GHUI) in D if and only if u(GI,D)>*minutil*. □

Example. The GHUIs mined from the dataset in Table 4.1 by enforcing a minimum utility threshold for generalized itemsets equal to 30 for GHUIs are enumerated in Table 4.4.

Given a transactional dataset D, a taxonomy T, and a minimum utility threshold *minutil*, the Generalized High-Utility Itemset Mining (GHUIM) problem addressed by this work entails discovering all the HUIs and GHUIs.

Per-level utility thresholds. The utility of a generalized itemset incorporates those of all of its descendant itemsets. Hence, itemsets at higher abstraction levels are more likely to satisfy a fixed minimum utility threshold than lower-level ones. To overcome this issue, the utility threshold is adapted to the level of generalization of the considered itemsets. The key idea is to set higher utility thresholds for itemsets including items at higher abstraction levels. The same utility threshold is set for all itemsets having the same level. Specifically, given a user-specified least minimum utility threshold *minutil* associated with non-generalized itemsets (level=0), the minimum utility threshold thr(l) associated with level-l generalized itemsets is *minutil* if $l = 0$, $thr(l) = \alpha(l) \cdot minutil$ otherwise, where $\alpha(l)$: $\mathbb{N} \rightarrow [1, +\infty)$ is a monotonically increasing (user-specified) function.

Algorithm 3 The ML-HUI Miner algorithm

```
Require: transactional dataset $D$, taxonomy $T$, minimum utility threshold $minutil$, function $\alpha(l)$
Ensure: $O$, the set of High-Utility Itemsets and Generalized High-Utility Itemsets satisfying the per-level
    utility thresholds
    {Initializations}
 1: $I$← set of items in $D$
 2: $GI$← set of generalized items in $T$
    {Preparation}
 3: scan $D$ and $T$ to compute Transaction-Weighted-Utility (TWU) of items in $I$
 4: Compute Transaction-Weighted-Utility of items in $GI$
 5: $I^*$← set of items in $D$ such that TWU is above $minutil$(level=0)
 6: $GI^*$← set of generalized items $g$ in $T$ such that TWU is above $\alpha \cdot minutil$(l($g$,$T$))
 7: Build utility list and Estimated Utility Co-occurrence Structure of (generalized) items in $I^* \cup GI^*$
    {Recursive depth-first search}
 8: $O$ ←Recursive generation of the combinations of (generalized) items in $I^*$ and $GI^*$ whose items
    share the same level and selection of all combinations satisfying the utility threshold $\alpha(l) \cdot minutil$
```

4.3 The ML-HUI Miner algorithm

To extract Generalized High-Utility Itemsets, a new algorithm is proposed, namely Multiple-Level High-Utility Itemset Miner (ML-HUI Miner). The main features of GHUI-Miner are summarized below.

(a) *Taxonomy-driven HUI mining.* ML-HUI Miner supports transactional data enriched with taxonomy information. This allows us to extract patterns at different abstraction levels.

(b) *Single-step extraction of generalized and non-generalized HUIs.* The proposed algorithms explores the dataset and the taxonomy to generate multiple-level patterns in a single phase (i.e., without the need for multiple runs).

(c) *Prevent the generation of uninteresting combinations of items.* Similar to [38, 74], in this study the focus is on extracting itemsets including only items with the same level. Thus, GHUI-Miner prevents the generation of itemsets consisting of items with different levels in the taxonomy.

A high-level pseudo-code of the ML-HUI Miner algorithm is given in Algorithm 3. First, ML-HUI Miner scans the dataset and the taxonomy to identify the single non-generalized and generalized items whose Transaction-Weighted-Utility (TWU) is above the per-level utility threshold (Lines 3-6 in Algorithm 3). To this aim, the taxonomy is explored in a bottom-up fashion. The dataset and the accessory structures are properly adapted to prevent the generation of combinations of mixed-level items (according to Point (c). To compute the per-level utility thresholds, the user-specified threshold *minutil* is adjusted using function α according to the level of each generalized item. Then, the utility list associated with all non-generalized and generalized items is computed,

Table 4.5: Dataset characteristics

Dataset	Ext./Int. utility	Transactions	Items	Avg. transactions length
Retail	real data	19,514	2,741	20.1
Mushroom	synthetic data	8,124	119	23
Chess	synthetic data	3,196	75	37
Connect	synthetic data	67,557	129	43

whose TWU satisfies the per-level utility threshold (Line 7 in Algorithm 3). The utility list is a compact data structure that contains for each (generalized) item i (i) the list of transactions t_j such that $i \in t_j$, (ii) the utility values of the (generalized) item in each transaction u(i,t_j), and (iii) the sum of the utilities of the remaining items with the same taxonomy level within each transaction, i.e., $\sum_{q \in t_j \wedge\, q \neq i \wedge l(q,T)=l(\mathrm{i,T})} U(q, t_j)$. In this study, the utility list proposed in [48] to integrate information about the generalized items at the same taxonomy level appearing in the taxonomy is extended. The utility list is provided as input to the depth-first recursive procedure (Line 8 in Algorithm 3), which does not need to access neither the dataset nor the taxonomy. Specifically, starting from single items (generalized and not) the recursive procedure computes their exact utility value and then explores all their extensions using a depth-first strategy based on the utility list. To avoid generating itemsets consisting of items with different level, extensions are selectively generated. The recursive procedure is similar to the one adopted by FHM [48].

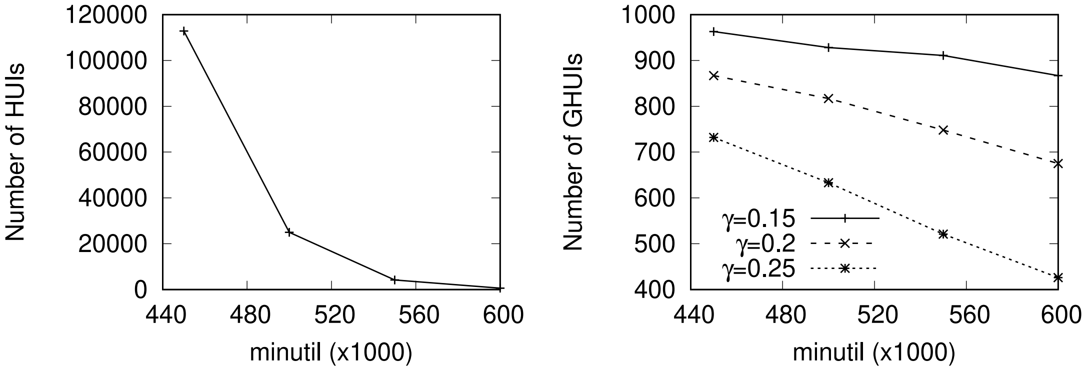
4.4 Experiments

To evaluate the performance of the ML-HUI Miner algorithm, experiments were conducted on four benchmark UCI datasets coming from different domains (*Connect*, *Mushroom*, *Chess*, *Retail* [29]), which have already been used to evaluate the performance of recently proposed High-Utility Itemset mining algorithms (e.g., FHM [48]).

Table 4.5 summarizes the main characteristics of the analyzed datasets, where the number of transactions per dataset varies from 3,196 (*Chess*) to 67,557 (*Retail*). The number of distinct items lies in the interval from 75 (*Chess*) to 2,741 (*Retail*).

It is important to note that the *Retail* dataset [29] stores real sales of an online store, where each transaction corresponds to a different sale, while items represent products. Also, it is the only dataset containing real values of external and internal utilities. The external utility represents the number of products bought, while the internal utility indicates the prices of the items purchased.

Instead, the values of external and internal utilities of the remaining datasets (i.e.,



(a) non-generalized HUIs

(b) GHUIs

Figure 4.2: Chess: number of mined patterns

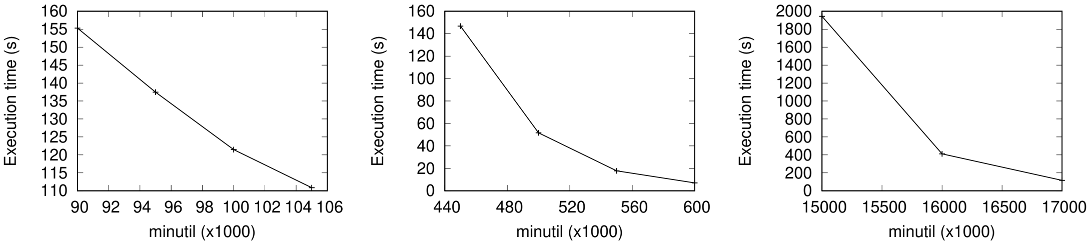
Connect, *Mushroom*, *Chess*) were synthetically generated by using a log-normal distribution. The external utilities range from 1 to 1,000, whereas the internal ones are uniformly distributed between 1 and 5.

To set the per-level utility threshold values ($thr(l)$), $\alpha(l)$=$\gamma \cdot f(l)$ was defined, where $f(l)$ is the average number of level-0 descendants per level-l item.

To generalize items at higher abstraction levels, taxonomies on top of data items were generated. Specifically, the *Retail* dataset, was enriched with a real 2-level taxonomy coming from a prominent online store, which allowed to aggregate products into the corresponding product group. The 2,741 available products are clustered into 38 product groups (e.g., *Kitchen*, *Toy*, *Home*, *Office product*, and *Musical Instrument*). For instance, *Kitchen* clusters 1,513 products, while *Toy* 236. On the other datasets, a synthetic taxonomy was generated, where each generalized item aggregates, on average, 10 randomly selected products. The experiments were performed on a 2.67 GHz Intel Xeon workstation with 32 GB of RAM, running Ubuntu 12.04.

4.4.1 Performance analysis

Figure 4.2 shows the number of mined (non-generalized) HUIs and GHUIs by varying the values of *minutil* and γ on a representative dataset (i.e., *Chess*). For both types of patterns, the number of mined itemsets is inversely proportional to the *minutil* value. By decreasing the values of *minutil* and γ the total number of mined patterns superlinearly increases. The number of GHUIs is two orders of magnitude lower than those of HUIs due to the significantly lower number of possible item combinations at higher taxonomy levels. The increase in the number of candidate itemsets due to taxonomy integration implies also a time complexity increase. Figure 4.3 shows the time spent by the ML-HUI Miner algorithm on the analyzed datasets with decreasing *minutil* values. Since the value of γ affects only the extraction of HGUIs, whose number is orders of



(a) Mushroom

(b) Chess

(c) Connect

Figure 4.3: Execution time (γ=0.2)

magnitude lower than the number of HUIs, varying γ value slightly affects the execution time. For this reason, the execution time is reported only for the representative value γ=0.2 on all datasets. Despite a larger number of combinations were explored, the extraction times remain acceptable (few ms) on all the analyzed datasets. The execution times of the newly proposed ML-HUI Miner were compared with those of the FHM algorithm [48], which extracts only non-generalized HUIs. ML-HUI Miner execution time approximately doubles that FHM time, even when more than two aggregation levels are integrated in the taxonomy (i.e., adding further top levels does not significantly increase the extraction time).

4.4.2 Knowledge discovery

Let's consider an example GHUIs mined from the *Retail* dataset($minutil$=10000, γ=0.2). The GHUIs with length 1 reveal the most profitable product groups. For example, *Kitchen* is the category with maximal utility. *Toy* ranked second (utility gap from *Kitchen* to *Toy* 87%), *Home* is the third (92%), *Office product* ranked 4th (96%) and *Law & Patio* 5th (96%). For all the remaining categories the utility gap with respect to *Kitchen* was above 97%. The GHUIs with length greater than 1 point out combinations of groups that yielded high profits when the respective products were sold together. Let us consider, for instance, GHUI {Musical-Instruments, Home}. It indicates that the jointly sale of products in these categories provided a high income. Among GHUIs including group *Musical-Instruments*, GHUI$_i$={*Musical-Instruments*, *Kitchen*} is the one with highest utility. The other GHUIs in order of decreasing utility are: {Musical-Instruments, Toy} (utility -75% w.r.t. GHUI$_i$), {Musical Instruments, Home} (utility -91% w.r.t. GHUI$_i$), and {Musical-Instruments, Office-Product} (utility -94% w.r.t. GHUI$_i$). These patterns can can be exploited to figure out which categories of products should be promoted in the same advertising campaign, e.g., while planning a campaign on products belonging to *Musical Instrument*, products of *Kitchen*, *Toy*, *Home* or *Office-Product category* should be advertised as well. The utility value provided us a ranking of the most appealing groups of products to advertise together with products of *Musical Instrument.* GHUIs provide high-level knowledge related to single products that do not satisfy the utility threshold. Let us consider again group *Musical Instrument*. Only two non-generalized HUIs were extracted, i.e., {Red-Harmonica} (utility 26,205) and {Blue-Harmonica} (utility 10,271). However, the utility of the GHUI {Musical-Instruments} is 40,741. Hence, the utility value of {Musical-Instruments} is not only due to the {Red-Harmonica} and {Blue-Harmonica} products, but even to other products of the same group that do not satisfy the utility threshold. Considering GHUI {Musical-Instruments} allows us to consider also the contribution of the other products, even if their single profits are averagely low. Moreover, GHUIs represent correlations among products that can not be easily inferred while considering only HUIs. For instance, even if correlations between products of group {Musical-Instruments} and products of other groups have not been extracted, the high-level correlations between *Musical Instrument* and other groups were extracted and can be analyzed.

4.5 Summary

This study describes a new pattern, called GHUI, which represents sets of items groups, each one characterized by a high total profit. The significance of the proposed pattern and the performance of the proposed GHUI mining algorithm have been evaluated on retail data, with the goal of planning advertising campaigns of retail products. Future extensions of the work could address the efficient extraction of significant subsets of GHUIs (e.g., closed or minimal GHUIs).

Chapter 5

Integration of business activities between different web directories

A pervasive problem on the web is represented by the integration of information coming from various sources. Specifically, all the major web directories of national and international level (e.g., Google Maps, Facebook Pages, Pagine Gialle) are continuously acquiring new information regarding business activities from external partner organizations. The acquired information is typically described using a taxonomy of categories. However, the taxonomy used by external partners is often significantly different from the one used internally by the company. Therefore, a correct categorization of the newly acquired information is a critical issue.

To address this problem, the approach usually applied is to create a static mapping between each category of the source taxonomy and one of the categories of the target taxonomy. The mappings can be created manually or by means of semi-automatic specialized software tools (e.g., PROMPT [60], OBSERVER [55], HCONE [44]). Still, this approach is ineffective when the levels of granularity of the source and target taxonomies are different. For example, a generic restaurant category, which lacks any specification, cannot be mapped to a finer level of granularity, such as an Indian, Chinese, or Italian restaurant.

To overcome this limitation, a system named *TACOMA* (*T*ext-b*A*sed *C*ateg*O*ry *MA*pping) is introduced in this study. In *TACOMA* the issue of mapping among taxonomies has been reformulated as a *classification problem*. Each target category could be treated as a class, and all the significant information about the business activity is taken into account to build a classification model. Additionally, the multi-level structure of a taxonomy allows to compensate the conceptual distance among inherently different taxonomies. When the target category of a specific instance cannot be found using classification, the parent, i.e., more general category can be used. The goal is to provide the most cohesive set of categories from different taxonomies corresponding to one and the same business activity.

The approach proposed in this thesis has been validated using different classification algorithms (e.g., Artificial Neural Networks, Support Vector Machines, Random Forest). The experiments were executed on a real dataset regarding business activities coming from a prominent business activity web directory (i.e., Pagine Gialle[1]). Due to the good level of accuracy, the results of this study have been integrated into the company production system.

5.1 Related work

The main argument of this study can be considered as a specific case of the more general issue of the ontology mapping, which is yet an open topic despite the fact that it has been extensively researched. Moreover, as reported in the survey performed by Falconer et al. [27] (a group of researchers from the universities of Stanford and Victoria), the fundamental ontology mapping issues, such as different model conceptualizations and language ambiguity, are difficult, if not impossible problems to solve even for a team of humans.

Thor et al. [78] have summarized the principal methodologies used to achieve a mapping among different ontologies as: (i) metadata-based, when just the information contained in the ontology are exploited to perform the mapping; (ii) instance-based, when the specific instances of each concept are leveraged in the process; and (iii) mixed-forms, when both metadata and instance approaches are combined.

Aanen et al. [1] proposed a metadata-based approach of product taxonomy mapping that incorporated an extension of the Park and Kim algorithm [63]. The work of Aanen et al. is mainly targeted at improving the process of the word sense disambiguation, which is out of scope of the current study.

Agrawal and Srikant [4] have proposed an example of instance-based approach, which leverages a modified Naïve Bayes classification to correctly map a source taxonomy to a master one. Their contribution was to exploit a locality principle: documents coming from the same source category will have a tendency to be placed into the same target category.

A mixed-form schema mapping approach called *SimilarityFlooding* is presented by Melnik et al. [54] using a graph-based database for its implementation. The data structures to be matched can be represented by data schemas, data instances, or both. To perform the initial mapping of the nodes, a simple string similarity is applied, which later transforms into calculation of similarities of schema nodes that are directly related (i.e. adjacent). The algorithm by Melnik et al. can be applied to taxonomies, however, according to [27] a possible downside is the lower efficiency of the process due to the hierarchical structure of the data.

A repository that represents a valuable ontology matching resource is the *Ontology Matching* website [72] which incorporates a list of state-of-art studies dedicated to this

[1] https://paginegialle.it

topic. Hereafter follow several studies considered particularly relevant in the field of ontology alignment by means of classification.

Nkisi-Orji et al. [59] have leveraged a random forest classifier to automatically match feature vectors composed of a combination of similarity measures. The advantages of this approach are: the ability to handle knowledge-light ontology resources and the absence of the need to calculate weights for each of the components of the similarity measures in the feature vector. However, the approach [59], such as any metadata-based approach, is not applicable to match categories at different granularity levels, and it cannot be applied in case of multi-language ontologies like the ones considered in this chapter.

Kejriwal and Miranker [43] have proposed an instance-based semi-supervised approach to match ontologies which reduces the effort of creating a sufficiently populated training set. The methodology relies on an ensemble of classifiers which are trained iteratively using only a small fraction of training data (i.e., around 2% of the data required by a standard supervised technique), thus, obtaining performance comparable to a fully supervised approach. Despite being an interesting technique, the approach proposed in [43] may provide a limited support in the context of this study, since a substantial number of labeled samples was already available from the beginning. In addition, a human effort was required to provide samples for the uncovered classes, and to filter out the ambiguous samples (e.g., merchants falling in multiple categories such as restaurants which offer Chinese and Japanese cuisine at the same time).

Another worthful resource, the *Ontology Alignment Evaluation Initiative* (OAEI) [26], is an established instrument to evaluate the ontology matching solutions in terms of an annual competition performed on a common dataset and benchmarking platform. Among the most recent proposals, Destro et at. [23] address the problem of matching multi-lingual ontologies using English as a pivot language and relying on a mixture of semantic and string distances to evaluate the result. Regardless of being a metadata-based approach, which is not applicable in the specific case of this study, the proposed methodology could be leveraged as a possible extension of the current study. This would automate the process of mapping categories at the same level of granularity but different in terms of language such as Apple (English) and Pagine Gialle (Italian), currently performed manually with the help of a domain expert.

5.2 Theoretical background

This section contains a reference to the *Term frequency - Inverse Document Frequency* technique used by *TACOMA* .

Term Frequency (TF) and Inverse Document Frequency (IDF)

Preliminaries Let C be a collection of documents, $D = \{d_1, \ldots, d_n\}$ the set of documents in C, and $T = \{t_1, \ldots, t_k\}$ the set of terms in C.

Definition 5.2.1 (Term Frequency (TF))**.** For each pair (d_i, t_j) in C, the *Term Frequency* TF_{d_i,t_j}, is the relative frequency of the term t_j in the document d_i. It is computed as $f_{d_i,t_j} / \sum_{1 \leq k \leq |\sum|} f_{d_i,t_j}$, where f_{d_i,t_j} is the number of times the document d_i contains the term t_j and $\sum_{1 \leq k \leq |\sum|} f_{d_i,t_j}$ is the total number of terms contained in d_i. □

Definition 5.2.2 (Inverse Document Frequency (IDF))**.** The *Inverse Document Frequency* IDF_{t_j} for a term t_j is the frequency of t_j in C. It is computed as $\log(|D|/|d_k \in D : f_{d_k,t_j} \neq 0|)$ where $|D|$ is the number documents in C and $|d_k \in D : f_{d_k,t_j} \neq 0|$ is the number of documents in C which contains at least one term t_j. □

Mathematically, the base of the log function for IDF computation does not matter and constitutes a constant multiplicative factor towards the overall result. The TF-IDF weight w_{d_i,t_j} for the pair (d_i, t_j) is high when the term t_j appears with high frequency in documents d_i and low frequency in documents in the collection C.

When the term t_j appears in more documents, the ration inside the IDS's log function approaches 1, and the IDF_{d_j} value and TF-IDF weight w_{d_i,t_j} become close to 0. Hence, the approach tends to filter out common terms. A more formal definition of TF-IDS follows.

Definition 5.2.3 (TF-IDF weight (w))**.** For each pair (d_i, t_j) in C, the TF-IDF weight w_{d_i,t_j} is computed as $w_{d_i,t_j} = TF_{t_i,d_j} * IDF_{t_j}$ where TF_{t_i,d_j} is the *Term Frequency* and IDF_{t_j} is the *Inverse Document Frequency*. □

5.3 Motivation and an industry use case

The thorough analysis of the approaches mentioned in Section 5.1 showed that the previous studies don't address a fundamental issue, which is performing the taxonomy mapping of the nodes at different levels of abstraction. Identification of this problem served as a motivation to fill in the research gap in terms of this thesis. Specifically, the objective of this study is to propose a method to map the instances of a certain hypernym category of the source taxonomy with its hyponyms in the target taxonomy.

Let's consider a running example of the issue coming from the industry which is illustrated in Figure 5.1. The taxonomy of Pagine Gialle (on the left) is organized rigidly according to three levels, while the Apple Maps[2] taxonomy (on the right) has a more flexible schema structure, with a variable number of hierarchy levels — currently from 3

[2] https://mapsconnect.apple.com/

to 5 — and extendible whenever necessary. The more generic hypernym *Ristoranti*, at the third level of the source taxonomy of Pagine Gialle, has a straightforward mapping with the target category *restaurants*, at the first level of the Apple Maps taxonomy. Nevertheless, this kind of mapping "hides" the 332 hyponyms available in the target Apple Maps category, ranging from *restaurants.italian.sicilian* to *restaurants.chinese.shanghainese*.

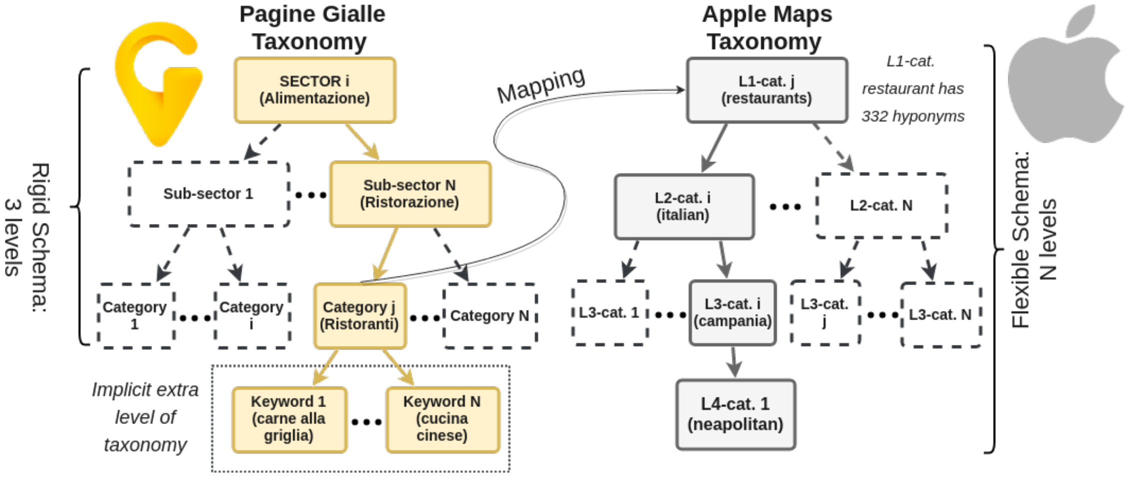


Figure 5.1: Mapping of categories at different abstraction levels.

To overcome this methodology limitation, *TACOMA* has been developed as an instance-based approach, to achieve the correct categorization of all the hyponyms in the target taxonomy leveraging a classification model. As illustrated in Figure 5.1, the *keywords*, *name*, and *description* fields altogether compose an auxiliary level of taxonomy that is exploited by the classification model to produce a mapping among the source and target categories.

It is worth to mention that *TACOMA* taxonomy mapping approach is agnostic of the schema structure, i.e., the number of levels in each taxonomy may be arbitrary. Using a *hierarchical classification* approach, each target taxonomy level could be treated independently: the result obtained by the classifier of a certain target level can be further refined using another classifier to descend the target taxonomy level if needed. In addition, the language used to define each Pagine Gialle taxonomy category, as well as their instances, is Italian. Instead, the working language of the Apple Maps taxonomy is English. All the methodologies reported in Section 5.1, apart from the one of Agrawal [4], would have failed to perform the mapping correctly due to this language mismatch. In its turn, another advantage of *TACOMA*, as an instance of a classification approach methodology, is to automatically overcome this language discrepancy.

5.4 Methodology

5.4.1 Data preparation

The labeled dataset used to train the classification model has been composed with the data coming from Pagine Gialle and Apple Maps information sources. Initially, noticing that Apple Maps has adopted the Yelp taxonomy to categorize its business activities, a custom software crawler was developed to obtain labeled data directly from the Yelp website[3]. Employing the Pagine Gialle business activities dataset as an input, which was composed of nearly 170000 entries, the crawler obtained around 3% of them (5000) as labeled matches on Yelp. Another contribution to the training set was given by Apple Maps, which has provided approximately 1000 manually labeled entries coming from their data quality review.

However, there were still two issues to be tackled. (i) Since *TACOMA* has been designed to address specific branches where the source taxonomy has a coarser granularity than the target taxonomy, just a fraction of these specific categories were covered by the training set. (ii) Additionally, the entries among the needed classes were strongly unbalanced. To deal with both issues, i.e., the lack of data and the unbalanced dataset, Pagine Gialle has given its contribution providing the labeled entries. Even synthetic samples were created when the real ones were lacking to equally populate the appropriate classes.

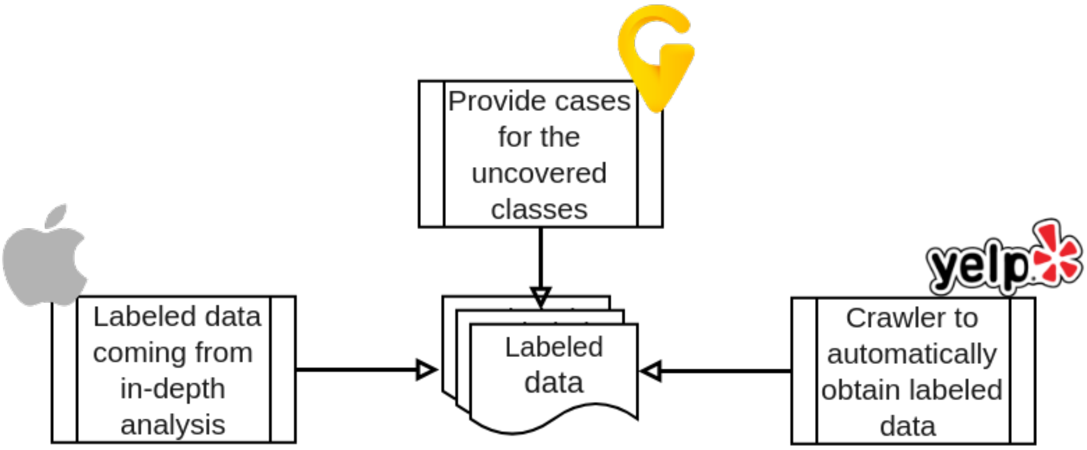

Provide cases for the uncovered classes
Labeled data coming from in-depth analysis
Labeled data
yelp
Crawler to automatically obtain labeled data

Figure 5.2: The training set building process

[3] https://www.yelp.it

5.4.2 Dataset description

The dataset resulted from the process illustrated in Section 5.4.1 is composed of 8 fields. Table 5.1 shows a real example reporting the data of a creperie. The first field is the *label*, available just in the training set, while the others are split in two groups: (i) pure free text fields (i.e., *name*, and *description*), (ii) and five keyword fields, which contain keywords coming from a set tailored per business activity (i.e., *activities*, *specialties*, *services*, *products*, and *brands*).

Empirical verification has shown that the free text fields, i.e., *description* and *name*, significantly contribute in some cases to accurately categorize a business activity working as an implicit source for keyword enrichment.

In the Pagine Gialle / Apple Maps use case, analyzing the taxonomies involved, four cases were found that required the application of *TACOMA* approach: *food*, *shopping.fashion*, and *restaurants* with its subset *restaurants.italian* which in its turn has to be processed to further refine the business activity category. Table 5.2 summarizes the characteristics regarding the various training sets leveraged in the use case.

Attribute	**Value**
Label	restaurants.creperies
Name	Crispus
Description	Located in the town of Alberobello ... it is a place where you can stay until late at night for a tasty snack.
Activities	”dinner aperitif”
Specialties	”main dishes” / ”traditional cuisine” / ”homemade desserts”
Services	-
Products	”sandwiches” / ”pizza” / ”ice cream”
Brands	-

Table 5.1: Example of a labeled entry from the training set (original data in Italian).

5.4.3 Classifier architecture

Figure 5.3 shows the processes to train the classification model, test it, and finally use it to classify unlabeled data. The three phases have in common the pre-processing steps used to prepare the textual data for the classifier to train the model or query it. Each of the input dataset fields (e.g., keywords, description) is filtered from the punctuation and tokenized into single words. The smallest tokens (i.e., less than 3 characters long)

Dataset	Level	Number of classes	Samples per class	Entries
restaurants	1	25	17	425
restaurants.italian	2	15	17	255
food	1	32	17	544
shopping.fashion	1	15	17	255

Table 5.2: Training set statistics.

are then eliminated along with the stopwords [4]. Then, the *stemming* step is applied, reducing each word to its *root term*. For instance, words like *fishing*, *fished*, and *fisher* will all be reduced to their stem *fish*.

The *TF-IDF* step (Term Frequency–Inverse Document Frequency, see Section 5.2) is then applied on the stems to obtain a series of vectors, one for each business activity case, containing a weight for each stem. The TF-IDF weight increases proportionally to the number of times the stem is repeated in the specific business activity data. Conversely, the weight is offset by its frequency in the entire business activities dataset. This effect compensates the high frequency of the most common, not significant words (and so, of their stems).

The stem matrix is then given as an input to the *SVM* classifier (Support Vector Machine [67]) to train the model or, alternatively, to query it to obtain an appropriate business activity category prediction.

To increase the classifier accuracy a *hierarchical classification* approach has been used when the target taxonomy has more than one finer levels with respect to the source taxonomy. Any additional finer level present in the target taxonomy is dealt with using a specialized classifier. In our use case, for instance, as reported in Table 5.2, when the *restaurants* classifier assigns a *restaurants.italian* category, the data regarding the business activity being analyzed are then passed to a second level classifier trained to recognize 15 different kinds of Italian restaurants.

In addition, each classifier prediction is evaluated against a given user-specified confidence threshold. If the confidence reported by the classifier is below a given threshold, the upper level category of the target taxonomy is used as a result. Namely, the category *restaurants.italian* will be used in place of a low confidence prediction of *restaurants.italian.altoatesine*.

[4] Commonly used and non significant words like *the, is, at, which, and on.*

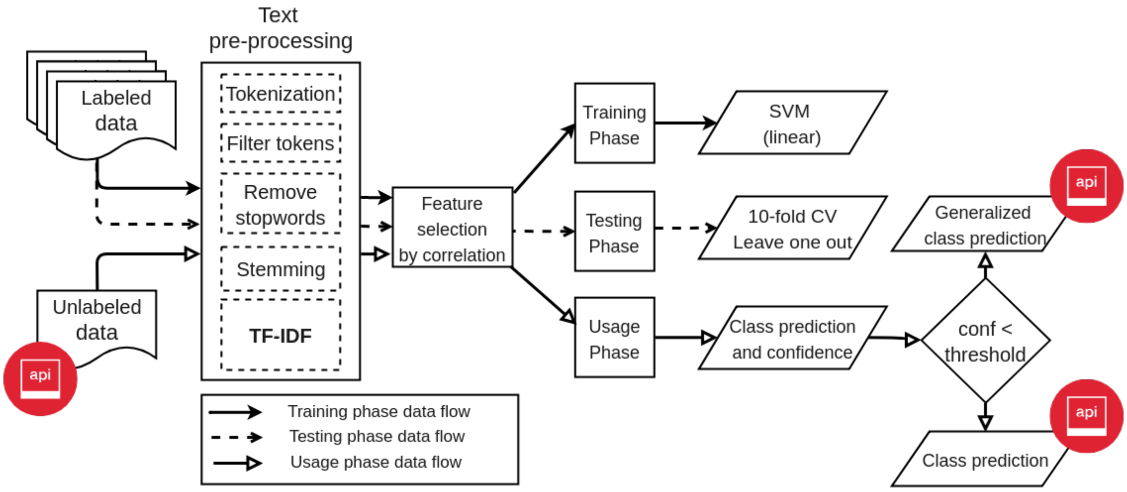


Figure 5.3: The TACOMA classifier architecture.

5.5 Experimental validation

The classification pipeline illustrated in Figure 5.3 has been developed using the fast prototyping environment Rapidminer [56]. During the experiments several kinds of classifiers (i.e., Decision Tree, Random Forest, Naive Bayes, GLM, Neural Net, SVM with different kernels) have been tested with the support of a grid search to optimize their parameters. The results reported hereafter are relative to the SVM classifier with a linear kernel, which is the one that has best performed with the analyzed dataset. The classification pipeline has been then deployed as a web service to be easily queryable from the software module responsible to assign the Apple Maps target category to each Pagine Gialle business activity.

The leave-one-out cross-validation (*LOOCV*) method has been adopted to evaluate the classification model. Using *LOOCV*, the model is trained against the entire dataset but for one single sample for which a prediction is made with the trained model. The values of the *accuracy*, *precision*, *recall*, and *f1-score* classifier performance indices are then updated according to the outcome of this prediction (i.e., correctly or incorrectly made). The process is then repeated, leaving out another sample and training the model with the rest, till the entire dataset have been covered.

The accuracy, measuring the overall quality of the classifier, is the ratio of the number of correctly classified business activities over the total number of them. Precision and recall analyze the performance of the classifier with respect to a given class *c*. Precision is defined as the number of business activities correctly classified in *c* divided by the total number classified in *c*. Recall is the number of business activities correctly classified in *c* divided by the number labeled with *c* in the dataset. *F1-score* is used to combine *precision* and *recall* in a single value and it is equal to their geometric mean.

Classifier	**Accuracy**	**Average precision**	**Average recall**	**Average f1-score**
restaurants	94.80% ± 22.20%	96.10%	94.84%	95.34%
restaurants.italian	94.12% ± 23.53%	95.55%	94.12%	94.63%

Table 5.3: Performance indices of the *restaurants* and *restaurants.italian* classifiers.

Class	**Precision**	**Recall**	**F1-score**
restaurants.japanese	88,24%	88,24%	88,24%
restaurants.tradamerican	100,00%	94,12%	97,02%
restaurants.brazilian	100,00%	94,12%	97,02%
restaurants.creperies	100,00%	94,12%	97,02%
restaurants.arabian	100,00%	94,12%	97,02%
restaurants.african	100,00%	94,12%	97,02%
restaurants.asianfusion	86,67%	76,47%	81,41%
restaurants.bbq	88,24%	88,24%	88,24%
restaurants.chinese	94,44%	100,00%	97,18%
restaurants.indian	100,00%	94,12%	97,02%
restaurants.pizza	94,44%	100,00%	97,18%
restaurants.seafood	93,75%	88,24%	90,95%
restaurants.vegan	100,00%	88,24%	93,94%
restaurants.hotdogs	100,00%	94,12%	97,02%
restaurants.italian	56,67%	94,44%	73,16%

Table 5.4: Performance of a subset of classes of the level 1 *restaurants* classifier.

Table 5.3 shows the values of the above indices for both the level 1 and 2 classifiers (i.e., *restaurants*, and *restaurants.italian*). In addition, Tables 5.4 and 5.5 illustrate the performance of a subset of the classes involved.

The accuracy value of the *restaurants* classifier is quite high (94.8%), with very close values for the *average precision* and *recall*, respectively 95.55% and 94.12%. The very high values, supported with the sample tests performed on the production data, guarantees the quality of the classification model. It is worth to comment the results relative to the *restaurants.italian* class, for which the precision value (56.67%) is the lowest among

Class	Precision	Recall	F1-score
restaurants.italian.apulian	100,00%	88,24%	93,94%
restaurants.italian.calabrian	94,44%	100,00%	97,18%
restaurants.italian.emilian	94,44%	100,00%	97,18%
restaurants.italian.friulan	100,00%	88,24%	93,94%
restaurants.italian.ligurian	100,00%	94,12%	97,02%
restaurants.italian.lumbard	61,54%	94,12%	76,11%
restaurants.italian.napoletana	93,33%	82,35%	87,67%
restaurants.italian.piemonte	100,00%	94,12%	97,02%
restaurants.italian.roman	100,00%	94,12%	97,02%
restaurants.italian.sardinian	89,47%	100,00%	94,59%
restaurants.italian.sicilian	100,00%	82,35%	90,75%
restaurants.italian.tuscan	100,00%	94,12%	97,02%

Table 5.5: Performance of a subset of classes of the level 2 *restaurants.italian* classifier.

all the other classes. This fairly low precision indicates that nearly the 50% of the predictions regarding the class are false positives. At the same time, the very high recall (94.44%) shows that almost all the real *restaurants.italian* cases are correctly categorized. This is most probably caused by the large variability of the cases falling into the *restaurants.italian* class, which has 15 different sub-categories.

A similar scenario is found exploring the performance indices of the *restaurants.italian* classifier which has again a very high *accuracy*: 94.12% and close values for the average precision, recall and f1-score. The *restaurants.italian.lumbard* class follows the same lower precision (61.54%) / high recall (94.12%) pattern previously analyzed in the *restaurants.italian* class while *restaurants.italian.friulan* has an opposite outcome. Its precision is 100%, but the lower recall value (88.24%) indicates that more than the 10% of the cases have been mistakenly placed in a different category.

5.6 Summary

This study defines a classification model to support the integration of business activities among different web directories (e.g., Pagine Gialle, Google Maps, Facebook pages) characterized by taxonomies of different granularity levels. In particular, it addresses the problem of a source taxonomy, which has categories at a coarser conceptual level of granularity of the target taxonomy. For instance, a source taxonomy with a hypernym concept of *furniture* whose instances should be mapped to the hyponym concepts of *chair*, *table*, or *bookcase* in the target taxonomy. The issue is addressed using an instance-based classification approach, where the textual data of each case is leveraged to correctly predict the appropriate category of the target taxonomy. The experiments performed on real data coming from a prominent Italian web directory (i.e., Pagine Gialle) have proven the efficacy of the proposed methodology, which has been already integrated in the production system of Pagine Gialle to export its data towards an international partners such as Apple Maps.

Chapter 6

Conclusions

The research activity described in this PhD thesis has focused on the design and development of proper techniques for the integration and analysis of huge volumes of heterogeneous data in urban and business application areas. Original contributions of this PhD thesis are the study and development of *novel data analysis frameworks* and *novel patterns* to extract useful insights from the targeted data collections tackling important issues such as the large dataset cardinality, dimensionality, and the variable data distribution. In the proposed frameworks, these results are obtained by means of enriching the data with *taxonomies* placed on top of them with the aim of analyzing data at multiple abstraction levels.

In Chapter 2, the fundamental issue of monitoring the air-pollution in the urban environment is addressed. Two data mining frameworks, *GECKO* (GEneralized Correlation analyzer of pOllution data) [14] and *ARQUATA* (AiR QUAlity patTern Analyzer) [12], are described. The data mining system *GECKO* leverages the power and expressiveness of the generalized association rules to extract, interpretable correlations among air pollution related data at different granularity levels. The data analyzed by *GECKO* system covered various aspects of air quality such as meteorological conditions, acquisition times, and vehicular traffic measurements. In its turn, the data analysis performed in *ARQUATA* is targeted at discovering combinations of pollutant concentrations that averagely are in a critical condition. The weighted frequent itemset pattern, is exploited to extract the novel type of pattern designed for this purpose, the *air quality pattern*. To provide different insights to domain experts and municipality actors, these patterns are extracted from several aggregations of the raw data following specific temporal and spatial granularities. For instance, the data coming from the city center area can be monitored along different seasons, or with the heating systems off and on, to be examined by domain experts. *GECKO* and *ARQUATA* engines were validated on real open data collected in a major Italian city (i.e., Milan). The discovered patterns demonstrate their effectiveness in extracting interesting knowledge, which can be easily exploited by public administrators to monitor the air quality in urban environments by means of the automatic reports generated.

In Chapter 3, a novel exploratory data-driven methodology, named *B*ike Station Ov*E*r*L*oad Ana*L*yzer (*BELL*) is presented with the aim of improving the user perception and ease of maintenance of bike-sharing systems. *BELL* analyzes the occupancy level data acquired from real systems to determine situations of dock overload in multiple stations which could lead to service disruption. The proposed methodology relies on a pattern mining approach. In particular, a new pattern type called Occupancy Monitoring Pattern, has been designed to detect situations of dock overload in multiple stations. Since stations are geo-referenced and their occupancy levels are periodically monitored, occupancy patterns can be filtered and evaluated by taking into consideration both the spatial and temporal correlation of the acquired measurements. The results achieved on real data highlight the potential of the proposed methodology in supporting domain experts in their maintenance activities, such as periodic re-balancing of the occupancy levels of the stations, as well as in improving user experience by suggesting alternative stations in the nearby area. During the empirical study, *BELL* has been thoroughly evaluated using a real dataset acquired from the bicycle sharing systems of two important smart cities, i.e., Barcelona and New York. The experimental results demonstrated the effectiveness of *BELL* in identifying useful knowledge regarding the spatio-temporal distribution of possible service disruptions for the end users of bicycle sharing systems.

Chapter 4 presents a study goal of which is to discover recurrent combinations of items characterized by high profit from transactional datasets. A novel type of pattern, namely the *Generalized High-utility Itemset* (*GHUI*), is defined and developed to combine the expressiveness of generalized and High-Utility itemsets. *GHUI* represents a combinations of items at different granularity levels characterized by high profit (utility). According to a user-defined taxonomy, items are first aggregated into semantically related categories. While profitable combinations of item categories provide interesting high-level information, GHUIs at lower abstraction levels represent more specific correlations among profitable items. A single-phase algorithm (i.e., *ML-HUI Miner*) is presented to efficiently discover utility itemsets at multiple abstraction levels. The experiments, which were performed on both real and synthetic data, demonstrate the effectiveness and usefulness of the proposed approach.

Chapter 5 introduces the *TACOMA* system. Starting from a real industry case it aims at supporting the integration of data regarding business activities between different web directories (e.g., Google Maps, Pagine Gialle, Apple Maps) characterized by taxonomies of different granularity levels. In particular, *TACOMA* addresses the problem of a source taxonomy, which has categories at a coarser conceptual level of granularity of the target taxonomy. For instance, a source taxonomy with a concept of furniture whose instances should be mapped to the concepts of chair, table, or bookcase in the target taxonomy. The issue is addressed using a classification approach, where the textual data of each case is leveraged to correctly predict the appropriate category of the target taxonomy. The experiments performed on real data coming from a prominent Italian web directory (i.e., Pagine Gialle) have proven the efficacy of the proposed methodology, which has been already integrated in the production system of the company to export its data towards international partner systems (i.e., Apple Maps, Amazon Alexa).

The data mining frameworks and patterns proposed in this research activity proved, through experimentation, to be effective solutions to get useful knowledge from heterogeneous data in the complex urban and business application domains. The results confirm the importance of leveraging taxonomies to obtain useful insights from the data at multiple levels of granularity. Several possibilities are open to further expand the studies presented in this PhD thesis:

1. the *GECKO* and *ARQUATA* frameworks could be combined and enriched with other kinds of interesting data affecting air quality such as people's mobility and private/public transport data;

2. *BELL* can be expanded integrating other data sources (e.g., environmental pollution, road network features, vehicular traffic, and the presence of cycling lanes) to enrich the quality of the generated model;

3. Future extensions of the *GHUI* patterns could address the efficient extraction of significant subsets of them (e.g., closed or minimal *GHUI*);

4. The *TACOMA* system can be expanded to support a multi-label classification.

Bibliography

[1] Steven S. Aanen, Damir Vandic, and Flavius Frasincar. "Automated product taxonomy mapping in an e-commerce environment". In: *Expert Syst. Appl.* 42.3 (2015), pp. 1298–1313. DOI: 10.1016/j.eswa.2014.09.032. URL: https://doi.org/10.1016/j.eswa.2014.09.032.

[2] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami. "Mining Association Rules between Sets of Items in Large Databases". In: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, May 26-28, 1993.* 1993, pp. 207–216. DOI: 10.1145/170035.170072. URL: https://doi.org/10.1145/170035.170072.

[3] Rakesh Agrawal and Ramakrishnan Srikant. "Fast Algorithms for Mining Association Rules in Large Databases". In: *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile.* 1994, pp. 487–499. URL: http://www.vldb.org/conf/1994/P487.PDF.

[4] Rakesh Agrawal and Ramakrishnan Srikant. "On Integrating Catalogs". In: *Proceedings of the 10th International Conference on World Wide Web.* WWW '01. Hong Kong, Hong Kong: ACM, 2001, pp. 603–612. ISBN: 1-58113-348-0. DOI: 10.1145/371920.372163. URL: http://doi.acm.org/10.1145/371920.372163.

[5] Maura Lodovici et al. "Polycyclic aromatic hydrocarbons air levels in Florence, Italy, and their correlation with other air pollutants". In: *Chemosphere* 50.3 (2003), pp. 377–382. ISSN: 0045-6535. DOI: https://doi.org/10.1016/S0045-6535(02)00404-6. URL: http://www.sciencedirect.com/science/article/pii/S0045653502004046.

[6] *Apache Spark website.* 2019. URL: https://spark.apache.org/docs/latest/ml-frequent-pattern-mining.html (visited on 07/13/2019).

[7] ARPA. Piedmont Region. *Regional Agency for the Protection of the Environment. Available at http://www.arpa.piemonte.it/english-version Last access: December 2014.*

[8] Sasha Babicki, David Arndt, Ana Marcu, Yongjie Liang, Jason R. Grant, Adam Maciejewski, and David S. Wishart. "Heatmapper: web-enabled heat mapping for all". In: *Nucleic Acids Research* 44.Webserver-Issue (2016), W147–W153. DOI: 10.1093/nar/gkw419. URL: https://doi.org/10.1093/nar/gkw419.

[9] Elena Baralis, Luca Cagliero, Tania Cerquitelli, Vincenzo D'Elia, and Paolo Garza. "Support driven opportunistic aggregation for generalized itemset extraction". In: *5th IEEE International Conference on Intelligent Systems, IS 2010, 7-9 July 2010, University of Westminster, London, UK.* 2010, pp. 102–107. DOI: 10.1109/IS.2010.5548348. URL: https://doi.org/10.1109/IS.2010.5548348.

[10] Pierre Borgnat, Patrice Abry, Patrick Flandrin, Céline Robardet, Jean-Baptiste Rouquier, and Eric Fleury. "Shared Bicycles in a City: a Signal Processing and Data Analysis Perspective". In: *Advances in Complex Systems* 14.3 (2011), pp. 415–438. DOI: 10.1142/S0219525911002950. URL: https://doi.org/10.1142/S0219525911002950.

[11] Luca Cagliero. "Discovering Temporal Change Patterns in the Presence of Taxonomies". In: *IEEE Trans. Knowl. Data Eng.* 25.3 (2013), pp. 541–555. DOI: 10.1109/TKDE.2011.233. URL: https://doi.org/10.1109/TKDE.2011.233.

[12] Luca Cagliero, Tania Cerquitelli, Silvia Chiusano, Paolo Garza, and Giuseppe Ricupero. "Discovering air quality patterns in urban environments". In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp Adjunct 2016, Heidelberg, Germany, September 12-16, 2016.* 2016, pp. 25–28. DOI: 10.1145/2968219.2971458. URL: http://doi.acm.org/10.1145/2968219.2971458.

[13] Luca Cagliero, Tania Cerquitelli, Silvia Chiusano, Paolo Garza, Giuseppe Ricupero, and Elena Baralis. "Characterizing Situations of Dock Overload in Bicycle Sharing Stations". In: *Applied Sciences* 8.12 (2018). ISSN: 2076-3417. DOI: 10.3390/app8122521. URL: http://www.mdpi.com/2076-3417/8/12/2521.

[14] Luca Cagliero, Tania Cerquitelli, Silvia Chiusano, Paolo Garza, Giuseppe Ricupero, and Xin Xiao. "Modeling Correlations among Air Pollution-Related Data through Generalized Association Rules". In: *2016 IEEE International Conference on Smart Computing, SMARTCOMP 2016, St Louis, MO, USA, May 18-20, 2016.* 2016, pp. 1–6. DOI: 10.1109/SMARTCOMP.2016.7501707. URL: https://doi.org/10.1109/SMARTCOMP.2016.7501707.

[15] Luca Cagliero, Tania Cerquitelli, Paolo Garza, and Luigi Grimaudo. "Twitter data analysis by means of Strong Flipping Generalized Itemsets". In: *Journal of Systems and Software* 94 (2014), pp. 16–29. DOI: 10.1016/j.jss.2014.03.060. URL: https://doi.org/10.1016/j.jss.2014.03.060.

[16] Luca Cagliero, Silvia Chiusano, Paolo Garza, and Giuseppe Ricupero. "Discovering High-Utility Itemsets at Multiple Abstraction Levels". In: *New Trends in Databases and Information Systems - ADBIS 2017 Short Papers and Workshops, AMSD, BigNovelTI, DAS, SW4CH, DC, Nicosia, Cyprus, September 24-27, 2017, Proceedings.* 2017, pp. 224–234. DOI: 10.1007/978-3-319-67162-8_22. URL: https://doi.org/10.1007/978-3-319-67162-8%5C_22.

[17] Luca Cagliero and Paolo Garza. "Improving classification models with taxonomy information". In: *Data Knowl. Eng.* 86 (2013), pp. 85–101. DOI: 10.1016/j.datak.2013.01.005. URL: http://dx.doi.org/10.1016/j.datak.2013.01.005.

[18] Luca Cagliero and Paolo Garza. "Infrequent Weighted Itemset Mining Using Frequent Pattern Growth". In: *IEEE Trans. Knowl. Data Eng.* 26.4 (2014), pp. 903–915. DOI: 10.1109/TKDE.2013.69. URL: https://doi.org/10.1109/TKDE.2013.69.

[19] Chun Hing Cai, Ada Wai-Chee Fu, Chun Hung Cheng, and Wang Wai Kwong. "Mining Association Rules with Weighted Items". In: *Proceedings of the 1998 International Database Engineering and Applications Symposium, IDEAS 1998, Cardiff, Wales, UK, July 8-10, 1998.* 1998, pp. 68–77. DOI: 10.1109/IDEAS.1998.694360. URL: https://doi.org/10.1109/IDEAS.1998.694360.

[20] Raymond Chan, Qiang Yang, and Yi-Dong Shen. "Mining High Utility Itemsets". In: *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19-22 December 2003, Melbourne, Florida, USA.* 2003, pp. 19–26. DOI: 10.1109/ICDM.2003.1250893. URL: https://doi.org/10.1109/ICDM.2003.1250893.

[21] Vincenzo Ciancia, Diego Latella, Mieke Massink, and Rytis Paskauskas. "Exploring Spatio-temporal Properties of Bike-Sharing Systems". In: *2015 IEEE International Conference on Self-Adaptive and Self-Organizing Systems Workshops, SASO Workshops 2015, Cambridge, MA, USA, September 21-25, 2015.* 2015, pp. 74–79. DOI: 10.1109/SASOW.2015.17. URL: https://doi.org/10.1109/SASOW.2015.17.

[22] Etienne Côme and Latifa Oukhellou. "Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the Vélib' System of Paris". In: *ACM TIST* 5.3 (2014), 39:1–39:21. DOI: 10.1145/2560188. URL: https://doi.org/10.1145/2560188.

[23] Juliana Medeiros Destro, Gabriel Oliveira dos Santos, Júlio Cesar dos Reis, Ricardo da Silva Torres, Ariadne Maria Brito Rizzoni Carvalho, and Ivan Ricarte. "EVOCROS: results for OAEI 2018". In: *Proceedings of the 13th International Workshop on Ontology Matching co-located with the 17th International Semantic Web Conference, OM@ISWC 2018, Monterey, CA, USA, October 8, 2018.* 2018, pp. 152–159. URL: http://ceur-ws.org/Vol-2288/oaei18%5C_paper6.pdf.

[24] Hamdy K. Elminir. "Dependence of urban air pollutants on meteorology". In: *Science of The Total Environment* 350.1-3 (2005), pp. 225–237. ISSN: 0048-9697. DOI: http://dx.doi.org/10.1016/j.scitotenv.2005.01.043.

[25] Alva Erwin, Raj P. Gopalan, and N. R. Achuthan. "CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach". In: *Seventh International Conference on Computer and Information Technology (CIT 2007), October 16-19, 2007, University of Aizu, Fukushima, Japan*. 2007, pp. 71–76. DOI: 10.1109/CIT.2007.120. URL: https://doi.org/10.1109/CIT.2007.120.

[26] Jérôme Euzenat. *Ontology Alignment Evaluation Initiative*. 2019. URL: http://oaei.ontologymatching.org (visited on 06/17/2019).

[27] Sean M. Falconer, Natalya Fridman Noy, and Margaret-Anne D. Storey. "Ontology Mapping - a User Survey". In: *Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) Collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), Busan, Korea, November 11, 2007*. 2007. URL: http://ceur-ws.org/Vol-304/paper5.pdf.

[28] Simone Formentin, Andrea G. Bianchessi, and Sergio M. Savaresi. "On the prediction of future vehicle locations in free-floating car sharing systems". In: *2015 IEEE Intelligent Vehicles Symposium, IV 2015, Seoul, South Korea, June 28 - July 1, 2015*. 2015, pp. 1006–1011. DOI: 10.1109/IVS.2015.7225816. URL: https://doi.org/10.1109/IVS.2015.7225816.

[29] Philippe Fournier-Viger, Antonio Gomariz, Ted Gueniche, Azadeh Soltani, Cheng-Wei Wu, and Vincent S. Tseng. "SPMF: a Java open-source pattern mining library". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 3389–3393. URL: http://dl.acm.org/citation.cfm?id=2750353.

[30] Philippe Fournier-Viger, Jerry Chun-Wei Lin, Cheng-Wei Wu, Vincent S. Tseng, and Usef Faghihi. "Mining Minimal High-Utility Itemsets". In: *Database and Expert Systems Applications - 27th International Conference, DEXA 2016, Porto, Portugal, September 5-8, 2016, Proceedings, Part I*. 2016, pp. 88–101. DOI: 10.1007/978-3-319-44403-1_6. URL: http://dx.doi.org/10.1007/978-3-319-44403-1_6.

[31] Philippe Fournier-Viger, Souleymane Zida, Jerry Chun-Wei Lin, Cheng-Wei Wu, and Vincent S. Tseng. "Efficient closed high-utility itemset mining". In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, April 4-8, 2016*. 2016, pp. 898–900. DOI: 10.1145/2851613.2851884. URL: http://doi.acm.org/10.1145/2851613.2851884.

[32] Philippe Fournier-Viger, Souleymane Zida, Jerry Chun-Wei Lin, Cheng-Wei Wu, and Vincent S. Tseng. "Efficient closed high-utility itemset mining". In: *Proceedings of the 31st Annual ACM Symposium on Applied Computing, Pisa, Italy, April 4-8, 2016.* 2016, pp. 898–900. DOI: 10.1145/2851613.2851884. URL: http://doi.acm.org/10.1145/2851613.2851884.

[33] Christine Fricker and Nicolas Gast. "Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity". In: *EURO J. Transportation and Logistics* 5.3 (2016), pp. 261–291. DOI: 10.1007/s13676-014-0053-5. URL: https://doi.org/10.1007/s13676-014-0053-5.

[34] Jon Froehlich, Joachim Neumann, and Nuria Oliver. "Measuring the Pulse of the City through Shared Bicycle Programs". In: *Proceedings of the International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems (UrbanSense08).* 2008.

[35] Wensheng Gan, Jerry Chun-Wei Lin, Philippe Fournier-Viger, and Han-Chieh Chao. "More Efficient Algorithms for Mining High-Utility Itemsets with Multiple Minimum Utility Thresholds". In: *Database and Expert Systems Applications - 27th International Conference, DEXA 2016, Porto, Portugal, September 5-8, 2016, Proceedings, Part I.* 2016, pp. 71–87. DOI: 10.1007/978-3-319-44403-1_5. URL: http://dx.doi.org/10.1007/978-3-319-44403-1_5.

[36] Fabien Girardin, Francesco Calabrese, Filippo Dal Fiore, Carlo Ratti, and Josep Blat. "Digital Footprinting: Uncovering Tourists with User-Generated Content". In: *IEEE Pervasive Computing* 7.4 (2008), pp. 36–43. DOI: 10.1109/MPRV.2008.71. URL: https://doi.org/10.1109/MPRV.2008.71.

[37] Rebecca Gleason and Laurie Miskimins. *Options for Federal Lands: Bike Sharing, Rentals and Employee Fleets.* Tech. rep. Western Transportation Institute, 2012. URL: www.nps.gov/transportation/pdfs/FHWA_bicycle_options.pdf.

[38] Jiawei Han and Yongjian Fu. "Discovery of Multiple-Level Association Rules from Large Databases". In: *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland.* 1995, pp. 420–431. URL: http://www.vldb.org/conf/1995/P420.PDF.

[39] Jiawei Han, Jian Pei, and Yiwen Yin. "Mining Frequent Patterns without Candidate Generation". In: *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, May 16-18, 2000, Dallas, Texas, USA.* 2000, pp. 1–12. DOI: 10.1145/342009.335372. URL: https://doi.org/10.1145/342009.335372.

[40] Samiul Hasan, Xianyuan Zhan, and Satish V. Ukkusuri. "Understanding urban human activity and mobility patterns using large-scale location-based data from online social media". In: *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing, UrbComp@KDD 2013, Chicago, Illinois, USA, August 11, 2013*. 2013, 6:1–6:8. DOI: 10.1145/2505821.2505823. URL: https://doi.org/10.1145/2505821.2505823.

[41] G. Henri ter Hofte, Kasper Løvborg Jensen, Petteri Nurmi, and Jon Froehlich. "Mobile Living Labs 09: Methods and Tools for Evaluation in the Wild: http://mll09.novay.nl". In: *Proceedings of the 11th Conference on Human-Computer Interaction with Mobile Devices and Services, Mobile HCI 2009, Bonn, Germany, September 15-18, 2009*. 2009. DOI: 10.1145/1613858.1613981. URL: https://doi.org/10.1145/1613858.1613981.

[42] Andreas Kaltenbrunner, Rodrigo Meza, Jens Grivolla, Joan Codina, and Rafael E. Banchs. "Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system". In: *Pervasive and Mobile Computing* 6.4 (2010), pp. 455–466. DOI: 10.1016/j.pmcj.2010.07.002. URL: https://doi.org/10.1016/j.pmcj.2010.07.002.

[43] Mayank Kejriwal and Daniel P. Miranker. "Semi-supervised Instance Matching Using Boosted Classifiers". In: *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings*. 2015, pp. 388–402. DOI: 10.1007/978-3-319-18818-8_24. URL: https://doi.org/10.1007/978-3-319-18818-8%5C_24.

[44] Konstantinos Kotis, George A. Vouros, and Konstantinos Stergiou. "Towards automatic merging of domain ontologies: The HCONE-merge approach". In: *J. Web Semant.* 4.1 (2006), pp. 60–79. DOI: 10.1016/j.websem.2005.09.004. URL: https://doi.org/10.1016/j.websem.2005.09.004.

[45] Srikumar Krishnamoorthy. "Pruning strategies for mining high utility itemsets". In: *Expert Syst. Appl.* 42.5 (2015), pp. 2371–2381. DOI: 10.1016/j.eswa.2014.11.001. URL: https://doi.org/10.1016/j.eswa.2014.11.001.

[46] Yexin Li, Yu Zheng, and Qiang Yang. "Dynamic Bike Reposition: A Spatio-Temporal Reinforcement Learning Approach". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. 2018, pp. 1724–1733. DOI: 10.1145/3219819.3220110. URL: https://doi.org/10.1145/3219819.3220110.

[47] Jerry Chun-Wei Lin, Philippe Fournier-Viger, and Wensheng Gan. "FHN: An efficient algorithm for mining high-utility itemsets with negative unit profits". In: *Knowl.-Based Syst.* 111 (2016), pp. 283–298. DOI: 10.1016/j.knosys.2016.08.022. URL: http://dx.doi.org/10.1016/j.knosys.2016.08.022.

[48] Jerry Chun-Wei Lin, Philippe Fournier-Viger, and Wensheng Gan. "FHN: An efficient algorithm for mining high-utility itemsets with negative unit profits". In: *Knowl.-Based Syst.* 111 (2016), pp. 283–298. DOI: 10.1016/j.knosys.2016.08.022. URL: https://doi.org/10.1016/j.knosys.2016.08.022.

[49] Junming Liu, Leilei Sun, Weiwei Chen, and Hui Xiong. "Rebalancing Bike Sharing Systems: A Multi-source Data Smart Optimization". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016.* 2016, pp. 1005–1014. DOI: 10.1145/2939672.2939776. URL: https://doi.org/10.1145/2939672.2939776.

[50] Junqiang Liu, Ke Wang, and Benjamin C. M. Fung. "Direct Discovery of High Utility Itemsets without Candidate Generation". In: *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012.* 2012, pp. 984–989. DOI: 10.1109/ICDM.2012.20. URL: https://doi.org/10.1109/ICDM.2012.20.

[51] Ying Liu, Wei-keng Liao, and Alok N. Choudhary. "A Two-Phase Algorithm for Fast Discovery of High Utility Itemsets". In: *Advances in Knowledge Discovery and Data Mining, 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005, Proceedings.* 2005, pp. 689–695. DOI: 10.1007/11430919_79. URL: https://doi.org/10.1007/11430919%5C_79.

[52] Alvaro Lozano, Juan F. De Paz, Gabriel Villarrubia Gonzalez, Daniel H. De La Iglesia, and Javier Bajo. "Multi-Agent System for Demand Prediction and Trip Visualization in Bike Sharing Systems". In: *Applied Sciences* 8.1 (2018). ISSN: 2076-3417. DOI: 10.3390/app8010067. URL: http://www.mdpi.com/2076-3417/8/1/67.

[53] Elliot Martin, Nelson Chan, Adam Cohen, and Mike Pogodzinski. *Public Bike sharing in North America During A Period of Rapid Expansion: Understanding Business Models, Industry Trends and User Impacts.* Tech. rep. Mineta Transportation Institute, 2014. URL: http://transweb.sjsu.edu/project/1131.html.

[54] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. "Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching". In: *Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, USA, February 26 - March 1, 2002.* 2002, pp. 117–128. DOI: 10.1109/ICDE.2002.994702. URL: https://doi.org/10.1109/ICDE.2002.994702.

[55] Eduardo Mena, Arantza Illarramendi, Vipul Kashyap, and Amit P. Sheth. "OBSERVER: An Approach for Query Processing in Global Information Systems Based on Interoperation Across Pre-Existing Ontologies". In: *Distributed and Parallel Databases* 8.2 (2000), pp. 223–271. DOI: 10.1023/A:1008741824956. URL: https://doi.org/10.1023/A:1008741824956.

[56] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. "YALE: Rapid Prototyping for Complex Data Mining Tasks". In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '06. Philadelphia, PA, USA: ACM, 2006, pp. 935–940. ISBN: 1-59593-339-5. DOI: 10.1145/1150402.1150531. URL: http://doi.acm.org/10.1145/1150402.1150531.

[57] Rahul Nair, Elise Miller-Hooks, Robert C. Hampshire, and Ana Bušić. "Large-Scale Vehicle Sharing Systems: Analysis of Vélib'". In: *International Journal of Sustainable Transportation* 7.1 (2013), pp. 85–106. DOI: 10.1080/15568318.2012.660115.

[58] Borecki Natalie, Darren Buck, Payton Chung, Patricia Happ, Nicholas Kushner, Tim Maher, Bradley Rawls, Paola Reyes, Matthew Steenhoek, Casey Studhalter, Austin Watkins, and Ralph Buehler. *Virginia Tech Capital Bikeshare Study*. Tech. rep. Virginia Tech, 2012. URL: https://ralphbu.files.wordpress.com/2012/01/vt-bike-share-study-final3.pdf.

[59] Ikechukwu Nkisi-Orji, Nirmalie Wiratunga, Stewart Massie, Kit-Ying Hui, and Rachel Heaven. "Ontology Alignment Based on Word Embedding and Random Forest Classification". In: *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2018, Dublin, Ireland, September 10-14, 2018, Proceedings, Part I*. 2018, pp. 557–572. DOI: 10.1007/978-3-030-10925-7_34. URL: https://doi.org/10.1007/978-3-030-10925-7%5C_34.

[60] Natalya Fridman Noy and Mark A. Musen. "Using Prompt Ontology-Comparison Tools in the EON Ontology Alignment Contest". In: *EON 2004, Evaluation of Ontology-based Tools, Proceedings of the 3rd International Workshop on Evaluation of Ontology-based Tools held at the 3rd International Semantic Web Conference ISWC 2004, 7th November 2004, Hiroshima Prince Hotel, Hiroshima, Japan*. 2004. URL: http://ceur-ws.org/Vol-128/EON2004%5C_EXP%5C_Noy.pdf.

[61] Eoin O'Mahony and David B. Shmoys. "Data Analysis and Optimization for (Citi)Bike Sharing". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA*. 2015, pp. 687–694. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9698.

[62] Oliver O'Brien, James Cheshire, and Michael Batty. "Mining bicycle sharing data for generating insights into sustainable transport systems". In: *Journal of Transport Geography* 34 (2014), pp. 262–273. ISSN: 0966-6923. DOI: https://doi.org/10.1016/j.jtrangeo.2013.06.007.

[63] Sangun Park and Wooju Kim. "Ontology Mapping Between Heterogeneous Product Taxonomies in an Electronic Commerce Environment". In: *Int. J. Electronic Commerce* 12.2 (2007), pp. 69–87. DOI: 10.2753/JEC1086-4415120203. URL: https://doi.org/10.2753/JEC1086-4415120203.

[64] *RapidMiner. Last access: February 2016.* Online. 2016. URL: http://rapid-i.com/content/view/181/190/.

[65] Tal Raviv, Michal Tzur, and Iris A. Forma. "Static repositioning in a bike-sharing system: models and solution approaches". In: *EURO J. Transportation and Logistics* 2.3 (2013), pp. 187–229. DOI: 10.1007/s13676-012-0017-6. URL: https://doi.org/10.1007/s13676-012-0017-6.

[66] Advait Sarkar, Neal Lathia, and Cecilia Mascolo. "Comparing cities' cycling patterns using online shared bicycle maps". In: *Transportation* 42.4 (2015), pp. 541–559. ISSN: 0049-4488.

[67] Bernhard Schölkopf and Alexander Johannes Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond.* Adaptive computation and machine learning series. MIT Press, 2002. ISBN: 9780262194754. URL: http://www.worldcat.org/oclc/48970254.

[68] J. Schuijbroek, Robert C. Hampshire, and W.-J. van Hoeve. "Inventory rebalancing and vehicle routing in bike sharing systems". In: *European Journal of Operational Research* 257.3 (2017), pp. 992–1004. DOI: 10.1016/j.ejor.2016.08.029. URL: https://doi.org/10.1016/j.ejor.2016.08.029.

[69] Susan Shaheen, Stacey Guzman, and Hua Zhang. "Bikesharing in Europe, the Americas, and Asia: Past, Present, and Future". In: *UC Davis: Institute of Transportation Studies (UCD)* (2010), pp. 8–15. URL: https://escholarship.org/uc/item/79v822k5.

[70] Susan Shaheen and Elliot Martin. "Unraveling the modal impacts of Bikesharing". In: *Access Magazine* (2009), pp. 8–15. URL: www.accessmagazine.org/wp-content/uploads/sites/7/2015/12/access47.shaheen.pdf.

[71] Yi-Dong Shen, Zhong Zhang, and Qiang Yang. "Objective-Oriented Utility-Based Association Mining". In: *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), 9-12 December 2002, Maebashi City, Japan.* 2002, pp. 426–433. DOI: 10.1109/ICDM.2002.1183938. URL: https://doi.org/10.1109/ICDM.2002.1183938.

[72] Pavel Shvaiko and Jérôme Euzenat. *Ontology Matching Initiative.* 2019. URL: http://www.ontologymatching.org (visited on 06/14/2019).

[73] Adish Singla, Marco Santoni, Gábor Bartók, Pratik Mukerji, Moritz Meenen, and Andreas Krause. "Incentivizing Users for Balancing Bike Sharing Systems". In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.* 2015, pp. 723–729. URL: http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9942.

[74] Ramakrishnan Srikant and Rakesh Agrawal. "Mining Generalized Association Rules". In: *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland.* 1995, pp. 407–419. URL: http://www.vldb.org/conf/1995/P407.PDF.

[75] M Statheropoulos, N Vassiliadis, and A Pappa. "Principal component and canonical correlation analysis for examining air pollution and meteorological data". In: *Atmospheric Environment* 32.6 (1998), pp. 1087–1095. ISSN: 1352-2310.

[76] Shaheen Susan, Elliot Martin, and Adam Cohen. "Bikesharing and Modal Shift Behavior: A Comparative Study of Early Bikesharing Systems in North America". In: *International Journal of Sustainable Transportation* 1.1 (2013), pp. 35–54. DOI: 10.14257/ijt.2013.1.1.03.

[77] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining.* Addison-Wesley, 2005. ISBN: 0-321-32136-7.

[78] Andreas Thor, Toralf Kirsten, and Erhard Rahm. "Instance-based matching of hierarchical ontologies". In: *Datenbanksysteme in Business, Technologie und Web (BTW 2007), 12. Fachtagung des GI-Fachbereichs "Datenbanken und Informationssysteme" (DBIS), Proceedings, 7.-9. März 2007, Aachen, Germany.* 2007, pp. 436–448. URL: http://subs.emis.de/LNI/Proceedings/Proceedings103/article1413.html.

[79] Vincent S. Tseng, Bai-En Shie, Cheng-Wei Wu, and Philip S. Yu. "Efficient Algorithms for Mining High Utility Itemsets from Transactional Databases". In: *IEEE Trans. Knowl. Data Eng.* 25.8 (2013), pp. 1772–1786. DOI: 10.1109/TKDE.2012.59. URL: https://doi.org/10.1109/TKDE.2012.59.

[80] Vincent S. Tseng, Cheng-Wei Wu, Philippe Fournier-Viger, and Philip S. Yu. "Efficient Algorithms for Mining Top-K High Utility Itemsets". In: *IEEE Trans. Knowl. Data Eng.* 28.1 (2016), pp. 54–67. DOI: 10.1109/TKDE.2015.2458860. URL: https://doi.org/10.1109/TKDE.2015.2458860.

[81] Vincent S. Tseng, Cheng-Wei Wu, Bai-En Shie, and Philip S. Yu. "UP-Growth: an efficient algorithm for high utility itemset mining". In: *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010.* 2010, pp. 253–262. DOI: 10.1145/1835804.1835839. URL: https://doi.org/10.1145/1835804.1835839.

[82] Patrick Vogel, Torsten Greiser, and Dirk Christian Mattfeld. "Understanding Bike-Sharing Systems using Data Mining: Exploring Activity Patterns". In: *Procedia - Social and Behavioral Sciences* 20 (2011), pp. 514–523. ISSN: 1877-0428.

[83] Patrick Vogel and Dirk C. Mattfeld. "Strategic and Operational Planning of Bike-Sharing Systems by Data Mining - A Case Study". In: *Computational Logistics - Second International Conference, ICCL 2011, Hamburg, Germany, September 19-22, 2011. Proceedings*. 2011, pp. 127–141. DOI: 10.1007/978-3-642-24264-9_10. URL: https://doi.org/10.1007/978-3-642-24264-9%5C_10.

[84] I-Lin Wang and Chun-Wei Wang. "Analyzing Bike Repositioning Strategies Based on Simulations for Public Bike Sharing Systems: Simulating Bike Repositioning Strategies for Bike Sharing Systems". In: *2013 Second IIAI International Conference on Advanced Applied Informatics, IIAI-AAI 2013, Matsue, Japan, August 31 - Sept. 4, 2013*. 2013, pp. 306–311. DOI: 10.1109/IIAI-AAI.2013.9. URL: https://doi.org/10.1109/IIAI-AAI.2013.9.

[85] Shang Wang, Jiangman Zhang, Liang Liu, and Zheng-yu Duan. "Bike-Sharing-A new public transportation mode: State of the practice and prospects". In: *Emergency Management and Management Sciences (ICEMMS), 2010 IEEE International Conference on*. Aug. 2010, pp. 222–225.

[86] *Wikipedia Meteo information about metereological data*. URLs: https://en.wikipedia.org/wiki/Rain, https://en.wikipedia.org/wiki/Wind, https://en.wikipedia.org/wiki/Ultravioletindex, https://en.wikipedia.org/wiki/Atmosphericpressure. (Visited on 02/08/2016).

[87] Yu Zheng, Furui Liu, and Hsun-Ping Hsieh. "U-Air: when urban air quality inference meets big data". In: *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*. 2013, pp. 1436–1444. DOI: 10.1145/2487575.2488188. URL: https://doi.org/10.1145/2487575.2488188.

[88] Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. "Forecasting Fine-Grained Air Quality Based on Big Data". In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, NSW, Australia, August 10-13, 2015*. 2015, pp. 2267–2276. DOI: 10.1145/2783258.2788573. URL: https://doi.org/10.1145/2783258.2788573.

[89] Souleymane Zida, Philippe Fournier-Viger, Jerry Chun-Wei Lin, Cheng-Wei Wu, and Vincent S. Tseng. "EFIM: A Highly Efficient Algorithm for High-Utility Itemset Mining". In: *Advances in Artificial Intelligence and Soft Computing - 14th Mexican International Conference on Artificial Intelligence, MICAI 2015, Cuernavaca, Morelos, Mexico, October 25-31, 2015, Proceedings, Part I*. 2015, pp. 530–546. DOI: 10.1007/978-3-319-27060-9_44. URL: https://doi.org/10.1007/978-3-319-27060-9%5C_44.

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.