Politecnico di Torino

COMPUTER AND CONTROL ENGINEERING
XXVIII CYCLE

PhD Thesis

Data Mining Techniques for Complex User-Generated Data

Supervisor:
Prof. Silvia Chiusano

Author:
Xin Xiao
Matr. 199914

May 2016

Politecnico di Torino

Abstract

Computer and Control Engineering
XXVIII Cycle

PhD

Data Mining Techniques for Complex User-Generated Data

by Xin Xiao
Matr. 199914

Nowadays, the amount of collected information is continuously growing in a variety of different domains. Data mining techniques are powerful instruments to effectively analyze these large data collections and extract hidden and useful knowledge.

Vast amount of User-Generated Data (UGD) is being created every day, such as user behavior, user-generated content, user exploitation of available services and user mobility in different domains. Some common critical issues arise for the UGD analysis process such as the large dataset cardinality and dimensionality, the variable data distribution and inherent sparseness, and the heterogeneous data to model the different facets of the targeted domain. Consequently, the extraction of useful knowledge from such data collections is a challenging task, and proper data mining solutions should be devised for the problem under analysis.

In this thesis work, we focus on the design and development of innovative solutions to support data mining activities over User-Generated Data characterised by different critical issues, via the integration of different data mining techniques in a unified framework. Real datasets coming from three example domains characterized by the above critical issues are considered as reference cases, i.e., health care, social network, and urban environment domains. Experimental results show the effectiveness of the proposed approaches to discover useful knowledge from different domains.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Silvia Chiusano, for her important suggestions and encouragement throughout my PhD course, and for all the opportunities to participate in the research activities in data mining. I'm extremely grateful for the great effort and support she put into training me. Her excellent guidance is extremely precious for my PhD study as well as future career, I hope I could be as enthusiastic and energetic as her in my work.

I would also like to express my great appreciation to Tania Cerquitelli, for her valuable suggestions and support in most of my research work, as well as offering me the opportunity of teaching activity.

In addition, I would like to thank all my research group colleagues, Prof. Elena Baralis, Luca Cagliero, Paolo Garza, and Giulia Bruno for their support and suggestions at times, especially Paolo who gave me advices during some experiments. Thank also the other group members, it's my pleasure to work with these nice people. I would also like to thank all the people I met during my PhD study.

I especially thank my parents and all the other family members. They were always supporting me and encouraging me with their best wishes, in the distant home.

Finally, a very special thank to Tao Su, my future husband, for his love, company, care and encouragement throughout all these years abroad.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Nowadays, the rapid development of online systems or devices have given rise to a huge amount of electronic data. The inexpensive effort for data capturing and storage has enabled huge data generation from various industries and innovations, such as banking, healthcare, social network, e-commerce and urban context. The need to extract useful knowledge from these huge, complex, information-rich data collections is becoming more and more important in the real world.

Data mining techniques are powerful instruments that can be effectively used to analyze these large data collections and extract hidden and useful knowledge. Data mining techniques have been successfully applied in various application domains as telecommunications, healthcare, and web. They allow extracting previously unknown interesting patterns such as discovering groups of similar data objects (cluster analysis), mining correlations among data objects (association analysis), building a model describing data classes and assigning a class label to a new unlabeled data object (classification), or predict the future values for continous data (regression).

The research activity carried out in the PhD is on using data mining techniques for data analysis on complex application domains. For these domains, some common critical issues arise for the data analysis process. Data collections can be characterized by a *large cardinality* and *dimensionality*, a *variable data distribution* and *inherent sparseness*. In addition, to model the different facets of the targeted domain *heterogeneous data* types should be considered in the data analysis process. *Historical data* should also be considered to properly characterize the problem under analysis. Consequently, the extraction of useful knowledge from such data collections is a challenging task, and proper data mining solutions should be devised.

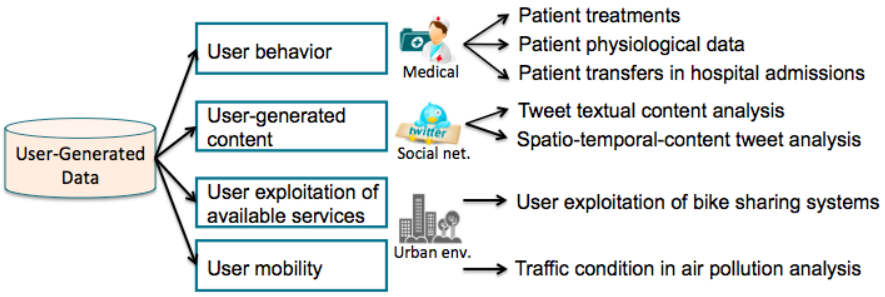


FIGURE 1.1: User-Generated Data in the research work

In the PhD research activity, three application domains characterized by the above critical issues are considered as reference example case. These domains are the health care domain, the social network domain, and the urban environment.

Particularly relevant for the knowledge extraction process in these domains is the analysis of User-Generated data (UGD). The vast amount of UGD created every day is a significant challenge for data analysis due to the various data types and their common critical issues. The UGD considered in this study for the three reference domains includes data on user behavior, on user exploitation of available services, on user mobility and on user-generated content (see Figure 1.1). For example, in the health care domain, data on user behavior refer to the patient history on the underwent treatments. These data can be analysed to extract a variety of information on the medical treatments currently adopted and gain insights for improving medical guidelines for a give disease. In the social network domain, User-Generated Data refer to the content in tweet messages, possibly also including spatio-temporal information on "when" and "where" the tweet has been posted. These data can be analysed to extract various knowledge such as user activities or topics of interest, personal options and user emotions.

This thesis work focuses on design and development of innovative solutions to analyze User-Generated Data (UGD) in complex application domains. Since a single data mining technique may not fit the heterogeneous characteristics of the data under analysis, the research activity addressed the integration of different data mining techniques in a unified framework, as the jointly exploitation of clustering and classification techniques and clustering and association analysis. Real datasets are considered to assess the proposed approaches. More specifically, data mining techniques have been studied and developed in the PhD research to address the following issues.

Analysis of sparse, high-dimensional data with variable distribution. Real-world data collections are usually characterized by an inherent sparseness and variable

distribution, since they are generated by a large variety of events, and high data dimensionality because features used to model real objects and human actions may have very large domains. The variability in data distribution grows with data volume, thus increasing the complexity of mining such data. In the research activity, some reference examples of sparse, high-dimensional data coming from the health care and social network domains have been considered. For example in health care domain, because of the variety of medical treatments usually adopted for the different degrees of severity of a given pathology, patient data collections are usually characterized by inherent sparseness, high dimensionality and variable distribution. To deal with these issues, in the research activity a unified framework named Multiplel-Level Data Analysis (MLDA) framework has been proposed, by jointly exploiting multiple-level clustering, association, and classification analysis.

Analysis of heterogeneous data with large cardinality. Heterogeneous data is a common characteristic of datasets in various domains to model data under different facets. Failing to take the heterogeneous issue into account can easily derail the discoveries from these data. In the research activity we considered some reference examples of heterogeneous data coming from the health care, social network and urban environment domains. For example, when analyzing patient treatments in the health care domain, despite patient examinations, patient profile information such as age and gender can be also taken into account. Innovative data analytics solutions able to acquire, integrate and analyze data containing large amount of heterogeneous dimensions are needed. To address the above issues, in the research activity, novel combined distance measures taking into account all considered facets of the problem under analysis have been proposed and integrated into the clustering process. When aimed at discovering interesting correlations in the heterogeneous UGD, data taxonomy integrated with association analysis has been also presented to discover correlations among heterogeneous data at different abstraction levels.

Analysis of historical data. Historical data is data collected in past-periods, used usually as a basis for forecasting the future data values or trends. Historical data is often represented as time series records, which has been useful in helping predict the future of a company and a market through predictive analyses. In the research work we considered some reference examples of historical data coming from the health care and urban environment domains. In health care domain, an example of historical UGD is the collection of physiological signal values describing the cardiac and respiratory response of patients during a cardiopulmonary exercise test, where the physiological

signal values are multivariate time series. Since the test is physically very demanding, innovative data analysis techniques are needed to predict patient response thus lowering body stress and avoiding cardiopulmonary overload. To deal with these issues, in the research activity, a framework has been proposed to predict future values based on historical UGD collected in a time window, by jointly exploiting windowing approach and classification or regression techniques.

This thesis is organized as follows. The analysis of sparse, high-dimensional data with variable distribution is described in Chapter 2. Chapter 3 presents the analysis of heterogeneous data with large cardinality. Chapter 4 describes the extraction of useful knowledge from historical data. Finally, Chapter 5 presents conclusions and discusses future developments for the proposed approaches.

Chapter 2

Analysis of sparse, high-dimensional data

This chapter describes data mining algorithms designed and developed in this PhD thesis to analyze sparse, high-dimensional User-Generated Data. Real-world data collections are usually characterized by an *inherent sparseness* and *variable distribution*, since they are generated by a large variety of events, and *high data dimensionality* because features used to model real objects and human actions may have very large domains. The variability in data distribution grows with data volume, thus increasing the complexity of mining such data. However, at present, most single data mining algorithms perform better with uniform data distribution, while their performance as well as the quality of the extracted knowledge tend to decrease in non-uniform collections. Consequently, the extraction of useful knowledge from such data collections is a challenging task. It's necessary to jointly exploit data mining techniques and proper data mining solutions should be devised for the problem under analysis.

In the research activity carried out during the PhD study, some reference examples characterized by these issues have been considered coming from the health-care domain and social network domain (see Figure 2.1). In the health care domain, data on treatments

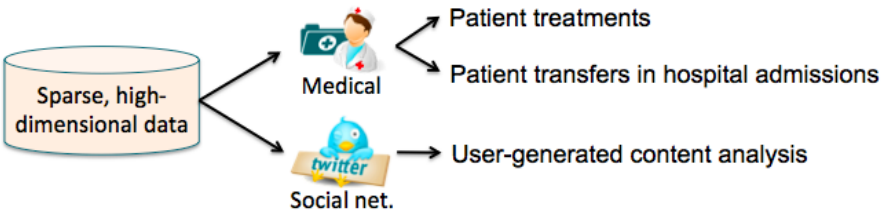


FIGURE 2.1: Considered sparse, high-dimensional data analysis

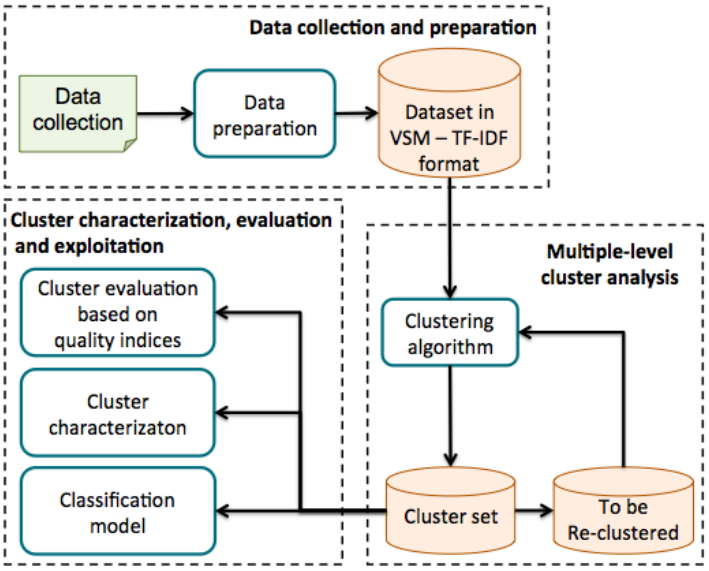


FIGURE 2.2: Multiplel-Level Data Analysis framework

underwent by patients and on patient transfers in hospital admissions have been considered. Health care data collections can have large volume due to the large cardinality of patient records. Because of the variety of medical treatments usually adopted for the different degrees of severity of a given pathology, patient data collections are also usually characterized by high dimensionality, variable data distribution and inherent sparseness. Because of the various wards among which patients may transfer in hospital admission, during the analysis of patient transfers the issues also exist. In social network domain, User Generated Content (UGC) from Twitter messages is also characterised by inherent sparseness, due to the limitation of 140 characters in the message.

Aimed at addressing the above issues, in the research activity, a unified data analysis framework named Multiplel-Level Data Analysis (MLDA) framework, has been proposed, by jointly exploiting multiple-level clustering, association, and classification analysis. The main architecture blocks, depicted in Figure 2.2, are (i) *Data collection and preparation*, the considered data collection is first prepared and represented in the Vector Space Model (VSM) [1] using the Term Frequency - Inverse Document Frequency (TF-IDF) method [2]. The VSM representation has been applied in previous works [1] to represent text documents, while the TF-IDF scheme has been used to weight the relevance of words appearing in the document. (ii) *Multiple-level cluster analysis*, where clustering algorithms are exploited in a multiple-level fashion to iteratively focus on different dataset portions and *locally* identify groups of correlated objects. It allows discovering cohesive and well-separated clusters with divers data distributions. (iii) *Cluster characterization, evaluation and exploitation*, where the quality of discovered clusters is evaluated based on various indices such as Sum of Squared Error (SSE), Silhouette, and

Overall Similarity (see Section 2.1.4). The cluster content is also concisely characterized through association analysis capturing correlations among data features. Moreover, for supporting the automatic categorization of a new data object into one of the discovered cluster, a classification model can be created starting from the computed cluster set.

In this thesis, the complete instance of the MLDA framework has been exploited to analyse diabetic patient treatments as a reference case study in Section 2.1. Sections 2.2 and 2.3 describe the application of preliminary instances of the MLDA framework on User-Generated Content in social network and on patient transfers in health-care domains, respectively.

2.1 Analysis of patient treatments

This section describes the exploitation of a complete instance of the MLDA framework to analyze treatments of diabetic patients. A real dataset including the examination log data of (anonymized) patients with overt diabetes has been considered as a reference case study. Diabetes describes a group of metabolic diseases in which the patient has high blood glucose. Diabetic patients may suffer by various disease complications as eye problems, neuropathy, kidney and cardiovascular diseases. Patients affected by disease complications (or at risk of them) should be tested with more specific examinations in addition to routine tests to monitor its status (or reveal the pathology). The work presented in this section has been published in [3].

The components of the MLDA framework for patient treatments analysis are depicted in Figure 2.3. In the *data preparation* phase the collection of diabetic patients' exam log data is tailored to the Vector Space Model (VSM) representation [1] using the TF-IDF method [2], with the aim of highlighting the relevance of specific data characteristics.

Then, in the *data clustering* phase prepared data are clustered using a multiple level clustering strategy. In this study, five different multiple-level clustering algorithms have been integrated into MLDA, based on K-means (i.e., bisecting and refined K-means [4]), K-medoids (i.e., bisecting and refined K-medoids [5]), and DBSCAN methods (i.e., multiple-level DBSCAN [6]). Clustering results have been then analyzed and compared using some well-established quality indices, as SSE, Silhouette and overall similarity, and Rand Index [2].

Finally, in the *cluster characterization, evaluation and exploitation* phase, the cluster content has been analysed with the aim of providing useful information to the final end-user. For cluster characterization, maximal sequential patterns [7] have been selected to concisely describe temporal correlations among data features appearing in each cluster.

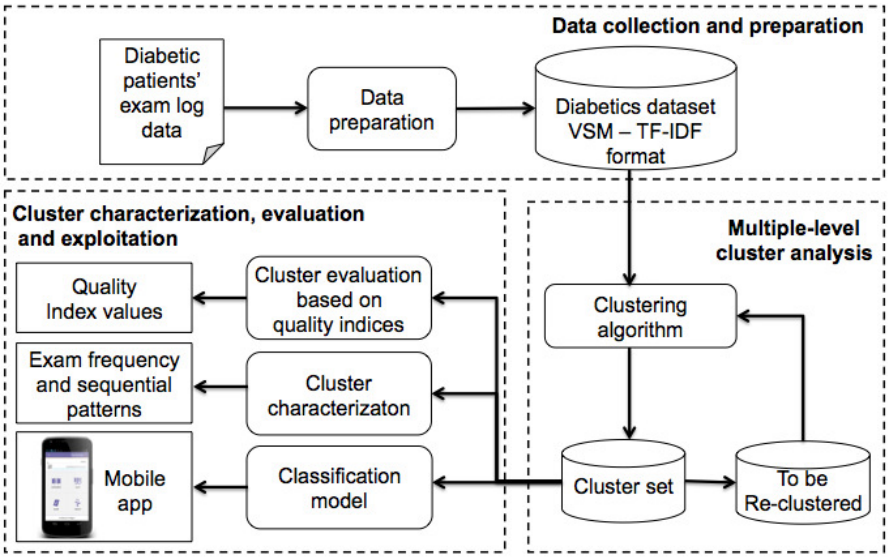


FIGURE 2.3: The MLDA framework on treatments of diabetic patients

A classification model has been built using decision trees [2], which have been shown to provide accurate models in various applications. To allow ubiquitous classification on new data, real-time analysis is executed on mobile devices through an ad-hoc Android application exploiting the classification model.

2.1.1 Related work

In health-care domain, many studies addressed the identification of correlated groups of patients affected by different diseases. For example, [8] reviewed the cluster methods used to diagnose heart valve diseases. In [9], clustering techniques were used to diagnose breast cancer based on tutor features, by recognising hidden patterns of benign and malignant tumors. Authors in [10] exploited the K-means algorithm to cluster a collection of patient records aimed at identifying relevant features of patients subjected to heart attack.

Some research efforts have been devoted to exploiting clustering techniques on data related to diabetic patients [11]. Different issues have been addressed as food analysis [12], gait patterns [13], discovering relationships among diabetes and risk factors [14], analyses of various imputation techniques [15], and discovering similar medical treatments [6]. [15] focuses on diabetes datasets using the K-means algorithm aimed at analysing various imputation techniques. Different from [15], in the framework we aim at identifying groups of patients with similar examination histories to provide a preliminar patient categorization into a set of predefined classes. Thus, we detailed each cluster

with sequential patterns to discover how examinations are interleaved and distributed over time.

The wide diffusion of mobile technologies and the increasing capabilities of mobile computing devices caused an increased interest in designing, implementing and testing innovative applications running on mobile devices to provide a wide range of useful services based on user-generated data. In the medical care scenario, some efforts [16, 17, 18] have been devoted on this appealing research. In [16], a distributed end-to-end pervasive healthcare system utilizing neural network computations for diagnosing diabetes was developed in small mobile devices. [17] developed a new mobile-based approach to automatically detect seizures, using k-means as unsupervised classification technique. [18] have presented Generalized Discriminant Analysis and Least Square Support Vector Machine models to diagnose the diabetes disease. Also in this study, we integrated in the framework a two-tier architecture to allows ubiquitous patient classification through a mobile application. The proposed solution allows to efficiently and effectively exploiting knowledge items discovered through multiple-level cluster analysis to different user profiles (e.g., medical staff, patients). Thus, the proposed mobile application allows ubiquitous patient classification on new unlabelled examination histories.

2.1.2 Data preparation

The considered data collection is characterized by an inherently sparse distribution due to the variety of possible examinations, covering both routine tests and more specific examinations for different degrees of severity in diabetes. In the considered collection of patient records, each record corresponds to a medical examination done by a patient in a given date. For instance, Table 2.1 shows a toy example dataset listing the medical examinations undergone by two patients p_1 and p_2 in year 2014. A more formal definition of a collection of patient records is given in Definition 2.1.1.

TABLE 2.1: Example of a collection of patient records

PatientID	Examination	Date	PatientID	Examination	Date
p_1	Glucose level	2014-02-10	p_2	Urine test	2014-12-01
p_2	Fundus oculi	2014-01-06	p_2	Triglycerides	2014-11-30
p_2	Urine test	2014-02-28	p_2	Urine test	2013-04-16
p_1	Fundus oculi	2014-03-10	p_1	Urine test	2014-09-06
p_2	Urine test	2014-04-11	p_2	Triglycerides	2014-08-01
p_1	Glucose level	2014-04-15	p_2	Urine test	2014-07-25
p_2	Electrocardiogram	2014-06-16	p_1	Fundus oculi	2014-07-10
p_1	Glucose level	2014-06-21	p_1	Urine test	2014-11-23

Definition 2.1.1. **Collection of patient records.** A collection of patient records $\mathcal{D}$ is a set of records, such that $\Sigma = \{e_1, \ldots, e_k\}$ is the set of examinations in $\mathcal{D}$ and

TABLE 2.2: VSM representation for dataset in Table 2.1

PatientID	Glucose level	Fundus oculi	Electrocardiogram	Urine test	Triglycerides
p_1	3	2	0	2	0
p_2	0	1	1	5	2

TABLE 2.3: VSM representation using the TF-IDF weighting score for dataset in Table 2.1

PatientID	Glucose level	Fundus oculi	Electrocardiogram	Urine test	Triglycerides
p_1	0.347	0	0	0	0
p_2	0	0	0.077	0	0.154

$\Theta = \{p_1, \ldots, p_n\}$ is the set of patients in $\mathcal{D}$. Each record r_k in $\mathcal{D}$ models an examination $e_j \in \Sigma$ done by a patient $p_i \in \Theta$ in a given date.

To enable the mining process and discover valuable knowledge, in the MLDA framework the collection of patient records is tailored to the Vector Space Model (VSM) representation [1] and the Term Frequency (TF) - Inverse Document Frequency (IDF) scheme [2] has been adopted to weight the examination frequency. In this study, we neglect the information on when an examination has been done because we focus on the frequency of performed examinations.

In the VSM representation, each patient p_i is a vector in the examination space. This vector represents the *patient examination history*. The vector cell (p_i, e_j) corresponds to examination e_j done by patient p_i. Cell (p_i, e_j) is a weight describing the relevance of examination e_j for patient p_i. A more formal definition of the patient examination history follows.

Definition 2.1.2. **Patient examination history.** Let $\mathcal{D}$ be a collection of patient records, $\Sigma = \{e_1, \ldots, e_k\}$ the set of examinations in $\mathcal{D}$ and $\Theta = \{p_1, \ldots, p_n\}$ the set of patients in $\mathcal{D}$. Each patient p_i in $\mathcal{D}$ is represented by a weighted examination frequency vector v_{p_i} of $|\Sigma|$ cells. Each cell $v_{p_i}[j]$ of vector v_{p_i} reports the weighted frequency w_{p_i,e_j} of examination e_j, $e_j \in \Sigma$, for patient p_i, $p_i \in \Theta$. Thus, $v_{p_i} = [w_{p_i,e_1}, \ldots, w_{p_i,e_{|\Sigma|}}]$.

Table 2.2 reports a base VSM representation for the example dataset in Table 2.1. Table 2.2 has one row for each patient in Table 2.1, and a number of columns equal to the number of different examinations in Table 2.1. Each cell (p_i, e_j) in Table 2.2 reports the weight of examination e_j for patient p_i. In this base VSM representation the weight is simply given by the number of times examination e_j was repeated by patient p_i. However, a patient data representation as in Table 2.2 may not properly characterize the patient condition. In fact, it may give more relevance to standard routine tests, which usually appear with higher frequency, than to more specific tests, which often appear with lower frequency. The adoption of the TF-IDF scheme allows highlighting

the relevance of specific examinations for a given patient condition. The TF-IDF value increases proportionally to the number of times an examination has been done by the patient, but it is offset by the frequency of the examination in the examination dataset, which helps to control the fact that some examinations are generally more common than others. The definitions of TF and IDF are given below.

Definition 2.1.3. **Term Frequency (TF) and Inverse Document Frequency (IDF).** Let $\mathcal{D}$ be a collection of patient records, $\Sigma = \{e_1, \ldots, e_k\}$ the set of examinations in $\mathcal{D}$, and $\Theta = \{p_1, \ldots, p_n\}$ the set of patients in $\mathcal{D}$.

1. For each pair (p_i,e_j) in $\mathcal{D}$, the Term Frequency TF_{p_i,e_j} is the relative frequency of examination e_j for patient p_i. It is computed as $f_{p_i,e_j} / \sum_{1 \leq k \leq |\Sigma|} f_{p_i,e_k}$, where f_{p_i,e_j} is the number of times patient p_i underwent examination e_j and $\sum_{1 \leq k \leq |\Sigma|} f_{p_i,e_k}$ is the total number of examinations done by p_i.
2. The Inverse Document Frequency IDF_{e_j} for examination e_j is the frequency of e_j in $\mathcal{D}$. It is computed as $Log[|\Theta|/|p_k \in \Theta : f_{p_k,e_j} \neq 0|]$ where $|\Theta|$ is the number of patients in $\mathcal{D}$ and $|p_k \in \Theta : f_{p_k,e_j} \neq 0|$ is the number of patients in $\mathcal{D}$ who underwent (at least once) examination e_j.

Mathematically, the base of the log function for IDF computation in Definition 2.1.3 does not matter and constitutes a constant multiplicative factor towards the overall result.

The TF-IDF weight w_{p_i,e_j} for the pair (p_i, e_j) is high when examination e_j appears with high frequency in patient p_i and low frequency in patients in the collection $\mathcal{D}$. When examination e_j appears in more patients, the ratio inside the IDF's log function approaches 1, and the IDF_{e_j} value and TF-IDF weight w_{p_i,e_j} become close to 0. Hence, the approach tends to filter out common examinations. A more formal definition of TF-IDF weight follows.

Definition 2.1.4. **TF-IDF weight.** For each pair (p_i,e_j) in $\mathcal{D}$, the TF-IDF weight w_{p_i,e_j} is computed as $w_{p_i,e_j} = TF_{p_i,e_j} * IDF_{e_j}$, where TF_{p_i,e_j} is the Term Frequency and IDF_{e_j} is the Inverse Document Frequency.

Table 2.3 reports the VSM representation using the TF-IDF scheme for the example dataset in Table 2.1. The TF-IDF weights for examinations Fondus oculi and Urine Test are equal to 0 since they are performed by both patients. Instead, TF-IDF weights are different than zero for the other examinations, which are performed by only one of the two patients.

2.1.3 Multiple-level cluster analysis

The MLDA framework applies clustering algorithms in a multiple-level fashion to progressively focus on different dataset portions and locally compute clusters. The pseudocode of the multiple-level clustering strategy is in Algorithm 1. It performs multiple runs over the considered data collection. Initially, the whole dataset is analysed. Then, at each subsequent iteration, the clustering algorithm is applied on a selected portion of the dataset, and clusters are locally identified on it. Clustering algorithm parameters can be properly set at each iteration according to the local data distribution of the considered dataset portion. Clusters computed at each iteration contribute to the final cluster set. The approach is iterated until the target objective is achieved, as the minimum threshold value of a given quality index or the maximum allowed number of clusters in the final cluster set.

Data: Initialize $\mathcal{D}$ with the whole initial data object collection
repeat
  if *first iteration* **then**
    select $\mathcal{D}$ as target dataset;
  else
    select a portion of $\mathcal{D}$ as target dataset;
  end
  apply basic clustering algorithm on the target dataset;
  update the final cluster set;
  evaluate the quality of the final cluster set;
until *target objective is verified*;

Algorithm 1: Multiple-level clustering strategy

Clustering algorithms currently integrated in MLDA are described in Section 2.1.3.1. Data objects in the analysed data collection corresponds to patients in our application scenario. For patient clustering, patient examination histories are compared using the cosine distance measure (see Section 2.1.3.2).

2.1.3.1 Multiple-level clustering algorithms

Clustering algorithms integrated in the MLDA framework are described in the following. Their main characteristics are summarized in Table 2.4, by highlighting the improvement with respect to the corresponding (not multiple-level) standard algorithms. Based on this evaluation, they appear as good candidates for the analysis considered in this study. Objects in the analyzed data collection correspond to patients in our application scenario.

Bisecting K-means [4] applies the standard K-means algorithm in a multiple-level fashion. K-means [19] discovers K clusters modeled by their representatives, named

centroids, given by the mean value of the objects in the clusters. Initially, K objects of the dataset are randomly chosen as centroids. Then, each object is assigned to the cluster whose centroid is the nearest to that object. Finally, centroids are relocated by computing the mean of the objects within each cluster. The process iterates until centroids do not change or some objective functions are achieved.

Nevertheless K-means is a widely used clustering method, it is biased to spherical clusters and it is sensitive to the initial choice of centroids. Aimed at overcoming this second limitation, the bisecting K-means algorithm adopts a multiple-level clustering approach based on a bisecting strategy. Instead of looking for all representative centroids (and corresponding clusters) at the same time, it iteratively focuses on a dataset portion and locally identifies centroids (and their clusters). More in detail, two clusters are initially generated using the standard K-means algorithm. Then, at each subsequent iteration level, a cluster is selected among those generated up to the current step. The selected cluster is split into two subclusters using K-means. K-1 level iterations are needed for discovering the desired K clusters. Different criteria can be exploited to choose the cluster to split: (i) The cluster size (i.e., the number of objects in the cluster), (ii) the cluster SSE (Sum of Squared Errors), which measures the squared total distances among cluster objects and cluster centroid, and (iii) a criterion based on both cluster size and SSE. In this study, the cluster with the largest SSE value is split.

Bisecting K-medoids [5] relies on the standard K-medoid algorithm (PAM) [20] for implementing a multiple-level clustering technique similar to bisecting K-means. K-medoid works similarly to K-means, but clusters are in this case represented by an object (*medoid*) instead of a mean point (centroid). As for bisecting K-means, bisecting K-medoids is less susceptible to the initialization problems than standard K-medoids. K-medoids methods were also investigated in this study, since they can be less sensitive to outliers than K-means methods.

Refined K-means and refined K-medoids[4]. Both bisecting strategies described above use the standard (K-means and K-medoids) clustering algorithms to bisect individual clusters. It follows that the final cluster set does not represent a local minimum with respect to the total SSE value over the whole cluster set. To deal with this problem, the cluster set generated by bisecting K-means and bisecting K-medoids can be refined as follows. The centroids (resp. medoids) in the computed cluster set are used as the initial centroids (resp. medoids) for the standard K-means (resp. K-medoids) algorithm.

Multiple-Level DBSCAN [6] progressively applies the standard DBSCAN [21] algorithm on different (disjoint) dataset portions. DBSCAN separates dense regions (with a

TABLE 2.4: Comparison of multiple-level clustering algorithms

	Bisecting and Refined K-means	**Bisecting and Refined K-medoids**	**Multiple-level DBSCAN**
Initialization problem	Reduced	Reduced	No
Sensitivity to outliers	Reduced	Reduced	No
Unclustered data objects	No	No	Reduced
Need of convex shape	Yes	Yes	No
Parameter specification	K	K	Eps, MinPts Num. of iterations
Num. of iterations	K-1	K-1	To be specified
Dealing with variable data distribution	Improved	Improved	Improved

similar density) from a sparse one in the dataset, driven by the user-specified parameters *Eps* and *MinPts*. A dense region in the data space is a n-dimensional sphere with radius *Eps* and containing at least *MinPts* objects. Objects are classified as being (i) in the interior of a dense region (a core point), (ii) on the edge of a dense region (a border point), or (iii) in a sparsely occupied region (an outlier point). A cluster contains any two core points close within a distance *Eps*, and any border point close within a distance *Eps* to at least one core point in the cluster. Outlier points are filtered out and they are unclustered.

Standard DBSCAN can discover clusters with different sizes and shapes, but it is weak in recognizing clusters with variant density. The multiple-level DBSCAN algorithm allows overcoming this limitation, by decomposing the clustering process into subsequent steps. The whole original dataset is clustered at the first level. Then, at each subsequent level, objects labeled as outliers in the previous level are re-clustered using the standard DBSCAN. With the multiple-level approach, parameters *Eps* and *MinPts* can be set at each level by adapting the definition of dense region to the local data density. Furthermore, the number of unclustered outlier points progressively reduces at each iteration level. Consequently, the multiple-level DBSCAN algorithm can finally provide a more homogenous but also richer cluster set, because it includes a larger portion of the original dataset. The number of iteration levels can be tuned based on the final number of unclustered objects and the number of computed clusters.

2.1.3.2 Comparing patient examination histories

For all clustering algorithms described above, the weighted examination frequency vectors representing the patient examination histories are compared using the cosine distance measure [2]. In our reference case study, let p_i and p_j be two arbitrary patients in the collection $\mathcal{D}$. Let v_{p_i} and v_{p_j} be the corresponding weighted examination frequency vectors. The cosine distance between patients p_i and p_j is computed as

$$dist(p_i, p_j) = \arccos\left(cos(v_{p_i}, v_{p_j})\right) \tag{2.1}$$

where the cosine similarity between patients p_i and p_j is computed as

$$cos(v_{p_i}, v_{p_j}) = \frac{v_{p_i} \bullet v_{p_j}}{\|v_{p_i}\| \left\|v_{p_j}\right\|} = \frac{\sum_{1 \le k \le |\Sigma|} v_{p_i}[k] v_{p_j}[k]}{\sqrt{\sum_{1 \le k \le |\Sigma|} v_{p_i}[k]^2} \sqrt{\sum_{1 \le k \le |\Sigma|} v_{p_j}[k]^2}}. \tag{2.2}$$

The cosine distance in Equation 2.1 verifies the triangle inequality. The cosine similarity is in the range [0,1]. $cos(v_{p_i}, v_{p_j})$ equal to 1 describes the exact similarity of examination histories for patients p_i and p_j, while $cos(v_{p_i}, v_{p_j})$ equal to 0 points out that patients have complementary histories (i.e., the sets of their examinations are disjoint).

2.1.4 Cluster evaluation based on quality indices

For the (internal) validation of clustering results, MLDA adopts the quality indices typically used for the considered algorithms. The Total SSE index [2] is used for K-means and K-medoids methods, while the Silhouette coefficient [22] for the multiple-level DBSCAN approach. Similar to [4], the overall similarity measure is used to compare cluster sets computed by different algorithms. Finally, the Rand Index [23] has been used to evaluate the agreement between different clustering results.

The **Sum of Squared Error (SSE)** is used to evaluate the cluster cohesion for center-based clusters, as clusters generated using K-means and K-medoids methods [2]. For an arbitrary patient, its error is computed as the squared distance between the patient and the centroid (resp. medoid) in the cluster including the patient. The SSE for a cluster C_i is computed as

$$SSE(C_i) = \sum_{p_j \in C_i} dist(c_i, p_j)^2 \tag{2.3}$$

where $dist(c_i, p_j)$ is the distance between the centroid (resp. medoid) c_i of cluster C_i and a patient p_j in C_i. The cosine distance metric in Equation 2.1 has been used for distance evaluation. The smaller the SSE, the better the quality of the cluster. The *Total SSE* on a set of K clusters is computed by summing up the SSE values of the K clusters.

The **Silhouette** index measures both intra-cluster cohesion and inter-cluster separation to evaluate the appropriateness of the assignment of a data object to a cluster rather

than to another one [22]. The silhouette value for a given patient p_i in a cluster C is computed as

$$s(p_i) = \frac{b(p_i) - a(p_i)}{max\{a(p_i), b(p_i)\}}, s(p_i) \in [-1, 1], \tag{2.4}$$

where $a(p_i)$ is the average distance of patient p_i from all other patients in cluster C, and $b(p_i)$ is the smallest of average distances from its neighbour clusters. The silhouette value for cluster C is the average silhouette value on all patients in C. Silhouette values in the range [0.51,0.70] and [0.71,1] show that a reasonable and a strong cluster structure has been found [20]. Lower silhouette values progressively indicate clusters with a weak structure until a no substantial structure. The cosine distance metric in Equation 2.1 has been used for silhouette evaluation.

The **Overall Similarity** index evaluates the cluster quality. In this study, it has been adopted for comparing the cluster sets from the algorithms integrated into the MLDA framework. Specifically, it is used to measure the cluster cohesiveness based on the pairwise cosine similarity of patients in a cluster. For each cluster C, the overall similarity is computed as

$$Overall\_Similarity(C) = \frac{1}{|C|^2} \sum_{\substack{v_{p_i} \in C \\ v_{p_j} \in C}} cos(v_{p_i}, v_{p_j}) \tag{2.5}$$

where $|C|$ is the cluster size, $cos(v_{p_i}, v_{p_j})$ is the cosine similarity between two patients p_i and p_j in C represented by their weighted examination frequency vectors v_{p_i} and v_{p_j}. The overall similarity on a set of K clusters is computed as the weighted similarity of the clusters

$$Overall\ Similarity = \sum_{i=1}^{K} \frac{|C_i|}{N} Overall\_Similarity(C_i) \tag{2.6}$$

where N is the total number of patients in the cluster set.

The **Rand Index** computes the number of pairwise agreements between two partitions of a set [23]. It is exploited to measure the similarity between the cluster sets obtained by two different clustering techniques. In our case study, let O be a set of N patients, and X and Y two different partitions of set O to be compared. The Rand Index R is computed as

$$R = \frac{a+b}{\binom{N}{2}} \tag{2.7}$$

where a denotes the number of pairs of patients in O which are in the same cluster both in X and Y, and b denotes the number of pairs of patients in O which do not belong to the same cluster neither in X nor in Y . Therefore, the term $a + b$ is the number of pair wise agreements of X and Y, while $\binom{N}{2}$ is the number of different pairs of elements which can be extracted from O. The Rand Index ranges from 0 to 1, where 0 indicates that the two partitions do not agree for any patient pair, and 1 that the two partitions are equivalent.

2.1.5 Cluster characterization based on exam frequency and sequential patterns

In the MLDA framework, the content of each computed cluster is concisely described as follows. (i) The *most representative examinations* occurring in their patient histories. (ii) The *temporal relationship among examinations* underwent by patients, i.e., which examinations frequently precede or follow other examinations. This information provides a more detailed characterizion of patient histories because the distribution of patient examinations over time is analysed. The two analyses can support a first categorization of the cluster content into a category of patients (possibly) affected by a given diabetes pathology. In fact, the different pathologies usually require monitoring the patient through some specific examinations.

To support temporal data analysis, the patient data collection contained in each cluster is represented as a *sequence database* [2]. Then, within each cluster, the *sequential patterns of medical examination sets* underwent by patients in subsequent days are analyzed.

Definition 2.1.5. **Sequence database.** Let $\mathcal{D}$ be a collection of patient records. Let $C \subseteq \mathcal{D}$ be a cluster on $\mathcal{D}$ containing a subset of patient records. The sequence database $\mathcal{D}_S$ defined on C is a collection of sequences $p_i{:}S$, where p_i is the patient identifier and $S{=}< s_1 \ldots s_n >$ is the temporal list of sets s_t of examinations e_j done by p_i.

When examinations are done within a short time frame, their temporal order may not be relevant being due to scheduling reasons rather than prescription constraint. For example, in the considered case study of diabetic patients, routine checks through blood tests are usually performed on the same day. Thus, in our data representation, each element s_t in a sequence $p_i{:}S$ represents the set of examinations done patient p_i on the same day.

The number of elements s_t in a sequence S is the *sequence length*. It corresponds to the number of different days in which the patient has performed at least one examination. The sequence length provides the information on how frequently the patient conditions have been monitored. For example, sequence $p_i : S =<(e_1)(e_2,e_3)(e_4)(e_1)>$ has length equal to 4 because patient p_i performed examinations on 4 different days. The sequence element $(e_2\ e_3)$ includes examinations e_2 and e_3 done on the same day.

A sequence S is said to *contain* a sequence S' if S' is a *subsequence* of S, i.e., S' contains a subset of the elements in S and preserves their order. S is called super-sequence of S'. For example, sequence $S' =<(e_3)(e_1)>$ is a subsequence of sequence $S =<(e_1)(e_2,e_3)(e_4)(e_1)>$. The *support* (or frequency) on the sequence database $\mathcal{D}_S$ of a sequence S' is the percentage of sequences in $\mathcal{D}_S$ that contain S'. A sequence S' is a *sequential pattern* if its support is above a user-specified minimum support threshold.

Mining the complete set of sequential patterns in all discovered clusters may often provide a too large solution set, making difficult for end-users the comprehension of the results. To overcome this limitation, compact representations of the sequential pattern set have been proposed (as closed sequential patterns and maximal sequential patterns). Among them, maximal sequential patterns [7] have been adopted in this study. The set of maximal sequential patterns is representative since it can be used to recover all sequential patterns, and the exact frequency of these latter can also be computed with a single database pass. Besides, the set of maximal sequential patterns is generally a small subset of the set of (closed) sequential patterns. A sequential pattern S' is said to be a *maximal sequential pattern* if there is no other sequential pattern S'' so that S'' is a superpattern of S' [7].

2.1.6 Classification model and mobile application

Clusters computed as described in Section 2.1.3.1 and characterized as reported in Section 2.1.5 can be analyzed with the support of a domain expert to describe their content from a medical perspective and assign a representative class label to each of them. Then, to automatically categorize a new patient into one cluster based on his/her examination history, a classification model can be created starting from the discovered cluster set. The possibility of automatically categorize patient histories using the classification model has been made accessible to end-users through a mobile application (app). The mobile app also allows collecting and updating the user-generated data such as examinations inserted by the patients.

2.1.6.1 Patient classification model

Classification is the task of learning a classification model that maps each data object to one of the predefined class labels [2]. A classification model is typically used to assign the class label for a new unlabeled data object. Among various classification methods, decision tree classifiers have been used in this study to characterize the results of the clustering process. *Decision trees* are powerful classification methods that have been widely used in many different application domains. Besides, they provide a readable classification model that can also serve to explain what features characterize objects in each class.

The decision tree is grown in a recursive fashion by iteratively partitioning the training records into successively purer subsets. In the tree structure, each node specifies a test on an attribute, and each branch descending from that node corresponds to one of the possible values for that attribute. Each leaf node represents class labels associated with the instances having, as attribute values, the values appearing in the path reaching the leaf node. Once the decision tree has been created, a new data object is classified by navigating the tree from the root to a leaf node, according to the outcome of the tests along the path.

For the patient representation considered in this work, each node represents one examination undergone by the patient, while each branch descending on a node represents a possible value, or a range of values, for the TF-IDF weight associated with each examination. Decision trees have been previously applied in text mining to classify documents weighted through the TF-IDF scheme [24]. In the analysis, the *Gini index* impurity-based criterion has been considered to split the record set for growing the tree. The Gini index [2] measures how often a randomly chosen instance from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset. To evaluate the quality of constructed classification model, we have adopted the three usually metrics, i.e., accuracy, precision and recall [2] (see Section 2.1.8.4).

2.1.6.2 Mobile application

This section describes the main functionalities of the proposed mobile applications, while some example screenshots are reported in Figure 2.4.

New patients can *register* to the application by inserting reference information as their fiscal code and birthdate (Figure 2.4(a)). The application allows both *visualizing and updating the list of examinations* underwent by the patient (Figures 2.4(b) and 2.4(c)). Any new underwent examination can be inserted by specifying the examination name

and code together with the date and time the patient underwent the examination. To enhance usability, the autocomplete feature is used for entering the examination name and code. Moreover, only one of the two fields must be specified, while the other is automatically filled by the application. For the diabetes dataset considered in this study, examination codes have been defined based on the ICD 9-CM (International Classification of Diseases, 9th revision, Clinical Modification) [25].

Through the application, the patient can be *classified* into one of a set of predefined categories based on his/her examination history and a precomputed classification model (Figure 2.4(d)). Moreover, the application allows *collecting feedbacks and suggestions* from the domain expert on the proposed categorization. Specifically, he/she can either confirm this categorization or suggest an alternative one, by also specifying his/her degree of expertise in the provided feedbacks (Figure 2.4(d)).

In the medical domain, possible end-users of the application are mainly medical staff and patients to some extent. The application can support the medical staff in the patient evaluation by automatically proposing the patient classification into one out of a set of predefined categories. This automatic categorization can be a valuable support since usually the classification model is computed considering large data collections, and tuned to guarantee an accurate classification. Medical staff still preserves the possibility of proposing an alternative classification based on his/her degree of expertise. The application can also support patients in a self-evaluation of their condition.

The proposed architecture includes mobile devices (e.g., tablet, smartphone) running the application and a server storing the collection of patient examination histories used for creating the classification model. A web server provides functionalities to query this repository and read/insert new data from the application.

To minimize data exchange between the server and the mobile devices at any new classification request, once generated on the server the classification model is downloaded on the mobile devices running the application. Consequently, any new classification request is locally processed by accessing the copy of the classification model stored on the mobile device. On the other hand, this local copy can be periodically updated by downloading the new version of the model generated on the server. More in detail, the classification model based on decision trees is stored on a text file as a list of if-then-else rules.

To allow enriching the central data repository available on the server, new data collected on the mobile device can be transmitted to the central server using the application. New data includes newly registered patients, updated examination histories and feedbacks provided by the domain expert. This enriched data collection can be later used for recomputing the classification model, aimed at increasing the classification accuracy.

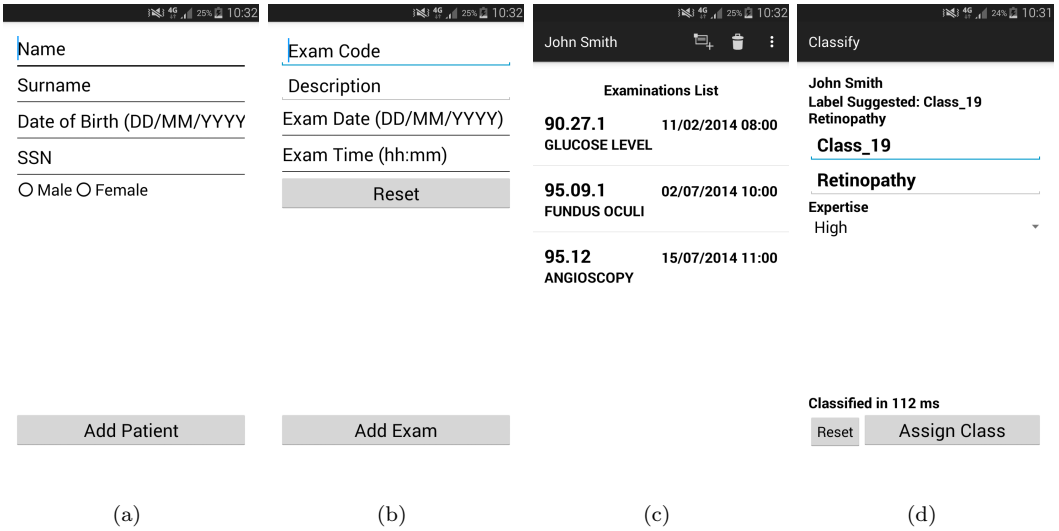


FIGURE 2.4: Mobile app: (a) patient registration, (b) insertion of a new examination done by the patient, (c) visualize patient examination history, (d) patient classification

2.1.7 Experimental results

This section presents the results of the experiments with the MLDA framework regarding (i) *quality evaluation* for the computed cluster sets, (ii) *execution time* for cluster set computation, and (iii) impact of *data dimensionality*, given by the number of different examinations used to describe patient histories, on the quality of the cluster sets. The MLDA methodology has been validated on a real collection of examination log data for diabetic patients.

2.1.7.1 Dataset

As a main reference case study we considered a real dataset of (anonymized) diabetic patients collected by an Italian Hospital. The diabetic patients dataset contains the examination log data of a set of 6,380 patients with overt diabetes, covering the time period of one year. Both male and female patients in a wide age range are included. The domain of the examinations includes 159 different examination types. Table 2.5 lists the most frequent examinations including routine examinations as well as more specific diagnostic tests for diabetes complications with varying degrees of severity. Complications due to diabetes can affect for example the cardiovascular system, eyes, and liver. The diagnostic and therapeutic procedures are defined using the ICD 9-CM (International Classification of Diseases, 9th revision, Clinical Modification) [25].

TABLE 2.5: Most frequent examinations for each category in the diabetes dataset

Category	*Examination*	*Freq.(%)*	*Category*	*Examination*	*Freq.(%)*
Routine	Glucose level	85	Liver	Alanine aminotransferase enzyme (ALT)	30
	Venous blood	79		Aspartate aminotransferase enzyme (AST)	30
	Capillary blood	75		Gamma GT	15
	Urine test	75		Bilirubin	2
	Glycated hemoglobin	46		Upper abdominal ultrasound	2
	Complete blood count	18	Kidney	Culture urine	25
Cardiovascular	Cholesterol	36		Uric acid	23
	Triglycerides	36		Microscopic urine analysis	23
	HDL Cholesterol	35		Microalbuminuria	21
	Electrocardiogram	23		Creatinine	20
Eye	Fundus oculi	27		Creatinine clearance	16
	Retinal photocoagulation	2	Carotid	ECO doppler carotid	3
	Eye examination	2	Limb	ECO doppler limb	3
	Angioscopy	2		Vibration sense thresholds	1

2.1.7.2 Evaluation setup and parameter configuration

The MLDA framework has been implemented as follows. To perform the multiple-level cluster analysis, the DBSCAN, K-means and K-medoids algorithms available in the RapidMiner toolkit have been used, and they have been applied in a multiple-level fashion. RapidMiner is an open-source platform including a number of data mining algorithms [26]. For a more accurate evaluation of the multiple-level strategy, also the standard (not multiple-level) K-means, K-medoids, and DBSCAN algorithms have been considered for performance comparison.

We developed in Java programming language the procedures for transforming the patient examination log data into the corresponding VSM representation using the TF-IDF weighting score, and for cluster evaluation through the SSE, silhouette, and overall similarity measures. Procedures for cluster evaluation have been implemented as a RapidMiner plugin. The procedure for Rand Index computation has been developed in Python programming language.

For K-means and K-medoids methods, experiments have been run by varying the K parameter, corresponding to the number of clusters in the final cluster set. For bisecting algorithms, this set is computed with K-1 iteration levels of the bisecting approach. For refined algorithms, the refinement process has been run for each final cluster set provided by bisecting algorithms. The usual approach has been adopted to address the problem of centroids and medoids initialization for bisecting algorithms, and for standard K-means and K-medoids when considered for performance comparison. Multiple runs, each with set of randomly chosen initial centroids (resp. medoids) have been performed, and then the cluster set with the minimum SSE has been selected. Specifically, RapidMiner parameters maximum number of random initialisations and maximum number of iterations for each initialisation have been set to 50 and 300, respectively, for K-means methods.

The same parameters have been set to 10 and 100 (default values in RapidMiner) for K-medoids methods because of their relevant execution time on the considered use case (see Section 2.1.7.4).

For the multiple-level DBSCAN, in setting the number of iterations, and the *Eps* and $MinPts$ values at each iteration level, we aimed at avoiding clusters with few patients, to discover representative examination sets, and at limiting the number of outlier patients, to take into account the contribution of various examination histories. Clusters should show good cohesion and separation (i.e., silhouette values greater than 0.5). Different *Eps* and $MinPts$ values have been selected at each iteration level due to the different data distribution of the dataset portion locally analyzed. This portion tends to be progressively sparser because it includes subsets of patients with more and more specific examinations (see Section 2.1.8). Consequently, at each subsequent iteration level, smaller $MinPts$ values are progressively selected to define a dense area region. The *Eps* value has been then locally tuned by trading-off the quality of the cluster set and the number of outlier patients.

Maximal sequential patterns have been extracted using the VMSP algorithm [27] available at [28]. This algorithm has been adopted because it uses a data vertical representation for a depth-first exploration of the search space, that has been shown to be effective in various domains.

To create the classification model, the decision trees algorithm available in the RapidMiner toolkit have been used. The mobile application has been developed on the Android environment version 4.4.

Experiments were performed on a 2.66-GHz Intel(R) Core(TM)2 Quad PC with 8 GBytes of main memory, running linux (kernel 3.2.0).

2.1.7.3 Cluster quality evaluation

The quality for the computed cluster sets has been evaluated based on the SSE (for K-means and K-medoids methods), Silhouette (for DBSCAN methods), and overall similarity (for all methods) measures.

Evaluation of K-means methods

For all K-means methods, the total SSE measure progressively decreases, and the overall similarity measure progressively increases, when growing the value of K and thus the number of clusters (see Figure 2.5). The bisecting K-means algorithm always provides the worst results for both measures, i.e., the cluster sets with the highest total SSE and

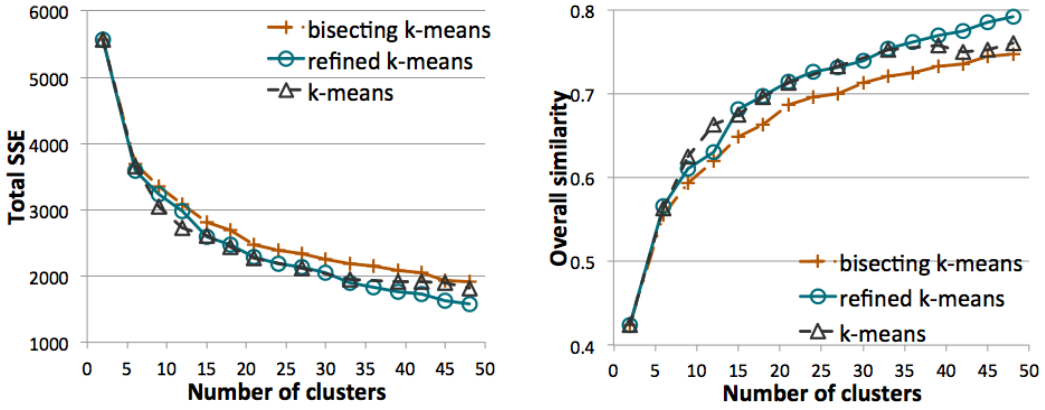


FIGURE 2.5: K-means methods: quality of the cluster set when varying the number of clusters

the lowest overall similarity values. Nevertheless, the refined K-means algorithm always provides better results than bisecting K-means, showing that the use in a subsequent clustering phase of the "centroids" computed with the bisecting K-means algorithm can improve the quality of the final cluster set.

Compared to standard K-means, the refined K-means algorithm provides better results when increasing K (about K> 30, i.e., more than 30 clusters). It is worse than standard K-means when a lower value of K is considered ($5 \leq K \leq 15$, i.e., between 5 and 15 clusters). It follows that the final cluster set can benefit from a multiple-level clustering strategy when the number of iteration levels, and thus the final number of clusters, increases. The K parameter can be selected based on the desired number of clusters and the expected quality of the cluster set.

As a reference example, Table 2.6 reports the main characteristics of the solution with 32 clusters, in terms of number of patients, different examinations, SSE and overall similarity for each cluster.

Evaluation of K-medoids methods

The experimental results reported in Figure 2.6 show that K-medoids methods exhibit a similar behavior to K-means ones. The bisecting K-medoids algorithm always provides the worst results in terms of overall similarity and total SSE values. The refined K-medoids algorithm always improves bisecting K-medoids and provides comparable results to standard K-medoids.

K-medoids methods showed a very high computational cost which limited their applicability in the MLDA framework (see Section 2.1.7.4). Due to this cost, solution sets with a larger number of clusters have not been generated.

TABLE 2.6: Detailed clustering results for refined K-means

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Number of patients	96	172	169	97	124	239	233	206	13	88	38	376
Number of examinations	42	39	25	18	52	51	44	40	31	34	38	60
SSE	67.6	112	39.9	8.72	43.3	105	88.7	65.5	5.17	22.4	14.7	134
Overall similarity	0.51	0.50	0.80	0.92	0.70	0.63	0.67	0.72	0.70	0.79	0.67	0.69
	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}	C_{21}	C_{22}	C_{23}	C_{24}
Number of patients	18	78	402	231	351	50	47	26	201	182	226	146
Number of examinations	33	30	56	44	67	28	34	51	37	41	46	54
SSE	7.43	38.3	137	100	149	24.2	17.6	15.7	98.7	48.9	76.3	113
Overall similarity	0.66	0.61	0.70	0.63	0.64	0.60	0.69	0.54	0.59	0.77	0.71	0.45
	C_{25}	C_{26}	C_{27}	C_{28}	C_{29}	C_{30}	C_{31}	C_{32}				
Number of patients	74	1,126	509	61	169	170	257	205				
Number of examinations	39	35	35	40	20	24	28	30				
SSE	58.7	55.3	57	34.2	22.5	65.5	43.9	55.8				
Overall similarity	0.43	0.96	0.90	0.57	0.88	0.76	0.85	0.76				

Whole cluster set	
Total SSE	1,926.01
Overall similarity	0.75

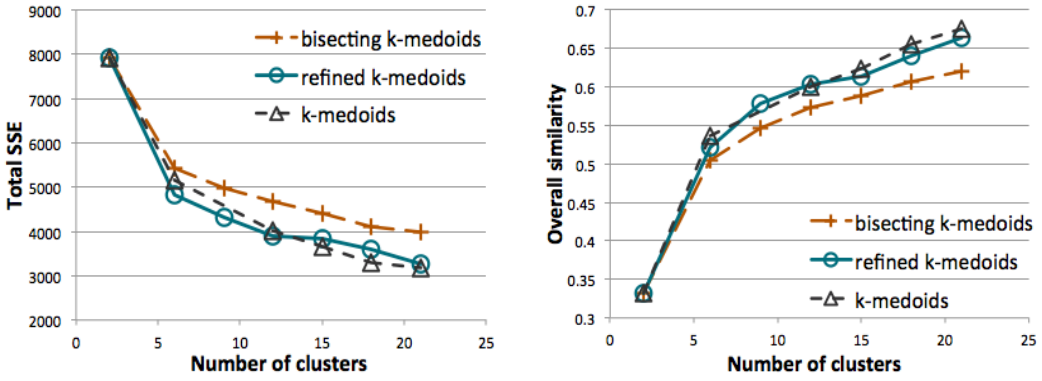


FIGURE 2.6: K-medoids methods: quality of the cluster set when varying the number of clusters

Evaluation of DBSCAN methods

As reported in Table 2.7, when iterating the multiple-level DBSCAN approach for four levels, 32 clusters are computed in total showing good overall similarity and silhouette values (greater than 0.5). These clusters globally includes 3,510 patients (about 55% of the diabetes dataset). Most patients belong to clusters computed at the first level, while a comparable number of patients is included in clusters computed at the next levels. After four iterations, 2,870 patients are labeled as outliers and remain unclustered. Note that these patients can be additionally clustered by iterating the approach for more levels.

Clustering about 55% of the patients using the standard DBSCAN algorithm generates a lower quality cluster set than when using the multiple-level DBSCAN approach. To deepen into the analysis of this point, Figure 2.7 plots the silhouette and overall similarity

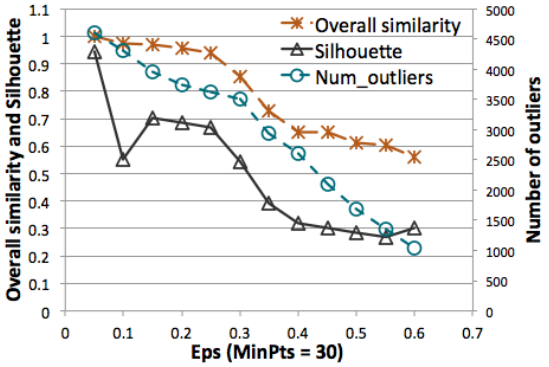


FIGURE 2.7: DBSCAN algorithm: quality of the cluster set and number of outlier patients when varying the *Eps* value ($MinPts$=30)

values, and number of outlier patients, when the whole patient collection is analyzed using the standard DBSCAN. With parameters Eps=0.36 and $MinPts$=30, a cluster set is generated including almost the same number of patients than the cluster set from the multiple-level DBSCAN approach, but with a significantly lower quality. The overall similarity value is 0.73 and the silhouette is 0.4 (i.e., lower than 0.5), while these values are 0.85 and 0.55, respectively, for the multiple-level DBSCAN when iterated for four levels (see Table 2.7). It follows that, also for the DBSCAN method, the final cluster set can benefit of the multiple-level strategy.

Clusters computed with four level iterations are described in Table 2.8 in terms of their number of patients, different examinations and overall similarity value, while silhouette plot is reported in Figure 2.8. Clusters mainly show a rather prominent silhouette. Few patients have negative silhouette values in clusters computed at the first level, i.e., 198 patients out of 1,764 in cluster C_{1_1}, 2 patients out of 223 in C_{2_1} and 7 patients out of 294 in C_{4_1}. At the fourth level, cluster C_{3_4} shows a less prominent silhouette, but the average silhouette value is almost 0.5.

TABLE 2.7: Clustering results for multiple-level DBSCAN

	1st level	2nd level	3rd level	4th level
(MinPts, Eps)	(30, 0.3)	(30, 0.5)	(20, 0.5)	(10, 0.35)
Number of clusters	11	5	4	12
Number of patients	2,872	260	104	274
Silhouette	0.54	0.61	0.66	0.6
Overall similarity	0.85	0.86	0.89	0.94

Whole cluster set	
Number of clusters	32
Number of clustered patients	3,510
Number of outliers	2,870
Silhouette	0.55
Overall similarity	0.86

TABLE 2.8: Detailed clustering results for multiple-level DBSCAN

	First-level										
	C_{1_1}	C_{2_1}	C_{3_1}	C_{4_1}	C_{5_1}	C_{6_1}	C_{7_1}	C_{8_1}	C_{9_1}	C_{10_1}	C_{11_1}
Number of patients	1,764	223	140	294	144	110	42	43	35	36	41
Number of examinations	10	6	8	7	6	2	7	8	9	19	2
Overall similarity	0.82	0.87	0.94	0.88	0.92	1.00	0.96	0.94	0.97	0.94	1.00

	Second-level					**Third-level**			
	C_{1_2}	C_{2_2}	C_{3_2}	C_{4_2}	C_{5_2}	C_{1_3}	C_{2_3}	C_{3_3}	C_{4_3}
Number of patients	75	73	49	30	33	32	29	21	22
Number of examinations	35	27	15	16	8	22	19	14	15
Overall similarity	0.84	0.85	0.91	0.89	0.86	0.9	0.83	0.92	0.91

	Fourth-level											
	C_{1_4}	C_{2_4}	C_{3_4}	C_{4_4}	C_{5_4}	C_{6_4}	C_{7_4}	C_{8_4}	C_{9_4}	C_{10_4}	C_{11_4}	C_{12_4}
Number of patients	19	19	100	12	14	14	24	30	10	10	12	10
Number of examinations	7	3	20	9	8	19	12	12	9	16	4	19
Overall similarity	0.93	1	0.91	0.98	0.95	0.94	0.94	0.92	0.94	0.94	0.99	0.95

Whole cluster set	
Overall similarity	0.86

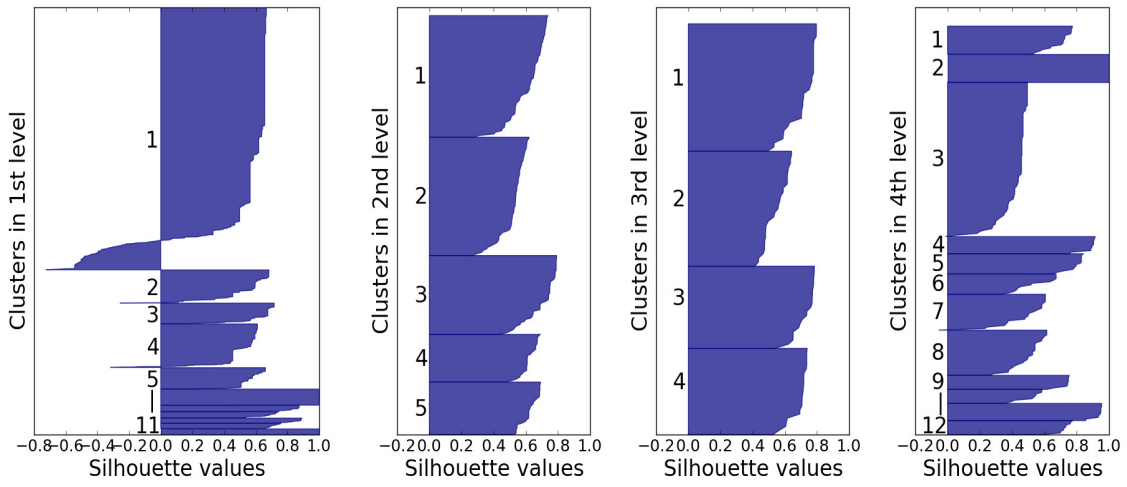


FIGURE 2.8: Silhouette plot for multiple-level DBSCAN

2.1.7.4 Execution time

For the multiple-level DBSCAN algorithm, the total run time for computing a solution with 32 clusters is 13min 40s. The first, second, third and fourth iteration level require 3min 34s, 3min 8s, 3min, and 2min 58s, respectively. The time tends to progressively reduce at each level because a smaller dataset portion is progressively analysed.

The run time for bisecting and refined K-means algorithms for computing a solution with 32 clusters is (slightly) lower than for the multiple-level DBSCAN approach. Bisecting k-means requires 10 min, while refined K-means requires 7s in addition for the refinement

of centroids (i.e., to run K-means after having initialized centroids). The time for K-means is about 2 minutes.

The run time is significantly higher for bisecting K-medoids, making the approach not suitable for datasets with many examinations as the one considered in this study. The time is approximately 38 hours for generating a set of 20 clusters, while refined K-medoids requires 34 min in addition for the refinement of medoids. The time for K-medoids is about 5 hours and a half.

2.1.7.5 Impact of data dimensionality on cluster sets

In the patient data representation considered in this study, the data dimensionality is given by the set of examinations describing the patient examination history. When the cardinality of this set increases, a larger set of facets characterizes patient care plans. Besides routine tests, also more specific examinations are considered, which are progressively undergone by a reduced number of patients. Consequently, the patient distribution tends to become increasingly sparser, and the computation of cohesive clusters becomes more complex.

To evaluate how data dimensionality impacts on the quality of the cluster set, in addition to the whole diabetes dataset (with 159 examinations), two other configurations of this dataset have been considered, including about 60% and 40% of the most frequent examinations (i.e., 60 and 30 examinations, respectively). The three datasets contain the same number of patients, showing that patient histories include various examinations, possibly repeated a different number of times by each patient. The multiple-level DBSCAN and the refined K-means algorithms have been considered as reference example methods for this analysis.

For refined K-means, given a number of clusters, the overall similarity value decreases, and the total SSE increases, as the number of examinations (and thus the dataset sparsity) increases (see Figure 2.9). Consequently, when the number of examinations increases, a larger number of clusters should be generated to discover cohesive groups of patients. For example, the overall similarity value gradually tends to 0.8 when considering 20 clusters for dataset with 30 examinations and 40 clusters for datasets with 60 and 159 examinations.

The multiple-level DBSCAN has been iterated for four levels for all three datasets, aimed at generating cluster sets with comparable good quality in terms of overall similarity and silhouette values. As the number of examinations increases (and thus the dataset sparsity), the final number of patients labeled as outliers, and thus unclustered, decreases.

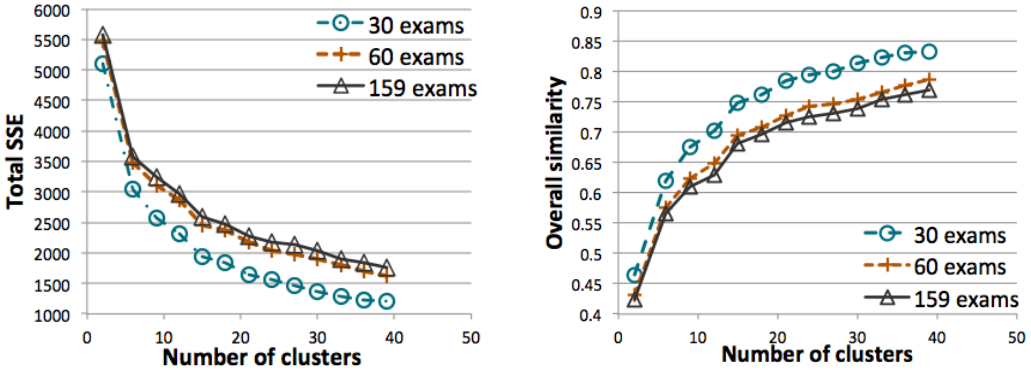


FIGURE 2.9: Refined K-means on the three datasets: quality of the cluster set when varying the number of clusters

After four iterations, the final number of outliers is 2,573, 2,678 and 2,870 for datasets with 30, 60, and 159 examinations, respectively (see Tables 2.7 and 2.9). It follows that when the dataset sparsity increases, more iterations are needed to cluster a larger subset of patients but preserving the quality of the cluster set.

TABLE 2.9: Clustering results for multiple-level DBSCAN on datasets with 30 and 60 examinations

	30 examinations				60 examinations			
	1st level	2nd level	3rd level	4th level	1st level	2nd level	3rd level	4th level
(MinPts, Eps)	(50, 0.3)	(20, 0.45)	(10, 0.4)	(15, 0.25)	(30, 0.3)	(30, 0.55)	(15, 0.25)	(10, 0.6)
Number of clusters	6	12	7	10	11	6	10	14
Number of patients	2,837	617	147	206	2,891	358	186	267
Silhouette	0.56	0.60	0.72	0.64	0.54	0.54	0.70	0.6
Overall similarity	0.84	0.89	0.90	0.98	0.85	0.83	0.99	0.65
Whole cluster set								
Number of clusters	35				41			
Number of clustered patients	3,807				3,702			
Number of outliers	2,573				2,678			
Silhouette	0.57				0.55			
Overall similarity	0.86				0.84			

2.1.8 Discussion

Here we discuss the clustering results discovered through the MLDA framework. The discussion addresses the performance comparison for clustering methods, the comparison from a medical perspective for discovered cluster sets, and the cluster characterization in terms of association rules.

2.1.8.1 Performance comparison

Concerning *K-means methods*, *refined K-means* in particular benefits of the multiple-level strategy. The quality of the final cluster set is at least comparable to the cluster quality of standard and bisecting K-means algorithms, but it outperforms them when the approach is iterated for more levels. Also the *multiple-level DBSCAN* algorithm pointed out the improvement in adopting a multiple-level strategy with respect to the standard DBSCAN in the considered case study. On the contrary, *K-medoids methods* do not seem suitable to be used in a multiple-level fashion in our case study, because they provide cluster sets with lower quality. For example, for the solution with 21 clusters, the overall similarity is 0.67 and total SSE is 3,200 for K-medoids methods (see Figure 2.6), while these measures are 0.71 and 2,275 for K-means methods (see Figure 2.5). In addition, the high computational time of K-medoids methods limits the possibility of iterating them for more levels, thus progressively improving cluster quality.

Based on the discussion above, we focused our attention on comparing the refined K-means and the multiple-level DBSCAN algorithms. Let us consider, as a reference example, the solutions with 32 clusters generated by the two algorithms on the whole dataset with 159 examinations. The following considerations hold. (i) Both *cluster sets exhibit good quality* in terms of overall similarity, even if this value is higher for multiple-level DBSCAN (0.86, see Table 2.7) than for refined K-means (0.75, see Figure 2.5). (ii) In both cases, the clustering process requires a *comparable and acceptable execution time*, slightly lower for refined K-means (about 10min) than for multiple-level DBSCAN (about 13min). Thus, (iii) in both cases the multiple-level strategy can be potentially *iterated for more levels* by further increasing the quality of the final cluster set. Specifically, the unclustered outlier patients can be progressively reduced for multiple-level DBSCAN, while clusters can be split into more cohesive subclusters for refined K-means.

To deepen into the comparison of the two algorithms, the agreement between the two cluster sets is evaluated using the Rand Index. While refined K-means clusters the whole dataset, the multiple-level DBSCAN clusters a subset, since outlier patients are grouped into a separate cluster. The following two options are considered to guarantee the same number of patients in the compared cluster sets. The separate cluster of outlier patients is (a) excluded from, or (b) it is included in, the final cluster set generated by the multiple-level DBSCAN algorithm. In case (a), the outlier patients are also removed from clusters computed by the refined K-means algorithm. The Rand Index value shows a good agreement between the two clustering results, higher in option (a) (Rand Index = 0.83) than in option (b) (Rand Index = 0.73). It follows that the two cluster sets mainly differ on the patients labeled as outliers. While they are isolated by multiple-level DBSCAN, they are clustered together with other patients by refined K-means.

2.1.8.2 Comparison from a medical perspective

Discovered cluster sets are also analysed from a medical perspective. Following the discussion on performance comparison in Section 2.1.8.1, we focused on the multiple-level DBSCAN and the refined k-means algorithms, and we analysed and compared the solutions with 32 clusters computed on the whole dataset with 159 examinations.

Nevertheless the two algorithms generate cluster sets with good quality and agreement, from a medical perspective the *multiple-level DBSCAN* appears as the *more suitable approach* for patient analysis. The refined K-means algorithm is less effective in partitioning the initial data collection into subsets with different data distributions, i.e., including patients with (significantly) different examination histories. Instead, the multiple-level DBSCAN algorithm isolates these outlier patients, and separately analyzes them in a subsequent clustering phase. Since refined K-means computes a cluster set including *all* the patients in the original dataset, these outlier patients are always assigned to some clusters, thus increasing the variety of examinations in each cluster.

More in detail, unlike refined K-means, the multiple-level DBSCAN approach computed clusters including, on average, a limited number of different examinations. These clusters contain from 2 to 35 different examinations and about 12 on average (see Table 2.8), while clusters from refined K-means include from 18 to 67 different examinations and about 38 on average (see Table 2.6). In addition, clusters from refined K-means mostly contain patients with diversified examination histories, including both routine and more specialized examinations to test different diabetes complications. Instead, in clusters from multiple-level DBSCAN, the number of examinations tend to increase with the iteration levels, thus progressively including more specialized examinations.

For both methods, the content of some example clusters, in terms of the most frequent examinations in the cluster, is reported in Table 2.10. For the multiple-level DBSCAN, first-level clusters contain patients who mostly performed standard routine tests to monitor diabetes conditions (cluster C_{2_1}). Second-level clusters contain patients tested with an increasing number of specific examinations, showing that patients can be affected by a particular disease complication or by more disease complications (e.g., on cardiovascular and eye system in cluster C_{5_2}). Examinations become progressively more numerous and specific in third- and fourth-level clusters, indicating patients that can have diabetes complications of increasing severity (clusters C_{1_3} and C_{12_4}). Instead, in clusters from refined K-means, examinations cover most categories. Thus, patients with different disease complications can be included in the same cluster (clusters C_2, C_5, C_{11} and C_{21}).

Being clusters computed using the multiple-level DBSCAN algorithm rather homogeneous in their patient examination histories, clinical domain experts can inspect the

TABLE 2.10: Multiple-level DBSCAN and refined K-means: most frequent examinations in some example clusters (examination frequencies are in %)

		Multiple-level DBSCAN				Refined K-means			
Category	*Examination*	1st level	2nd level	3rd level	4th level	C_2	C_5	C_{11}	C_{21}
		C_{2_1}	C_{5_2}	C_{1_3}	C_{12_4}				
Routine	Glucose level	78	100	75	100	68	94	63	90
	Capillary blood	72	97	72	100	58	69	61	57
	Urine test	72	100	72	100	60	68	61	55
	Venous blood	96	91	69	70	56	98	68	96
	Glycated Hemoglobin	100	76	16	10	24	90	40	79
	Complete Blood Count	-	-	-	-	5	73	16	100
Cardiovascular	Cholesterol	-	-	13	10	10	85	37	70
	Triglycerides	-	-	13	1	11	84	37	69
	HDL Cholesterol	-	-	13	10	10	84	37	67
	Electrocardiogram	-	79	25	-	20	25	26	15
Eye	Fundus oculi	-	100	-	20	26	34	45	20
	Retinal photocoagulation	-	-	-	-	-	1	3	-
	Eye examination	-	-	-	-	1	7	8	1
	Angioscopy	-	-	100	-	-	2	8	-
Liver	ALT	-	-	-	10	9	95	26	50
	AST	-	-	-	10	10	97	29	49
	Gamma GT	-	-	-	10	5	83	18	10
	Bilirubin	-	-	-	-	-	95	-	-
	Upper abdominal ultrasound	-	-	-	-	1	6	3	2
Kidney	Culture urine	-	-	-	-	7	52	37	20
	Uric acid	-	-	-	10	6	65	21	33
	Microscopic urine analysis	-	-	-	10	4	69	13	50
	Microalbuminuria	-	-	-	-	6	44	26	11
	Creatinine	-	-	-	-	4	61	13	29
	Creatinine clearance	-	-	-	10	6	29	18	11
Carotid	ECO doppler carotid	-	-	-	-	67	4	11	2
Limb	ECO doppler limb	-	-	-	10	53	2	16	2
	Vibration sense thresholds	-	-	-	100	-	2	-	2

cluster content from a medical perspective to support various analysis as for example those reported below. (a) Discover, for each cluster, the examinations actually prescribed to diabetic patients included in the cluster. (b) Check the coherence between the underwent examinations in each cluster and the existing medical guidelines for diabetes disease [25]. (c) Provide feedbacks to health care organizations to improve the application of the existing medical guidelines, but also to enrich these guidelines or assess new ones.

2.1.8.3 Cluster characterization based on sequence pattern analysis

To analyse the temporal order of examinations when testing patients, the cluster content has been described using sequence patterns. The average length of sequences describing patient histories increases from each subsequent level of clustering. The sequence length represents the number of different days in which patients had at least one examination. It corresponds to the frequency used to monitor patients within the time period of one

year covered in the considered dataset. The average sequence length is lower for patients in clusters at the first level (about 2.4), including patients mainly monitored through periodic routine tests. Instead, it increases in the next levels being patients tested with more specific examinations to check diabetes complications in addition to routine tests (about 4.3 in the second level and 3.6 in the third and fourth level).

As an example of the type of information that can be mined, some maximal sequential patterns are reported in Table 2.11 for the multiple-level DBSCAN clusters in Table 2.10. Sequences S_1 and S_2 in the first-level cluster C_{2_1} mainly show the periodic repetitions of the routine examinations used to monitor patient conditions. Routine blood examinations are usually performed on the same day, possibly together with urine test. In next level clusters, sequences include routine examinations interleaved with more specific examinations to test diabetes at different degrees of severity. Sequences tend to be progressively characterized by lower support values, being patient histories more diversified.

In the second level cluster C_{5_2}, sequence S_3 show that the eye examination fundus oculi is followed by repetition of routine tests. In third level cluster C_{1_3}, the angioscopy examination preceded and/or follows routine tests (sequence S_5) and/or more specific examinations to monitor possible cardiovascular complications and cholesterol concentration (sequence S_6). The angioscopy examination is an eye examination allowing a deeper analysis of patient eye condition (than other examination as fundus oculi) and it is typically underwent by patients with possible retinopathy.

TABLE 2.11: Example of maximal sequential patterns for some clusters from multiple-level DBSCAN

Cluster	**Maximal sequential patterns**	**Sup.(%)**
C_{2_1}	S_1: < (Venous blood,Glycated Hemoglobin)(Glucose level,Capillary blood, Urine test,Venous blood)(Glucose level,Capillary blood,Urine test,Venous blood) >	14.8
	S_2: < (Venous blood,Glycated Hemoglobin)(Glucose level,Venous blood, Glycated Hemoglobin)(Glucose level,Venous blood) >	5.38
C_{5_2}	S_3: < (Fundus oculi)(Glucose level,Capillary blood,Urine test,Venous blood, Glycated Hemoglobin)(Glucose level,Capillary blood,Urine test,Venous blood)>	18.18
C_{1_3}	S_4: <(Angioscopy)(Triglycerides,Cholesterol,Glycated Hemoglobin,HDL Cholesterol) (Glucose level,Capillary blood,Urine test,Venous blood) >	6.25
	S_5: <(Glucose level,Capillary blood,Urine test,Venous blood)(Electrocardiogram) (Angioscopy) >	9.38
C_{12_4}	S_6: <(Capillary blood,Urine test,Glucose level) (Capillary blood,Vibration sense thresholds,Glucose level,Urine test) (Capillary blood,Urine test,Glucose level)(Glucose level,Urine test,Capillary blood) (Venous blood,Glucose level,Capillary blood,Urine test)>	30

2.1.8.4 Patient classification results

The clustering results were also evaluated by a domain expert to describe the cluster content from a medical perspective and assign a class label to each cluster. For example,

considering clusters in Table 2.10, patients in cluster C_{5_2} may be affected by retinopathy, while patients in cluster C_{2_1} are (probably) not affected by diabetes complications. Then, a classification model based on decision trees was built starting from the cluster set computed with the multiple-level DBSCAN approach when iterated for four levels summarized in Table 2.7 and detailed in Table 2.8. To preserve the characteristics of the discovered clusters, where patients with similar examination histories have been grouped together, each cluster has been labeled with a different class label.

The 7-fold cross validation method has been adopted for evaluating the classification model based on accuracy, precision and recall measures. The *accuracy*, measuring the overall quality of the classifier, is the ratio of the number of correctly classified patients over the total number of given patients. Precision and recall analyse the performance of the classifier with respect to a given class c. *Precision* is the number of patients correctly classified in c divided by the number of patients classified in c. *Recall* is the number of patients correctly classified in c divided by the number of patients labeled with c in the collection. The experimental result showed the goodness of the constructed model. The accuracy value is about 97.3%. The average recall value is around 88%, except for clusters C_{6_4} (78.6%), C_{1_3} (71.9%) and C_{2_4} (47.4%), and the precision value has an average of 91%, apart from clusters C_{2_4} (69.2%) and C_{10_4} (72.7%). The values are all very high, which guarantees the quality of the classification model.

The final decision tree contains 146 nodes, 74 paths with average length 9.53, and leaf nodes with quite good degree of purity. The creation time was about 30 sec on a 2.66-GHz Intel(R) Core(TM)2 Quad PC with 8 GBytes of main memory, running linux (kernel 3.2.0). For locally accessing the classification model on the mobile device, the decision tree was transformed into the corresponding textual representation as an ordered list of if-the-else rules. This text file has size 16 Kbytes. The classification time of a new patient is about 140 milliseconds using as mobile device a Samsung Galaxy A5 smartphone running Android 4.4.4 based on 1.2 GHz Quad Core Qualcomm Snapdragon 410 Cortex-A53 processor with 2 GB RAM.

2.2 Analysis of User-Generated Content from Twitter

Recently, social networks and online communities, such as Twitter and Facebook, have become a powerful source of knowledge being daily accessed by millions of people. A particular attention has been paid to the analysis of the User-Generated Content (UGC) coming from Twitter, which is one of the most popular microblogging websites. Twitter, currently the leading microblogging social network, has attracted a great body of

research works. Tweets are short, user-generated, textual messages of at most 140 characters long and publicly visible by default. For each tweet a list of additional features (e.g., GPS coordinates, timestamp) on the context in which tweets have been posted is also available.

This section focuses on the analysis of the textual part of Twitter data (i.e., on tweets) to provide summary insight into some specific aspects of an event or discover user thoughts associated with specific events. The MLDA framework is applied to discover groups of similar twitter messages posted on a given event. By analyzing these groups, user emotions or thoughts that seem to be associated with specific events can be extracted, as well as aspects characterizing events according to user perception. To deal with the inherent sparseness of micro-messages, the proposed approach relies on the multiple-level clustering strategy that allows clustering text data with a variable distribution. Clusters are then characterized through the most representative words appearing in their messages, and association rules are used to highlight correlations among these words. Association rules [29] identify collections of itemsets (i.e., sets of words in the tweet analysis) that are statistically related in the underlying dataset. To measure the relevance of specific words for a given event, text data has been represented in the Vector Space Model using the TF-IDF weighting score. As a reference case study, the proposed framework has been applied to two real datasets retrieved from Twitter.

A simplified example of the textual part of two Twitter messages is shown in Figure 2.10. Both tweets regard the Paralympic Games that took place in London in year 2012. As described in Section 2.2.2, to suit the textual data to the subsequent data mining steps, tweets are preprocessed in the framework by removing links, stopwords, no-ascii chars, mentions, and replies.

Our proposed framework assigns the two example tweets to two different clusters, due to their quite unlike textual data. Both example tweets contain words as $\{paralympics, olympic, stadium\}$, overall describing the paralympics event. In addition, $\{fireworks, closingceremony\}$ and $\{amazing, athletics\}$ are the representative word sets for Tweets 1 and 2, respectively, reporting the specific subject of each message. While the first tweet talks about a specific event in the closing ceremony (i.e., the fireworks), the second one reports a positive opinion of people attending the event.

```
TWEET 1 - text: {Fireworks on! paralympics closingceremony at Olympic Stadium}
TWEET 2 - text: {go to Olympic Stadium for amazing athletics at Paralympics}
```

FIGURE 2.10: Two simplified example tweets

The proposed framework to analyse Twitter data is shown in Figure 2.11 and detailed in the following subsections. This research work has been summarized and published in paper [30].

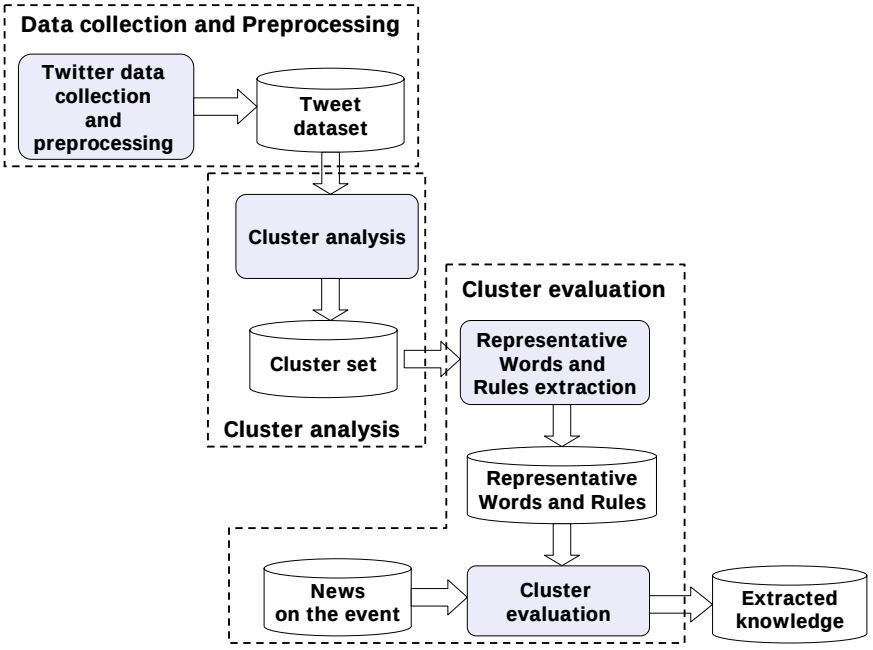


FIGURE 2.11: The proposed multiple-level clustering framework for tweet analysis

2.2.1 Related work

The application of data mining techniques to discover relevant knowledge from the User-Generated Content (UGC) of online communities and social networks has become an appealing research topic. Many research efforts have been devoted to improving the understanding of online resources [31, 32], designing and building query engines that fruitfully exploit semantics in social networks [33, 34], and identifying the emergent topics [35, 36]. Research activity has been carried out to on Twitter data to discover hidden co-occurrences [32] and associations among Twitter UGC [37, 38, 39], and analyse Twitter UGC using clustering algorithms [40, 41, 42].

In [32] frequently co-occurring user-generated tags are extracted to discover social interests for users, while in [39] association rules are exploited to visualize relevant topics within a textual document collection. [38] discovers trend patterns in Twitter data to identify users who contribute towards the discussions on specific trends. The approach proposed in [37], instead, exploits generalized association rules for topic trend analysis. A parallel effort has been devoted to studying the emergent topics from Twitter UGC [35, 36]. For example, in [36] bursty keywords (i.e., keywords that unexpectedly increase the appearance rate) are firstly identified. Then, they are clustered based on their co-occurrences.

Research works also addressed the Twitter data analysis using clustering techniques. [40] proposed to overcome the short-length tweet messages with an extended feature

vector along with a semi-supervised clustering technique. The wikipedia search has been exploited to expand the feature set, while the bisecting k-Means has been used to analyze the training set. In [41], the Core-Topic-based Clustering (CTC) method has been proposed to extract topics and cluster tweets. Community detection in social networks using density-based clustering has been addressed in [42] using the density-based OPTICS clustering algorithm.

Unlike the above cited papers, the PhD research work summarised in [30] jointly exploits a multiple-level clustering technique and association rules mining to compactly point out, in tweet collections with a variable distribution, the information posted on an event.

2.2.2 Data collection and preprocessing

Tweet content and their relative contextual data are retrieved through the Stream Application Programming Interfaces (APIs). Data is gathered by establishing and maintaining a continuous connection with the stream endpoint.

To suit the raw tweet textual to the following mining process, some preliminary data cleaning and processing steps have been applied. The textual message content is first preprocessed by eliminating stopwords, numbers, links, non-ascii characters, mentions, and replies. Then, it is represented by means of the Bag-of-Word (BOW) representation, usually adopted in text mining [4]. The message is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping word multiplicity.

Tweets are transformed using the Vector Space Model (VSM) [4]. Each tweet is a vector in the word space. Each vector element corresponds to a different word and is associated with a weight describing the word relevance for the tweet. The Term Frequency (TF) - Inverse Document Frequency (IDF) scheme [4] has been adopted to weight word frequency. This data representation allows highlighting the relevance of specific words for each tweet. It reduces the importance of common terms in the collection, ensuring that the matching of tweets is more influenced by discriminative words with relatively low frequency in the collection. In short-messages as tweets, the TF-IDF weighting score could actually boiled down to a pure IDF due to the limited word frequency within each tweet. Nevertheless, we preserved the TF-IDF approach to consider also possible word repetitions.

The tweet collection is then partitioned based on trending topics, identified by analysing the most frequent hashtags. A dataset partition is analyzed as described in the following sections.

2.2.3 Cluster analysis

Among the multiple-level clustering algorithms integrated in the MLDA framework, the multiple-level DBSCAN method has been selected for tweet analysis. As described in Section 2.1, the experiments on patient treatments analysis show that the multiple-level DBSCAN provides better performance than the other algorithms. Differently from other clustering methods, density-based algorithms can effectively discover clusters of arbitrary shape and filter out outliers, thus increasing cluster homogeneity. Additionally, the number of expected clusters in the data is not required. Tweet datasets can include outliers as messages posted on some specific topics and clusters can be non-spherical shaped. Besides, the expected number of clusters can be hardly guessed a priori, because our aim is discovering groups of similar tweets through an explorative data analysis.

However, one single execution of DBSCAN discovers dense groups of tweets according to one specific setting of the Eps and $MinPts$ parameters. Tweets in lower density areas are labeled as outliers and not assigned to any cluster. Hence, different parameter settings are needed to discover clusters in datasets with a variable data distribution as the one considered in this study. To overcome these issues, DBSCAN has been applied in multiple-level fashion. The whole original dataset is clustered at the first level. Then, at each subsequent level, tweets labeled as outliers in the previous level are re-clustered. The DBSCAN parameters Eps and $MinPts$ are properly set at each level by addressing the following issues. To discover representative clusters for the dataset, we aim at avoiding clusters including few tweets. In addition, to consider all different posted information, we aim at limiting the number of tweets labeled as outliers and thus unclustered.

The cosine similarity measure has been adopted to evaluate the similarity between tweets represented in the VSM model using the TF-IDF method. This measure has been often used to compare documents in text mining [4].

2.2.4 Cluster evaluation

The discovered cluster set is evaluated using the Silhouette index [20]. Silhouette allows evaluating the appropriateness of the assignment of a data object to a cluster rather than to another by measuring both intra-cluster cohesion and inter-cluster separation (see Section 2.1.4).

Negative silhouette values represent wrong tweet placements, while positive silhouette values a better tweet assignments. Clusters with silhouette values in the range [0.51,0.70] and [0.71,1] respectively show that a reasonable and a strong structure have been found

[20]. The cosine similarity metric has been used for silhouette evaluation, since this measure was used to evaluate tweet similarity in the cluster analysis (see Section 2.2.3).

Each cluster has been characterized in terms of the words appearing in its tweets and the association rules modeling strong correlations among these words. News available on the web are used to properly frame the context in which tweets were posted and validate the extracted information. Specifically, the most representative words for each cluster are highlighted. These words are the relevant words for the cluster based on the TF-IDF weight. They occur with higher frequency in tweets in the cluster than in tweets contained in other clusters.

An association rule is an implication in the form $X \rightarrow Y$ on dataset $\mathcal{D}_{tweets}$, where X and Y are disjoint itemsets (i.e., sets of data items). X and Y are also denoted as antecedent and consequent of the rule. The quality of an association rule $X \rightarrow Y$ is usually measured by rule support and confidence. Rule support is the percentage of tweets containing both X and Y, i.e., $supp(R) = \frac{supp(X \cup Y)}{|\mathcal{D}_{tweets}|}$, where $supp(X \cup Y)$ is the number of tweets in $|\mathcal{D}_{tweets}|$ containing both X and Y and $|\mathcal{D}_{tweets}|$ is the number of tweets in $\mathcal{D}_{tweets}$. Rule confidence is the percentage of tweets with X that also contain Y, and describes the strength of the implication, i.e., $conf(R) = \frac{supp(X \cup Y)}{sup(X)}$. 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 [2] may be used, rather than/beyond the confidence index, to measure the (symmetric) correlation between sets X and Y. The lift index is defined as the ratio $lift(R) = \frac{conf(R)}{sup(Y)}$. 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.

Consider the example tweets in Figure 2.10, the association rules $\{closingceremony \rightarrow fireworks\}$ and $\{amazing \rightarrow athletics\}$ model correlations among representative words in the two tweets. They allow us to point out in a compact form the representative information characterizing the two messages. In this work, to mine association rules representing strong word correlations, rules with high confidence value and lift grater than one have been selected.

2.2.5 Experimental results

This section presents and discusses the results obtained when analysing two real collections of twitter messages with the proposed framework.

2.2.5.1 Datasets

We evaluated the usefulness and applicability of the proposed approach on two real datasets retrieved from Twitter (http://twitter.com). Our framework exploits a crawler to access the Twitter global stream efficiently. To generate the real Twitter datasets we monitored the public stream endpoint offered by the Twitter APIs over a 1-month time period and tracked a selection of keywords ranging over two different topics, i.e., Sport and Music. The crawler establishes and maintains a continuous connection with the stream endpoint to collect and store Twitter data.

For both Twitter data collections, we analyzed the most frequent hashtags to discover trending topics. Among them, we selected the following two reference datasets for our experimental evaluation: the *paralympics* and the *concert* datasets. The *paralympics* dataset contains tweets on the Paralympic Games that took place in London in year 2012. The *concert* dataset contains tweets on the Madonna's concert held in September 6, 2012, at the Yankee Stadium located at The Bronx in New York City. Madonna is an American singer-songwriter and this concert was part of the "Mdna 2012 World Tour". Tweets in each dataset are preprocessed as described in Section 2.2.2. Hashtags used for tweets selection have been removed from the corresponding dataset, because appearing in all its tweets.

The main characteristics of the two datasets are as follows. The paralympics dataset contains 1,696 tweets with average length 6.89. The concert dataset contains 2,960 tweets with average length 6.38. Datasets used in the experiments are available at [43].

2.2.5.2 Evaluation set up and parameter configuration

The procedures for data transformation and cluster evaluation have been developed in the Java programming language. These procedures transform the tweet collection into the VSM representation using the TF-IDF scheme and compute the silhouette values for the cluster set provided by the cluster analysis. The DBSCAN [21] and FPGrowth [29] algorithms available in the RapidMiner toolkit [26] have been used for the cluster analysis and association rule extraction, respectively.

To select the number of iterations for the multiple-level clustering strategy and the DBSCAN parameters for each level, we addressed the following issues. We aim at avoiding clusters including few tweets, to discover representative clusters, and at limiting the number of unclustered tweets, to consider all posted information. For both datasets we adopted a three-level clustering approach, with each level focusing on a different

dataset part. The *Eps* and $MinPts$ values at each iteration level for the two datasets are reported in Section 2.2.5.3.

To extract association rules representing strong correlations among words appearing in tweets contained in each cluster, we considered a minimum confidence threshold greater than or equal to 80%, lift greater than 1, and a minimum support threshold greater than or equal to 10%.

2.2.5.3 Analysis of the clustering results

Starting from a collection of Twitter data related to an event, the proposed framework allows the discovery of a set of clusters containing similar tweets. The multiple-level DBSCAN approach, iterated for three levels, computed clusters progressively containing longer tweets, that (i) describe the event through a more varied vocabulary, (ii) focus on some specific aspects of the event, or (ii) report user emotions and thoughts associated with the event.

First-level clusters contain tweets mainly describing general aspects of the event. Second-level clusters collect more diversified tweets that describe some specific aspects of the event or express user opinions about the event. Tweets become progressively longer and more focused in third-level clusters, indicating that some additionally specific aspects have been addressed. Since at each level clusters contain more specific messages, a lower number of tweets are contained in each cluster and the cluster size tends to reduce progressively. By further applying the DBSCAN algorithm on the subsequent levels, fragmented groups of tweets can be identified. Clusters show good cohesion and separation as they are characterized by high silhouette values. Both the meaning and the importance of the information extracted from the two datasets has been validated with the support of news on the event available on the web.

Cluster properties are discussed in detail in the following subsections. Tables 2.12 and 2.13 report, for each first- and second-level cluster in the two datasets, the number of tweets, the average tweet length, the silhouette value, and the most representative words. Representative association rules are also reported, pointing out in a compact form the discriminative information characterizing each cluster. Clusters are named as C_{i_j} in the tables, where j denotes the level of the multiple-level DBSCAN approach providing the cluster and i locally identifies the cluster at each level j.

Tweet analysis in the paralympics dataset. First-level clusters can be partitioned into the following groups: clusters containing tweets that (i) post general information about the event (clusters C_{1_1} and C_{2_1}), (ii) regard a specific discipline (C_{3_1}) or team

(C_{4_1} and C_{5_1}) among those involved in the event, (iii) report user emotions (C_{6_1}), and (iv) talk about the closing ceremony (C_{7_1}).

Specifically, clusters C_{1_1} and C_{2_1} mainly contains information about the event location (rule $\{london\} \rightarrow \{stadium, olympics\}$). Clusters C_{4_1} is about the Great Britain team taking part in the Paralympics event (rule $\{teamgb\} \rightarrow \{olympic\}$). Clusters C_{3_1} and C_{6_1} focus on the athletics discipline. While cluster C_{3_1} simply associates athletics with the Olympic event, users in cluster C_{6_1} express their appreciation on the athletics competitions they are attending (rule $\{athletics\} \rightarrow \{amazing, day\}$). Finally, tweets in cluster C_{7_1} talk about the seats of people attending the final ceremony (rule $\{closingceremony, stadium\} \rightarrow \{seats\}$).

Second-level clusters contain more diversified tweets. The following categories of clusters can be identified: clusters with tweets posting information on (i) specific events in the closing ceremony (clusters C_{1_2} and C_{2_2}), (ii) specific teams (cluster C_{3_2}) or competitions (cluster C_{4_2}) in Paralympics, and (iii) thoughts of people attending Paralympics (cluster C_{5_2}).

More in detail, cluster C_{1_2} focuses on the flame that was put out on the day of the closing celebration (rule $\{stadium, london\} \rightarrow \{flame, closingceremony\}$), while cluster C_{2_2} is on the fireworks that lit up London's Olympic stadium in the closing ceremony (rule $\{stadium, closingceremony\} \rightarrow \{fireworks\}$). Cluster C_{3_2} is about the Great Britain team taking part to athletics discipline (rule $\{teamgb, park\} \rightarrow \{athletics\}$). Tweets in cluster C_{4_2} address the final basketball competition in the North Greenwich Arena. They contain the information about the event location and the German women's team involved in the competition (rules $\{final\} \rightarrow \{north, germany\}$ and $\{final\} \rightarrow \{basketball, germany\}$). Tweets in cluster C_{5_2} show an enthusiastic feeling on Paralympics (rule $\{stadium, olympic\} \rightarrow \{london, fantasticfriday\}$) and the desire to share pictures on them (rule $\{pic, dreams\} \rightarrow \{stadium, time\}$).

Third-level clusters (with DBSCAN parameters $MinPts = 15$, $Eps = 0.65$) show a similar trend to second-level clusters. For example, clusters contain tweets on some specific aspects of the closing ceremony, as the participation of the ColdPlay band (rule $\{london\} \rightarrow \{coldplay, watching\}$), or tweets about a positive feeling on the Paralympics event (rules $\{love\} \rightarrow \{summer, olympics\}$ and $\{gorgeous\} \rightarrow \{day\}$). By stopping the multiple-level DBSCAN approach at this level, 808 tweets labeled as outliers remain unclustered, with respect to the initial collection of 1,696 tweets.

Tweet analysis in the concert dataset. Among first-level clusters, we can identify groups of tweets mainly posting information on the concert location (clusters C_{1_1}, C_{2_1}, and C_{3_1} with rule $\{concert, mdna\} \rightarrow \{yankee\}$). The remaining clusters talk

TABLE 2.12: First- and second-level clusters in the paralympics dataset (DBSCAN parameters $MinPts$=30, Eps=0.39 and $MinPts$=25, Eps=0.49 for first- and second-level iterations, respectively)

First-level clusters

Cluster	Tweets	Avg Length	Avg Sil	Words	Association Rules
C_{1_1}	70	3	1	olympic, stadium	olympic→ stadium
C_{2_1}	30	7.33	0.773	olympics, london, stadium	london→ stadium, olympics
C_{3_1}	124	4.47	0.603	london, park, athletics, day	london, day→ athletics olympic→ park, athletics
C_{4_1}	30	6.67	0.710	heats, teamgb, olympic	teamgb→ olympic heats→ teamgb
C_{5_1}	30	5.67	0.806	mens, olympic, stadium	mens→ olympic
C_{6_1}	40	6	0.620	day, pic, amazing, athletics	athletics→ amazing, day day, pic→ stadium
C_{7_1}	36	5.72	0.804	closingceremony, seats, park, stadium	closingceremony, stadium→ seats olympic, park→ closingceremony

Second-level clusters

Cluster	Tweets	Avg Length	Avg Sil	Words	Association Rules
C_{1_2}	90	5.67	0.398	flame, closingceremony, london, stadium	stadium,london→ flame,closingceremony
C_{2_2}	36	6.67	0.616	fireworks, closingceremony, hart, stadium	stadium,closingceremony→ fireworks fireworks, hart→ stadium
C_{3_2}	26	6.08	0.722	teamgb, athletics, park, olympic, london	teamgb, park→ olympic teamgb, park→ athletics olympic, park→ teamgb, london
C_{4_2}	34	9.65	0.502	greenwich, north, arena, basketball germany, final, womens	final→ north, germany final→ basketball, germany final→ womens, germany
C_{5_2}	40	6.5	0.670	fantasticfriday, dreams, time, pic olympic, london, stadium	pic, dreams→ stadium,time stadium,olympic→ london, fantasticfriday

about some aspects of the concert. For example, cluster C_{4_1} regards the opening act (rule $\{yankee, stadium\} \rightarrow \{opening, act\}$). Cluster C_{5_1} is on the participation of the Avicii singer (rule $\{wait\} \rightarrow \{yankee, avicii\}$), cluster C_{6_1} on the "forgive" writing on Madonna's back (rule $\{forgive\} \rightarrow \{stadium, nyc\}$), and cluster C_{7_1} is about the raining weather (rule $\{rain\} \rightarrow \{yankee, stadium\}$). Finally, cluster C_{8_1} regards people sharing concert pictures (rule $\{queen\} \rightarrow \{instagram\}$).

In second-level clusters, tweets focus on more specific aspects related to the concert. For example tweets in cluster C_{2_2} refer to Madonna with the "madge" nickname typically used by her fans (rule $\{singing\} \rightarrow \{stadium, madge\}$).

Similar to the paralympics dataset, also in the concert dataset third-level clusters (with DBSCAN parameter Eps=0.77 and $MinPts$=23) show a similar trend to second-level clusters. For example, clusters contain tweets regarding some particular songs. At this stage, 1660 tweets labeled as outliers remain unclustered, with respect to the initial collection of 2,960 tweets considered at the first level.

TABLE 2.13: First- and second- level clusters in the concert dataset (DBSCAN parameters $MinPts$=40, Eps=0.41 and $MinPts$=21, Eps=0.62 for the first- and second-level iterations, respectively)

First-level clusters					
Cluster	**Tweets**	**Avg Length**	**Avg Sil**	**Words**	**Association Rules**
C_{1_1}	148	5.05	0.817	concert, mdna, yankee, stadium	concert, yankee→ stadium concert, mdna→ yankee
C_{2_1}	340	4	1	bronx, yankee, stadium	yankee, stadium→ bronx
C_{3_1}	160	3	1	yankee, stadium	stadium→ yankee
C_{4_1}	40	6	0.950	opening, act, mdna, yankee, stadium	act→ opening yankee, stadium→ opening, act
C_{5_1}	60	6	0.779	avicii, wait, concert	wait→ yankee, avicii
C_{6_1}	84	6.19	0.794	forgive, nyc, mdna, stadium	forgive→ stadium, nyc
C_{7_1}	40	7	0.986	rain, yankee, stadium	rain→ yankee, stadium
C_{8_1}	40	6	0.751	queen, instagram, nyc	queen→ instagram
Second-level clusters					
Cluster	**Tweets**	**Avg Length**	**Avg Sil**	**Words**	**Association Rules**
C_{1_2}	60	6.67	0.523	raining, mdna, stop	raining→ mdna, stop
C_{2_2}	40	7	0.667	madge, dame, named, singing	singing→ stadium, madge madge, singing, named→ stadium, dame
C_{3_2}	44	7.64	0.535	surprise, brother, birthday, avicii, minute	yankee, stadium, surprise→ birthday
C_{4_2}	22	8.55	0.893	style, way, vip, row livingthedream	style→ vip, livingthedream

2.2.5.4 Performance evaluation

Experiments were performed on a 2.66 GHz Intel(R) Core(TM)2 Quad PC with 8 GB main memory running linux (kernel 3.2.0). The run time of DBScan at the first, second, and third level is respectively 2 min 9 sec, 1 min 9 sec, and 48 sec for the paralympics dataset, and 4 min 4 sec, 1 min 53 sec, and 47 sec for the concert dataset. The run time progressively reduces because less tweets are considered at each subsequent level. The time for association rule extraction is about 24 sec for the cluster set at each level.

2.3 Analysis of patient transfers in hospital admissions

Lean thinking was originally a production philosophy and quality system to organise complex production processes aimed at encouraging flow and reducing waste [44]. Lean production was first pioneered at the Toyota Corporation and it was later used in automotive, manufacturing and service industry. From the early 2000s, lean thinking has been eventually applied in health care organizations, e.g., process improvement in surgical clinic experience [45], hospital discharge planning process improvement [46], operating room efficiency improvement [47], and in Emergency department to improve patient flow [48] or to reduce length of stay [49]. However in these work, data mining techniques were rarely used together with the lean strategy in health-care. Hospitals adopting the lean philosophy are structured based on a *functional organisation*, where

units specialise in their own particular processes and facilities with similar functions are grouped together. Thus, all staff and appliances used in the treatment of illness are grouped together as a multiprofessional group that completes patient's care. Mainly, the lean-hospital approach allows the elimination (or at least the reduction) of unnecessary patient transfers and reduces the risk of delay while transferring the patient to another phase of care.

This section presents PATRAN (*PA*tient *TR*ansfer *AN*alysis), an exploratory data mining framework to analyse historical data about patient flows in a hospital. The goal is to mine useful insights to support an effective organization of hospital activities according to the lean strategy. Specifically, data analysis in PATRAN addresses the following issues. (A) Evaluate if the actual patient flows adhere (or not) with a reference functional hospital organization. Gather useful insights to (B) improve the reference functional organization and/or (C) properly shape a new one taking into account the actual patient flows. The reference functional organization can correspond to the one currently adopted, or plan to be possibly adopted, in the hospital.

In this study, *patient flow* is described as the wards visited by patients during hospital admissions. As an example case of lean-hospital organization, we considered the *area-based organization* recently adopted in Italian hospitals for managing the activities, corresponding to standard and short patient hospitalizations. Hospital activities are organized according to some predefined *functional areas*, each one containing a set of wards based on the provided patient assistance and hospital stay/recovery. This area-based hospital organization aims at optimizing the use of human resources and materials, and minimizing patient transfers. In italian hospitals this organization tends to gradually replace the standard structure based on separate and individual wards of specialistic disciplines.

In the proposed PATRAN framework, for data analysis hospital admissions are categorized into two main groups, named *intra-area* and *inter-area* patient flows. Intra-area flows are the admissions that cohere with the area-based organization, since patients visited only wards belonging to the *same functional area*. Instead, inter-area flows corresponds to the admissions not (completely) consistent with the area-based organization, since patients transferred between wards of *different functional areas*. While the analysis of intra-area flows can reveal the consistency of hospital admissions with the reference area-based organization, the analysis of inter-area flows can points out critical conditions in the lean-hospital structure.

The main components of the proposed PATRAN framework are shown in Figure 2.12. The hospital admission log data has been first collected and represented in a data format suitable for the subsequent data analysis phase. Then, the dataset is partitioned into

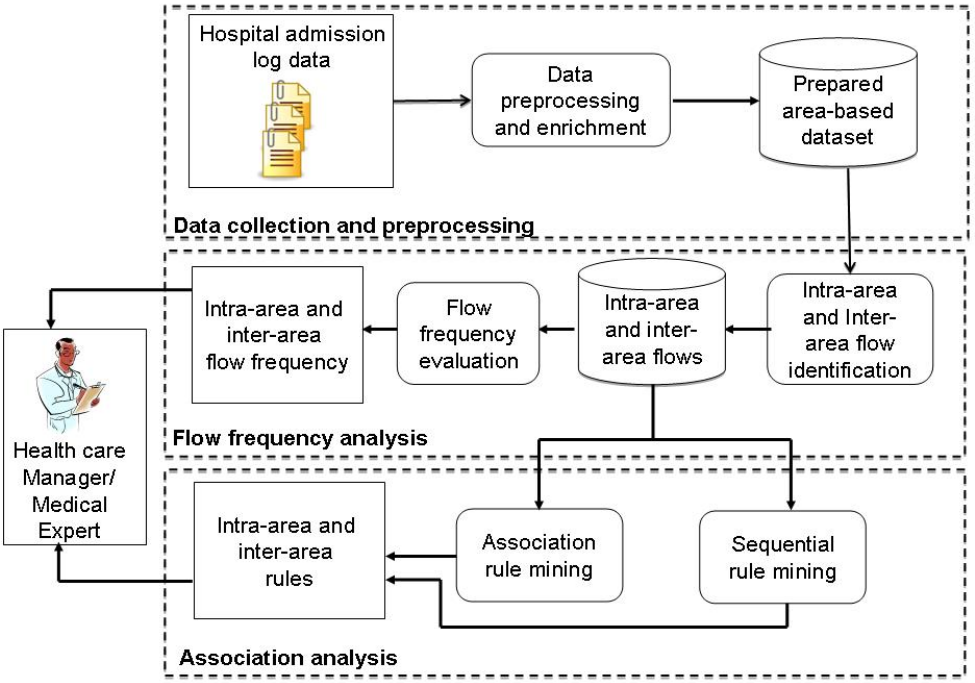


FIGURE 2.12: Framework to support the lean reorganization of hospitals

disjoint groups (named intra- and inter-area patient flows) based on the areas appearing in hospital admissions, with each group containing a subset of patients visiting different combination of functional areas. The frequency of these groups is evaluated to assess the degree of adherence of actual patient transfers with the area-based organization. Flows are then locally investigated through association analysis to discover underlying correlations among wards visited in hospital admissions. Both association and sequential rules are used for the analysis. To easily explore discovered rules, they have been categorized into two main classes (named intra- and inter-area rules) according to their represented information. Rules in the two categories provide different insights since they separately analyse correlations among wards in admissions that adhere or not with the area-based organization. A more thorough description of the main blocks is given in the following sections.

2.3.1 Related work

The application of data mining techniques to discover relevant knowledge from large amounts of medical records has become an appealing research topic. Association analysis has been widely used for discovering correlations among itemsets in medical applications, such as ischemic beats detection in electrocardiographic (ECG) recordings [50], breast cancer discovery [51], and heart attack prediction [10]. Frequent itemset mining, a type

of association analysis, has been applied to analyze patient flows/transfers in previous research works. In [52] frequent itemset search and lattice based classification are used to discover patient flows between hospitals within a healthcare network. Patients are treated as objects and the hospitals as attributes. By searching for frequent itemsets of hospitals sharing the same patients, significant flows of patient between hospitals can be achieved. works in [53][54][55][56], have applied frequent itemset mining to analyze the Taiwan's National Health Insurance claims databases. In [56], they analyzed the frequency and patterns of "one-stop visits" in Taiwan, which refers to a patient's visits to several specialties of the same healthcare facility in one day. By applying frequent itemsets mining, different combination patterns of specialties in one-stop visits have been computed and analyzed.

Research works also addressed the patient flow analysis using other data mining techniques. [57] reports the work in adopting the Markov Models and b-coloring based clustering approach for discovering a typology of clinical pathways in the French medical information system. [58] describes a novel approach employing time based clustering of health data for visualization and analysis of patient flow. [59] develops an integrated statistical data mining and simulation model for nurse activity which could be used to evaluate nurse-patient assignments. Classification trees are used to provide state transition probabilities to determine nurse movements. Then regression trees combined with kernel density are used to predict the amount of time a nurse spends in a location.

Unlike the previous works, in the research work on patient transfer analysis, lean strategy has been used together with the association analysis, through a preliminary instance of the MLDA framework by applying "functional area" based segmentation instead of clustering. With this approach, each segmented group containing a subset of patients visiting different combination of functional areas. Association analysis such as association rules and sequential rules have been analysed to extract correlations among wards within each group.

2.3.2 Data collection and preparation

Health care systems usually collect heterogeneous information about patients into log datasets. Hospital admission data consists of log files holding information about wards accessed by patients during hospital admissions. The PATRAN framework collects admission log files and properly prepare them to enable the subsequent data analysis phase. Mainly, at this stage the original log files are enriched with the information on the functional area where each ward appearing in the admission is located.

TABLE 2.14: Functional areas and corresponding wards

Functional area	**Wards**
Medical treatment (M)	General Medicine, Cardiology, Coronary Care Geriatrics, Infectious Diseases, Nephrology, Neurology, Psychiatry
Surgical treatment (S)	General Surgery, Maxillofacial Surgery, Vascular Surgery Ophthalmology, Otolaryngology, Orthopaedic Trauma, Urology
Maternal and child (MC)	Obstetrics-Gynecology, Daycare Centre, Neonatology, Pediatrics

TABLE 2.15: Example of hospital admission dataset

id	Area:Ward	Timestamp	id	Area:Ward	Timestamp
1	MC:Obstetrics-Gynecology	2009/04/27	3	M:General-Medicine	2009/04/19
1	S:General-Surgery	2009/04/16	3	M:Neurology	2009/05/03
1	M:General-Medicine	2009/04/01	4	M:Geriatrics	2009/04/09
2	S:General-Surgery	2009/04/07	4	M:General-Medicine	2009/04/28
2	MC:Obstetrics-Gynecology	2009/04/20			

An area-based hospital organization based on three functional areas, i.e., Medical treatment, Surgical treatment, and Maternal and Child area, has been selected as a reference example of area-based hospital organization in this study. The *Medical treatment* area contains wards devoted to medical diagnosis and disease cure, treatment and prevention (as Infectious diseases and Cardiology wards), while the *Surgical treatment* area contains wards involving surgical procedures (as General Surgery and Vascular Surgery wards). Wards in the *Maternal and child* area guarantee the health of pregnant women, mothers and children. The main wards included in each area are listed Table 2.14.

A toy example dataset with four hospital admissions is reported in Table 2.15. Each record contains the admission identifier (id) and the sequence of wards visited by the patient during the admission. The first ward in the sequence corresponds to the admission ward. For each ward, the functional area to which the ward belongs is also reported. In the example, the functional areas has been adopted for defining the ward location, i.e., M stands for Medical treatment, S for Surgical tratement, and MC for Maternal and child area.

2.3.3 Analysis of intra- and inter-area patient flows

To assess if actual patient transfers adhere with a given area-based hospital organization, PATRAN evaluates and compares the volumes of admissions with patient transfers (i) *all within one single area* and (ii) *crossing more areas*.

In the framework, hospital admissions are described in terms of *patient flow*, according to the *areas* visited by patients during the admission. PATRAN classifies admissions into two main categories, named *intra-area* and *inter-area patient flow*.

To support a more accurate evaluation, both intra- and inter-area flows are described in terms of the areas visited by patients during the hospital admission. Specifically, each intra-/inter-area flow is named with all the areas appearing in the admission. Each flow represents all admissions including all (and only) the areas naming the flow. In the example dataset, the intra-area flow [Medical], with admissions $id = 3$ and $id = 4$, is identified. Moreover, two inter-area flows are also available, i.e., [Maternal-Child, Surgical, Medical] and [Maternal-Child, Surgical], including admissions $id = 1$ and $id = 2$ respectively.

For gathering useful insights on the consistency of actual patient transfers with a given area-based organization, PATRAN evaluates the volume of both intra- and inter-area flows. The volume or *frequency* of an intra-/inter-area flow is given by the number of admissions in the flow. Specifically,

(i) The degree of adherence with a given area-based organization is evaluated in the framework based on the global frequency of all inter-area flows, also in contrast with the global frequency of all intra-area flows. The degree of adherence is higher when inter-area flows cover a limited subset of admissions, and they are (significantly) less frequent than the intra-area flows.

(ii) The analysis of inter-area flows can also reveal the *most critical areas* to be compliant with the area-based organization. These areas occur in many inter-area flows and globally appear in many admissions.

In the example dataset, both intra-area and inter-area flows globally cover $\frac{2}{4}$ admissions. The unique intra-area flow [Medical], with admissions $id = 3$ and $id = 4$, has frequency $\frac{2}{4}$. Each inter-area flow has frequency $\frac{1}{4}$. For example [Maternal-Child, Surgical] includes only admission $id = 2$ over 4 admissions in the dataset. The Surgical treatment area appears in both inter-area flows, and in all admissions included in these flows.

2.3.4 Association analysis

To deep the analysis of patient flows, assessing the volume of each flow is often not enough. Thus, in the framework each intra-/inter-area flow is locally further investigated using association analysis to discover underlying correlations among wards visited by patients during hospital admissions.

Two kinds of patterns are extracted from each flow, i.e., *association* and *sequential patterns*, represented in the form of *association* and *sequential rules*. These patterns provide complementary information that jointly allow describing different facets of patient flows. *Association patterns* capture the co-occurrence relationships among wards accessed in

TABLE 2.16: Transactional format of hospital admission data

Intra-/inter-area flow	id	ward set
[Medical, Surgical, Maternal-Child]	1	M:General-Medicine, S:General-Surgery, MC:Obstetrics-Gynecology
[Surgical, Maternal-Child]	2	S:General-Surgery, MC:Obstetrics-Gynecology
[Medical]	3	M:General-Medicine, M:Neurology
	4	M:Geriatrics, M:General-Medicine

hospital admissions but disregard the sequential information of the data. The extracted knowldege can be further characterized through *sequential patterns* taking into account the temporal precedence in visiting wards. They allow discovering temporal correlations among visited wards in hospital admissions. In the following sections, both patterns are formally defined (Sections 2.3.4.1 and 2.3.4.2), and their application in patient flow analysis is discussed (Section 2.3.4.3).

2.3.4.1 Association rule mining

To enable association rule extraction, each intra-/inter-area flow is tailored to the transactional data format. A *transactional hospital admission dataset* $\mathcal{D}_{trans}$ is a set of transactions in which each *transaction* consists of a set of features called *items*. Specifically, in our application scenario, each transaction corresponds to an hospital admission and items correspond to wards. This representation neglects the temporal order in accessing wards and ward repetitition in the hospital admission. Table 2.16 shows the transactional format for intra- and inter-area flows in the example dataset in Table 2.15.

Association rules described in Section 2.2.4 has been applied on the transactional dataset $\mathcal{D}_{trans}$. An *association rule* R is an implication in the form $R : X \rightarrow Y$, where X and Y are two disjoint ward sets. X and Y are also denoted as *antecedent* and *consequent* of the rule. In this study, each item set is described as a *ward set*, i.e., a set of wards accessed in the same hospital admissions. The interpretation of a rule $R : X \rightarrow Y$ is that if wards in X occur in an hospital admission, also wards in Y (tend to) occur in the same admission. *Support, confidence* and *lift* have been used to evaluate the rules. *Rule support* ($supp(R)$) is the percentage of hospital admissions in dataset $\mathcal{D}_{trans}$ containing both X and Y. *Rule confidence* ($conf(R)$) is the percentage of hospital admissions with X that also contain Y in dataset $\mathcal{D}_{trans}$. The lift index for a rule $R : X \rightarrow Y$ is the ratio $lift(R) = \frac{conf(R)}{sup(Y)}$. Lift values below 1 show a negative correlation between sets X and Y, whereas values above 1 indicate a positive correlation.

For instance, rule R_i:{M:Neurology} $\rightarrow$ {M:General-Medicine} (supp=50%, conf=100%) from the intra-area flow [Medical] in Table 2.16 indicates that Neurology co-occurs with

TABLE 2.17: Sequential format of hospital admission data

Intra-/inter-area flow	id	Ward sequence
[Medical, Surgical, Maternal-Child]	1	<M:General-Medicine><S:General-Surgery><MC:Obstetrics-Gynecology>
[Surgical, Maternal-Child]	2	<S:General-Surgery><MC:Obstetrics-Gynecology>
[Medical]	3	<M:General-Medicine><M:Neurology>
	4	<M:Geriatrics><M:General-Medicine>

General Medicine in $\frac{1}{2}$ of the transactions in the analysed flow (*id* 3). Since the implication holds in 100% of cases, rule R_i highlights a strong correlation between the two wards, because *all patients* having visited Neurology also visited General Medicine. Note that the confidence index is an asymmetric measure since the value might not been preserved when inverting rule antecedent and consequent. For example rule R_j:{M:General-Medicine}→{M:Neurology}, obtained by inverting terms in rule R_i, has the same support than R_i (50%), but the confidence is 50% (instead of 100%) because only 50% of patients who visited General Medicine also accessed Neurology.

2.3.4.2 Sequential rule mining

To take into account in rule extraction the temporal precedence in visiting wards, each intra-/inter-area flow is tailored to the *sequence database Dseq* format. *Dseq* is a set of sequences, and an ordered list of items occurs in these sequences. In this study, each sequence corresponds to an hospital admission. Items correspond to wards and they occur in a sequence following the appereance order in the admission (i.e, the order in which they have been visited by the patient during the admission). Table 2.17 shows the sequence database for patient flows in the example dataset in Table 2.15.

A *sequential rule* $R^s :< X >\rightarrow< Y >$ on a sequence database *Dseq* is defined as the relationship between two subsequences $< X >$ and $< Y >$, each one including an ordered list of wards. The interpretation of rule R^s is that if subsequence of wards $< X >$ occurs in an hospital admission, then subsequence of wards $< Y >$ will occur afterward in the same admission.

Some interestingness measures are defined for sequential rules, which are similar to those used in association rule mining and presented in Section 2.3.4.1. The *support* of rule $R^s :< X >\rightarrow< Y >$ in *Dseq* is the fraction of sequences in *Dseq* including $< X >$ followed (afterward) by $< Y >$. The *confidence* of rule R^s is the fraction of sequences in *Dseq* including $< X >$ in which $< X >$ is followed (afterward) by $< Y >$. Similarly also the lift measure could be adapted for sequential rules. For example, in the [Medical] flow, rule $R^s :< M : General - Medicine >\rightarrow< M : Neurology >$ has support $\frac{1}{2}$, since the two wards co-occur in one admission (*id*=3) over two. Confidence is $\frac{1}{2}$ because

General-Medicine appears in two admissions (id=3, id=4), but only in one them (id=3) it is followed by Neurology.

2.3.4.3 Using rules for patient flow analysis

Rules mined through the PATRAN framework are explored by domain experts for discovering valuable information to assess and/or improve the area-based hospital organization.

Association rules capture correlations among wards appearing in the same hospital admissions, neglecting temporal details about the temporal order used to visit wards. These rules can provide a valuable support in answering questions as *"What is the subset of wards X which appears in hospital admissions including the set of ward Y?"* Sequential rules taking into account also the temporal order in visiting wards, can answer questions as *"What is the subsequence of wards $< X >$ which follows in hospital admissions the subsequence of wards $< Y >$?"* In both questions above, sets (resp. subsequences) X and Y are the antecedent and consequent of an association (resp. sequential) rule. In the framework rules are automatically generated and filtered through quality indexes. Selected rules representing the most relevant correlations can be finally evaluated by domain experts.

To profitably support this evaluation process, PATRAN partitions rules into two representative categories, named intra-area and inter-area rules, which are described in the following. These categories are used to classify both association and sequential rules.

Intra-area rules model correlations among wards that (i) co-occur in hospital admissions (ii) and belong to the *same area*. These rules are extracted from intra-area flows, and represent implications where wards in their antecedent and consequent are all included in the same area. For example rule $R : \{M : Neurology\} \rightarrow \{M : GeneralMedicine\}$ is an intra-area rule, since both *Neurology* and *General Medicine* are in the Medical treatment area (M). Rules in this group allows domain experts to analyse flows that adhere with the area-based organization, and get useful insights for futher improving processes and structures within the area. For example, strong correlations among subsets of wards can reveal the existance of a sort of sub-area within the functional area.

Inter-area rules indicate correlations among wards that (i) co-occur in hospital admissions (ii) but are included in *different areas*. For this rule category, wards in the rule antecedent are located in a different area than those in the rule consequent. These rules are extracted from inter-area flows. For example $R : \{S : Surgical\} \rightarrow \{M : GeneralMedicine\}$ is an inter-area rule mined from the [Surgical, Medical] inter-area

flow. *Surgical* and *General Medicine* wards appearing in the rule are localed in the Surgical (S) and Medical (M) treatment area, respectively. Domain experts can exploit this kind of rules for analysing flows that do not adhere with the area-based organization, and gathering useful information to increase the degree of adherence. For instance, a strong implication across two areas, as between *Surgical* and *General Medicine* wards in the example above, can reveal the need of replicating the wards in the two areas.

2.3.5 Experimental results

This section describes the experiments performed to assess the patient framework. As a reference case study we considered a real dataset of (anonymized) hospital admissions collected on a large Italian hospital from 2007 to 2013. Since our aim was analysing patient transfers between different wards, admissions with one single ward have been discarded in a preprocessing step. The resulting dataset contains 17982 admissions globally including 20 different visited wards. The average, minimum and maximum admission length, given by the number of wards visited in the admissions are 2.09, 1 and 6 respectively.

In this study, the functional area definition and ward categorization proposed in [60] has been considered as a reference for the lean hospital organization. Wards occurring in the considered dataset have been classified into four reference areas according to [60]. The resulting ward categorization reported in Table 2.14 has been used to functionally characterize the considered dataset. Specifically, in every admission, for each appearing ward the corresponding functional area is reported.

The RapidMiner toolkit has been exploited for association analysis in the PATRAN framework. RapidMiner [26] is an open-source platform providing different algorithms to support various data mining tasks. Specifically, the RapidMiner implementations of the FP-Growth [29] and GSP [61] state-of-the-art algorithms have been used respectively for association rule and sequential rule extraction. The support threshold was set to 0.1%, and the confidence threshold to 5%.

2.3.5.1 Evaluation of flow frequency

The prepared dataset has been partitioned into disjoint groups based on the areas accessed by patients during admissions. This process generates three intra-area patient flows and three inter-area patient flows (see Fig. 2.13).

Intra-area patient flows globally cover 59,47% of admissions, showing that the patient transfers are quite compliant with an hospital organization based on the three reference

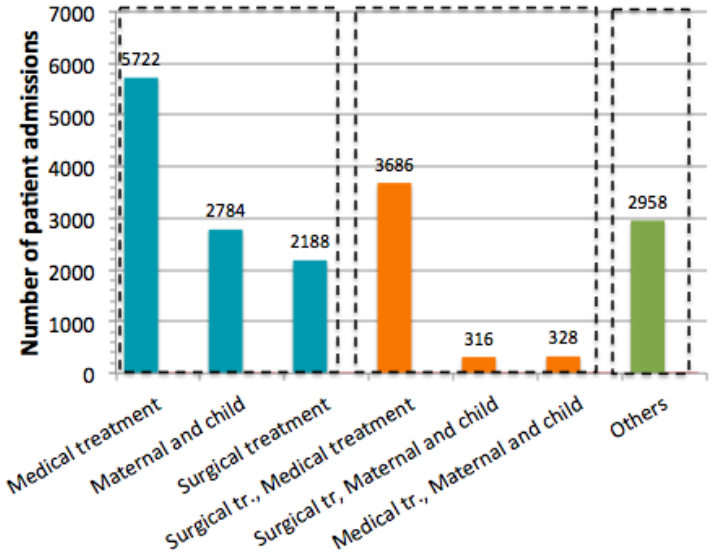


FIGURE 2.13: Distribution of hospital admissions in the functional areas

areas. The most frequent segment refers to patient flows within the Medical treatment area, followed by Surgical treatment and Maternal-and-child areas with almost the same cardinality.

Inter-patient flows mainly cross two functional areas (see Others in Fig. 2.13). Most of the flows contain patient transfers between the Medical and Surgical treatment areas, and some flows between the Surgical or Medical treatment area and the Maternal and child area.

2.3.5.2 Analysis of intra-area rules

This section presents the intra-area rules, reporting for every intra-area flow the most significant rules representing relevant correlations among wards in the same functional area. Rules with lift greater than 1 are considered as positive correlations. A selection of (the most relevant) correlations between wards are reported in Table 2.18.

Patient flow [Medical]. The analysis of hospital admissions points out the relevance of the General medicine ward in the Medical treatment area. These results are consistent with the services provided by General medicine, where patients are typically admitted before being transferred to more specialistic wards. A strong pairwise association emerges between General medicine and each of the other ward in the Medical treatment area (rules R_1-R_5), except Cardiology and Coronary care. Based on the rule confidence value, General medicine appears in most admissions including Geriatrics (80%, rule R_1) or Nephrology (76%, rule R_2), and in more than half admissions with Infectious disease,

Neurology, or Psychiatry ward (64%, rules R_3-R_5). Corncerning the Cardiology and Coronary care wards, according to rule analysis they constitute as a sub-area within the Medical treatment area. These wards are strongly correlated (rules R_6-R_7) and almost all patients who accessed one ward also accessed the other (rule confidence about 94%). Instead, they hold a negative correlation with any other ward in the area. A few other pairwise correlations emerge in the Medical treatment area, as between Neurology and Geriatrics (rule R_{10}), but these implications hold for a limited number of admissions (rule confidence below 30%).

Patient flow [Surgical]. Mostly weak implications hold between wards appearing in admissions included in this flow. The most relevant implications are between the three surgical wards (General, Vascular, and Maxillofacial surgery) in the Surgical treatment area and the other wards in the area. General surgery, devoted to surgical procedures especially on the abdominal cavity, thyroid and hernias, is positively correlated with Urology and Ortopaedic Trauma wards (rules R_9-R_{10}), but it is more frequently associated with the former than with the latter ward (rule confidence 64% and 38%). (Weak) pairwise correlation hold between Vascular surgery and Ophthalmology wards (rule R_{11}), and between Maxillofacial surgery and Otolaryngology (rule R_{12}). The Vascular surgery ward addresses problems on the vascular system through medical therapy, minimally-invasive catheter procedures, and surgical reconstruction, while the Maxillofacial surgery ward treats diseases, injuries and defects in head, neck, face and jaws. Based on rule confidence values, Ophthalmology appears in one third of admissions also including with Vascular surgery, and Otolaryngology in half of admissions having Maxillofacial surgery. Eventually, results point out a few other implications between wards in the Surgical treatment area, such as between Urology and Orthopaedic Trauma (rule R_{13}).

Patient flow [Maternal and child]. Two sub-areas emerge within this segment given respectively by Obstetrics ginecology and Pediatrics, and Daycare Center and Neonatology.

Obstetrics ginecology and Pediatrics are strongly positively correlated according to the rule lift value significantly greater than 1 (rules R_{14}-R_{15}). Based on the rule confidence value, Obstetrics ginecology is more frequently associated with Pediatrics in hospital admissions (37.5%) than Pediatrics with Obstetrics ginecology (3%). This result is coherent with the medical care provided by the two wards. Obstetrics-Gynecology provides assistance to women for pregnancy and for the diagnosis and treatment of the diseases of the female reproductive system. Pediatrics admitts children from the nursery to 16 years suffering of diseases that may affect different organs.

TABLE 2.18: Intra-area association rules

Patient flow	**Association rules**	**Sup. (%)**	**Conf. (%)**	**Lift**
[Medical]	R_1:{Geriatrics} → {General medicine}	14.93	80.55	2.05
	R_2:{Nephrology} → {General medicine}	2.43	76.06	1.93
	R_3:{Neurology} → {General medicine}	6.02	64.01	1.63
	R_4:{Infectious diseases} → {General medicine}	2.40	65.58	1.67
	R_5:{Psychiatry} → {General medicine}	0.49	61.70	1.57
	R_6:{Coronary care} → {Cardiology}	59.54	94.70	1.52
	R_7:{Cardiology} → {Coronary care}	59.54	95.47	1.52
	R_8:{Neurology} → {Geriatrics}	2.64	28.03	1.51
[Surgical]	R_9:{Urology} → {General surgery}	7.23	64.52	2.07
	R_{10}:{Orthopaedic trauma} → {General surgery}	6.33	37.94	1.21
	R_{11}:{Vascular surgery} → {Ophthalmology}	5.65	34.53	1.03
	R_{12}:{Maxillofacial surgery} → {Otolaryngology}	19.03	50.72	0.99
	R_{13}:{Urology} → {Orthopaedic trauma}	3.03	27.02	1.62
[Maternal & child]	R_{14}:{Obstetrics-Gynecology} → {Pediatrics}	0.10	37.5	14.12
	R_{15}:{Pediatrics} → {Obstetrics-Gynecology}	0.10	3.9	14.12
	R_{16}: {Daycare center → {Neonatology}	96.93	97.43	1.004
	R_{17}: {Neonatology} → {Daycare center}	96.93	99.89	1.004
	R_{18}: {Pediatrics} → {Daycare center}	2.53	88.31	0.89
	R_{19}:{Pediatrics} → {Neonatology}	0.17	6.49	0.07

Concerning Neonatology and Daycare Center wards, they co-occur in almost all the admissions in the segment, and most patients who have accessed one ward also accessed the other ward (rule confidence and support 99%). This result is reliable because Neonatology addresses the care of neonatal diseases and premature births, while in the Daycare Center infants spend a short period for adaptation to extra-uterine life.

2.3.5.3 Analysis of inter-area patient transfer

This section presents the inter-area rules, reporting for every inter-area flow the most significant rules representing relevant correlations among wards in different functional areas. Rules with lift greater than 1 are considered as positive correlations. A selection of (the most relevant) correlations between wards are reported in Table 2.19.

Patient flow [Medical,Surgical]. Various positive pairwise correlations emerge between wards located in the Medical and Surgical treatment areas.

The General medicine ward (in Medical treatments area), appears to be positively correlated with most wards in the Surgical treatment area as General surgery, Maxillofacial surgery, Ophthalmology, and Otolaryngology (rules R_1-R_4). General medicine frequently occurs in admissions including these wards (rule confidence 75%-80%), and globally it appears in about 62% of the admissions in the [Medical,Surgical] patient flow. These results point out that General medicine is probably often the access ward also for patients than are then admitted to wards of the Surgical treatment area.

Results also indicate the positive pairwise correlation between the three surgical wards in the Surgical treatment area (General, Vascular, and Maxillofacial surgery) and various wards in the Medical treatment area. The most relevant rules hold for the General surgery ward (rules R_5-R_8), that quite frequently appears in admissions also including Cardiology, Infectious disease, Psychiatry, and General medicine wards (rule confidence about 30%). For the other two surgical wards, rules highlight associations characterized by high lift values but holding on a limited number of admissions. Specifically, Vascular surgery is correlated with Cardiology and Coronary care wards (R_9-R_{10}). Similarly, Neurology is correlated with Maxillofacial surgey, and it is included in about 15% of admissions also including Maxillofacial surgery (rule R_{11}).

About associations between other wards of the two areas, a strong implication holds between Urology and Nephrology wards (in the Medical and Surgical treatment area respectively), both dealing with the excretory system (rules R_{12}-R_{13}). Almost all admissions including Nephrology also contains Urology (rule confidence 90%), wheares half of admissions with Urology contain Nephrology (confidence 56%). This result appears to be consistent with services provided by the two wards. While Nephrology addresses diagnosis, prevention and treatment of renal diseases, Urology deals with kidneys, ureters, and bladder from a surgical point of view. Other implications refer to Geriatrics and Orthopaedic Trauma, Infectious disease and Orthopaedic Trauma, and Neurology and Otolaryngology) (rules R_{14}-R_{16}, with confidence 24%-38%).

Patient flow [Medical treatment, Maternal and child]. From patients who cross the two areas, rule analysis mainly points out implications between Obstetric-Gynecology (in Maternal and Child area) and some wards in the Medical treatment area as General Medicine, Geriatrics, Coronary care and Cardiology.

The most relevant implication is between Obstetric-Gynecology and General Medicine (rules R_{17}-R_{18}). Most admissions in the segment includes the two wards (rule support 80%). Moreover, all patients admitted to General Medicine are also admitted to Obstetric-Gynecology (rule confidence 100%), and most patients admitted to Obstetric-Gynecology are also admitted to General Medicine (rule confidence 80%).

Obstetric-Gynecology and Geriatrics appear in about 6% of admissions (rules R_{19}-R_{20}). It is worth notice that all admissions with Geriatrics also contain Obstetric-Gynecology, while a limited subset of admissions with Obstetric-Gynecology include Geriatrics (about 6%).

Eventually, another strong implication emerge among Obstetrics-gynecology, Coronary care and Cardiology (rules R_{21}-R_{22}). Specifically, the rule lift value significantly higher

than 1 denote that the observed co-occurence frequency of these wards is (much) higher than the expected one.

Patient flow [Surgical treatment, Maternal and child]. The analysis reveals that these inter-area admissions mainly refers to two wards in the Maternal and Child area (i.e., Obstetric-Gynecology and Pediatrics) and some wards in the Surgival treatment area devoted to surgical procedures (Vascular Surgery, General Surgery) and to some other surgical operations.

More in detail, rules indicate a positive pairwise correlation between Obstetric-Gynecology and wards devoted to surgical procedures as Vascular surgery and General surgery (rules R_{26}-R_{30}). Among them, the implication between Obstetric-Gynecology and General surgery is particularly relevant because it holds for about half of the admissions in the segment. Moreover, rule confidence values show that Obstetric-Gynecology occurs in most admissions including General surgery (rule confidence 80%), and General surgery appears in 68% of admissions with Obstetric-Gynecology.

Concerning the Pediatrics ward, it results to be correlated with Opthalmology, Orthopaedic Trauma, Otolaryngology, and Maxillofacial surgery wards. Specifically, Pediatrics occurs in half of admissions with Orthopaedic Trauma, and the two wards together recurr in 17% of admissions in the segment. Instead, rules including other wards associated with Pediatrics are usually characterized by lower supporrt values.

2.3.5.4 Sequential rules

Ward correlations discovered using association rule analysis have been further investigated using sequential rules. These patterns have been exploited to discover the temporal order in which the wards are visited by patients. The dataset is inherently sparse, but considering the constraint of temporal order it becomes even more difficult to discover interesting sequential rules. And the discovered rules are usually characterized by support with fairly low confidence and low lift values. Some interesting sequential rules are reported in Table 2.20 and described in the following.

In general, the appeared sequential rules reflect the fact that the corresponding correlations also exist in association rules which ignore the temporal orders. Normally in the correlation between A and B, if rule A $\rightarrow$ B appears with very high confidence value, then the more interesting corresponding sequential rules should be <A> $\rightarrow$ <B>. For example, sequential rule R_1^s in Table 2.20 shows correlation between wards in Surgical treatment. It reports a sequential rule consistent with the association rule {Urology} $\rightarrow$ {General surgery} shown in Table 2.18. With 66% probability the patients visit General

TABLE 2.19: Support and confidence of inter-area association rules (2007 - 2013)

Functional areas	**Association rules**	**Sup.(%)**	**Conf.(%**
[Medical, Surgical]	R_1: {S:General surgery} → {M:General medicine}	18.05	75.68
	R_2: {M:General medicine} → {S:General surgery}	18.05	28.93
	R_3: {S:Maxillofacial Surgery} → {M:General medicine}	6.45	80
	R_4: {S:Ophthalmology} → {M:General medicine}	3.25	78.06
	R_5: {S:Otolaryngology} → {M:General medicine}	12.52	76.39
	R_6: {M:Cardiology} → {S:General surgery}	0.56	31.82
	R_7: {M:Infectious disease} → {S:General surgery}	0.97	30.25
	R_8: {M:Psychiatry} → {S:General surgery}	0.19	38.89
	R_9: {M:Cardiology} → {S:Vascular Surgery}	0.30	16.67
	R_{10}: {M:Coronary care} → {S:Vascular Surgery}	0.40	20.00
	R_{11}: {S:Maxillofacial Surgery} → {M:Neurology}	1.26	15.67
	R_12: {M:Nephrology} → {S:Urology}	17.38	90.36
	R_{13}: {S:Urology} → {M:Nephrology}	17.38	56.70
	R_{14}: {M:Geriatrics} → {S:Orthopaedic trauma}	3.17	38.56
	R_{15}: {M:Infectious Diseases} → {S:Orthopaedic trauma}	1.02	31.93
	R_{16}: {M:Neurology} → {S:Otolaryngology}	2.69	24.88
[Medical, Maternal & child]	R_{17}: {M:General Medicine} → {MC:Obstetrics-Gynecology}	80.30	100
	R_{18}: {MC:Obstetrics-Gynecology} → {M:General Medicine}	80.30	82.55
	R_{19}: {M:Geriatrics} → {MC:Obstetrics-Gynecology}	6.06	100
	R_{20}: {MC:Obstetrics-Gynecology} → {M:Geriatrics}	6.06	6.23
	R_{21}: {MC:Obstetrics-Gynecology, M:Cardiology} → {M:Coronary Care}	1.21	66.67
	R_{22}: {MC:Obstetrics-Gynecology, M:Coronary Care} → {M:Cardiology}	1.21	57.14
[Surgical, Maternal & child]	R_{23}: {S:Vascular Surgery} → {MC:Obstetrics-Gynecology}	1.26	100
	R_{24}: { S:General surgery} → {MC:Obstetrics-Gynecology}	46.06	89.57
	R_{25}: {MC:Obstetrics-Gynecology} → {S:General surgery}	46.06	68.54
	R_{26}: {S:Maxillofacial Surgery} → {MC:Pediatrics}	2.52	80.00
	R_{27}: {S:Ophthalmology} → {MC:Pediatrics}	1.89	75.00
	R_{28}: {S:Otolaryngology} → {MC:Pediatrics}	2.84	64.29
	R_{29}: {S:Orthopaedic trauma} → {MC:Pediatrics}	17.98	54.81
	R_{30}: {S:Urology} → {MC:Pediatrics}	2.84	40.91

TABLE 2.20: Example of intra-area Sequential rules

Sequential rules	**Sup. (%)**	**Conf. (%)**	**Lift**
R_1^s:<S:Urology> → <S:General surgery>	6.87	61.29	1.96
R_2^s:<S:Urology> → <S:Orthopaedic trauma>	2.71	24.19	1.45
R_3^s:<M:Nephrology> → <M:General medicine>	1.65	51.6	1.31
R_4^s:<M:Neurology> → <M:General medicine>	3.96	42.13	1.07
R_5^s:<M:Coronary care> → <M:Cardiology>	57.82	91.97	1.47
R_6^s:<M:Neurology> → <M:Geriatrics>	2.64	28.03	1.51
R_7^s:<M:General medicine> → <M:Geriatrics>	14.79	37.6	2.03
R_8^s:<M:General medicine> → <M:Infectious diseases>	1.55	3.93	1.08
R_9^s:<MC:Daycare center> → <MC:Neonatology>	96.93	97.43	1.004

surgery after being admitted to Urology. Some other rules such as R_2^s - R_6^s appear also in a similar way.

However, some exceptions may occur. For example, among the association rules in Medical treatment in Table 2.18, R_1 highlighted the correlation {Geriatrics} → {General Medicine} (support 14.93%, confidence 80.55%). While the sequential rule <General medicine> → <Geriatrics> in Table 2.20 indicates that access to Geriatrics is preceded by the access to General Medicine.

For the association rules representing symmetric correlations between wards, sequential rules may help identify the temporal order in visiting wards. For example, association rules R_6 and R_7 in Table 2.18 represent symmetric correlations between Coronary Care and Cardiology, while sequential rule R_5^s shows that patients usually visit Cardiology after Coronary Care. There's also a similar symmetric correlation between Neonatology and Daycare center, where Neonatology is usually visited after Daycare center, even if the lift value (1.004) does not indicate a strong sequential order.

2.3.5.5 Execution time

Experiments were performed on a 2.8 GHz x2 Intel Pentium(R) 4 CPU PC with 2 GB memory. The average run time for extraction is about 1 s for each segment.

Chapter 3

Analysis of heterogeneous data with large cardinality

This chapter describes data mining techniques developed in this PhD study for the analysis of heterogeneous User-Generated Data (UGD) collections with large cardinality. Heterogeneous data is a common characteristic of datasets in various domains to model data under different facets. Heterogeneous data contains in the same dataset, data coming from different sources, in varying formats, and of different nature such as text, binary or categorical value, continuous value, spatial information and so on. Failing to take the heterogeneous issue into account can easily derail the discoveries from these data. As said by Vaupel & Yashin [62], "both theoretical and empirical research may be unnecessarily complicated by failure to recognize the effects of heterogeneity".

In this study we considered some reference examples of heterogeneous UGD coming from three different application domains, i.e., health care, social network and urban environment domains (see Figure 3.1).

In health care domain, when analyzing medical examinations, despite *patient treatments*, *patient profile* information such as patient age and patient gender can be also taken into account. To fully characterize the examination history of specific patients, these heterogeneous aspects should also be considered and properly treated due to the different properties of the patient they represent.

In social network domain, Twitter data has recently been considered to perform a large variety of advanced analysis. On August 2009, Twitter released the location service that enables mobile users to publish their messages with geographical information of their location when posting the messages. Since then, heterogeneous kind of data are available in each tweet messages. Specifically, each tweet may include *textual content*, *temporal*

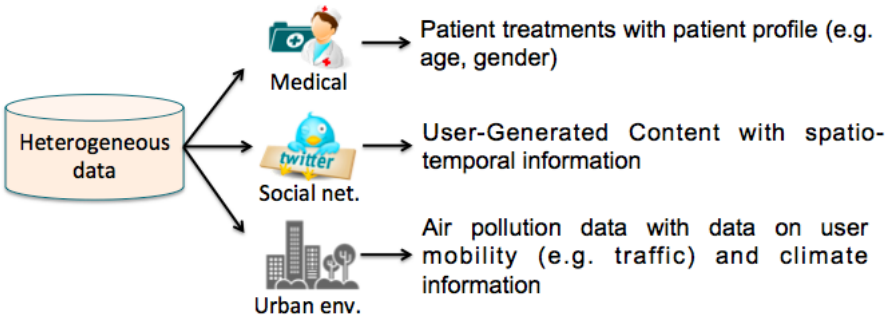


FIGURE 3.1: Heterogeneous data analysis

information on tweet publication time and *geographical location* describing where the tweet is posted. To gain interesting knowledge (e.g., topic and sentiment detection) from tweets, the rather heterogeneous dimensions should be considered in data analysis process. Since tweet textual content is limited to maximum 140 characters in length, the textual content is inherently sparse, while spatial and temporal information may be spread out over a large temporal and spatial window.

In the area of urban data, the analysis of the air quality data is continuously a relevant research issue due to its possible impact on the public health. The quality of the air can vary over time and across different areas of the same city. It is influenced by different factors such as weather conditions (e.g. humidity, temperature and atmospheric pressure) and User-Generated Data as human activities (e.g., traffic flows, people's mobility). When monitoring *pollutant concentrations* and their relationship with *traffic conditions* representing people's mobility, taking also into account the *meteorological conditions*, the various dimensions included in the data collection become rather heterogeneous. Innovative data analytics solutions able to acquire, integrate and analyze data containing very large amount of heterogeneous dimensions are needed.

To address the above issues, in the research activity, the following approaches have been proposed. As shown in Figure 3.2, *novel combined distance measures* taking into account all considered facets of the problem under analysis have been proposed and integrated into the clustering process in the MLDA framework described in Chapter 2. More specifically, to extract useful knowledge from patient treatments with profile information, a novel combined distance measure has been proposed to cluster patients based on the performed examinations and patient profiles (as patient age and gender). Based on the discovered cluster set, a classification model has been created to characterize the content of clusters and evaluate the robustness of the clustering process. A detailed description of this work has been summarized in [63].

To address the issues of heterogeneous User-Generated Data from Twitter, a novel spatio-temporal distance measure has been proposed to group twitter messages based on their

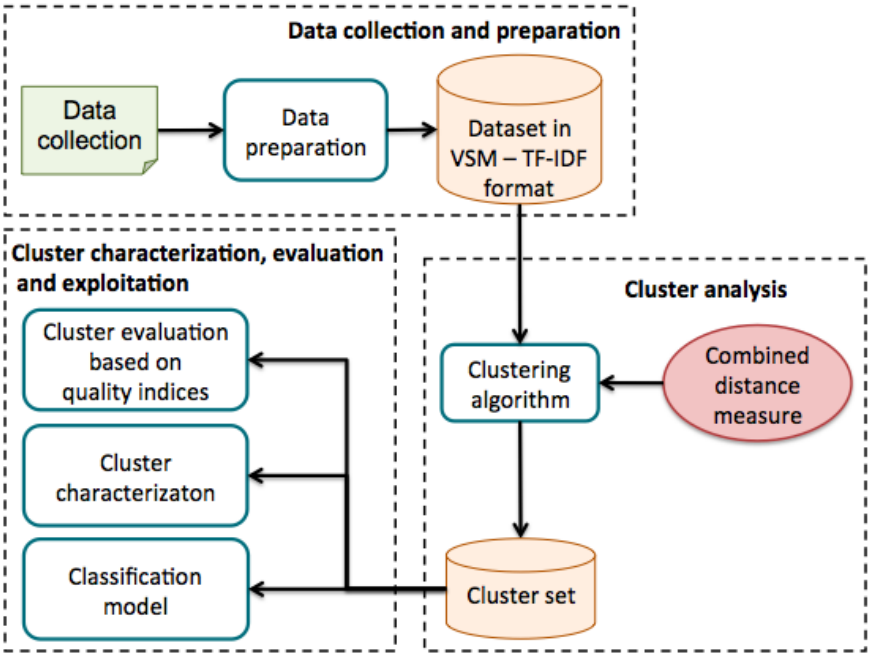


FIGURE 3.2: Heterogeneous data analysis

spatio-temporal features and textual content. Each computed cluster is then locally characterized through association rules to point out in a succinct form the relevant content of its messages. The approach allows discovering cohesive clusters of tweets that are not only similar in content but also close in space and time. The proposed methodology exploits the computational advantages of distributed frameworks of Apache Spark.

Nevertheless the MLDA framework is able to discover cohesive clusters and find interesting correlations characterizing each cluster, it may have limitation to find correlations in heterogeneous data. Because the different facets of heterogeneous data may significantly increase the data dimensionality and sparseness, which can enforce to discover, at some iteration levels, clusters with limited size, thus limit the amount of discovered correlations. To cope with this issue, *data taxonomy integrated with association analysis* has been presented for reducing data sparseness by climbing the abstraction level. More specifically, to monitor air pollutant concentrations and their relationship with traffic and climate conditions, the knowledge extraction process is driven by taxonomy to generalize low-level measurement values as the corresponding categories. The approach allows discovering correlations among heterogeneous data describing the urban environment at different abstraction levels, and the discovered knowledge is potentially useful for supporting city administrators in decision-making.

This chapter is organized as follows. Sections 3.1 and 3.2 present the new combined distance measures integrated in the MLDA framework to analyse patient treatments with

patient profile information and Twitter data containing textual, spatial and temporal information respectively. Section 3.3 describes the application of association analysis to extract correlations in multiple abstraction levels between air pollution data and traffic conditions representing people's mobilities.

3.1 Analysis of patient treatments with patient profile information

This section presents the application of the MLDA framework (see Chapter 2) to cluster patients not only undergoing similar medical treatments, but also sharing common patient profiles (i.e., patient age and gender). For clustering this heterogeneous patient data collections, a novel combined distance measure has been proposed taking into account both patient characteristics and patient examination history. Since among the different clustering techniques, density-based clustering techniques have been proven to outperform the other techniques such as prototype based techniques (see Chapter 2), a density-based clustering algorithm has been exploited in the MLDA clustering process, to focus on different dataset portions and locally identify groups of patients. Based on the computed cluster set, a classification model has been created to characterize the content of clusters and measure the effectiveness of the clustering process. As a case study, the proposed approach has been applied to the diabetic scenario, by considering a real dataset of (anonymized) diabetic patients. This research work has been presented and published in [63].

3.1.1 Related work

Differently from [6, 12, 14], the study in the research work aims at identifying groups of patients with similar conditions by analyzing both patient examination history and patient factual data (i.e., age, gender). While in [6] a multiple-level strategy is used for clustering patients according to their examinations, in this work a new combined measure has been defined for patient comparison, and the cluster analysis has been coupled with the classification process.

Patient profiles have been considered for various analysis, such as blood glucose prediction in diabetic patients [64], asthma attacks prediction [65], heart attacks prediction [66] and adverse drug events identification [67]. More specifically, in [65] the authors built two classifiers named Pattern Based Decision Tree and Pattern Based Class-Association Rules, based on patients' daily bio-signal records and environmental data to predict the chances of asthma attacks. In [66] various classification techniques such as Rule based,

Decision tree, Naïve Bayes and Artificial Neural Network have been applied to patient profile information (e.g., age, gender, blood pressure), to predict the likelihood of patients getting a heart disease. [67] exploited classification rules on patient profile such as age, gender and race to identify the presence of an adverse drug even. Differently from these work, in this section we applied the MLDA framework and proposed a new combined distance measure based on patient treatments and patient profile information.

3.1.2 Patient representation

Let $\mathcal{D}$ be a collection of patient records and $\Sigma = \{e_1, \ldots, e_k\}$ the set of examinations done by at least one patient in $\mathcal{D}$. Let p_i be an arbitrary patient in $\mathcal{D}$. Patient p_i is represented as a triplet $(a_{p_i}, g_{p_i}, E_{p_i})$ where a_{p_i} and g_{p_i} are patient age and gender, respectively, and E_{p_i} is the patient examination history. In the triplet representing patient p_i, age a_{p_i} is an integer number, and gender g_{p_i} is represented as a boolean value.

As addressed in Section 2.1, the patient examination history is tailored to the Vector Space Model (VSM) representation and the TF-IDF (Term Frequency (TF) - Inverse Document Frequency (IDF)) score is used to weight the relevance of examinations underwent by patients. E_{p_i} is a vector of $|\Sigma|$ cells in the examination space. Each vector element $E_{p_i}[e_j]$ corresponds to a different examination e_j done by patient p_i. $E_{p_i}[e_j]$ is the TF-IDF weight describing the relevance of examination e_j for patient p_i. When analysing the patient examination history, TF-IDF weights allow focusing on the examinations specific for each set of patients and discarding the examinations done by all the patients. Each vector element $E_{p_i}[e_j]$ reports the TF-IDF weighted frequency of examination e_j for patient p_i. It has a high value when examination e_j appears with high frequency in patient p_i and with a global low frequency in $\mathcal{D}$.

3.1.3 Patient clustering through a new distance measure

Clustering medical data requires dealing with some critical issues usually characterizing medical datasets. They can show variable data distributions due to various possible examinations for different disease severity, and clusters could be of arbitrary shapes. Besides, outliers can also be included as specific set of examinations for some disease conditions. Finally, since this study aims at discovering the examinations usually adopted for a given disease through an explorative data analysis, the expected number of clusters can be hardly guessed a priori. For these reasons, density based algorithms are suitable for the analysis.

In this study, the very effective density-based algorithm DBSCAN [21] has been selected for patient clustering in the MLDA framework. The *multiple-level clustering* approach in MLDA allows clustering datasets with a variable distribution by iteratively applying the DBSCAN algorithm on different (disjoint) dataset portions. The whole original dataset is clustered at the first level. Then, at each subsequent level, patients labeled as outliers in the previous level are re-clustered. The DBSCAN parameters Eps and $MinPts$ are properly set at each level.

Clustering patients requires the definition of a metric to evaluate the distance (or similarity) between patients based on the features describing them. Then, the quality of computed clusters is evaluated according to some indices. Sections 3.1.3.1 and 3.1.3.2 describe how these issues have been addressed in this study.

3.1.3.1 Computation of distance between patients

In this study a new distance measure has been defined to evaluate patient distance according to the three aspects characterizing patients, i.e., patient age, gender and examination history. Specifically, the distance between two arbitrary patients p_i and p_j in $\mathcal{D}$ is computed as follows:

$$d(p_i, p_j) = w_a d_a(a_{p_i}, a_{p_j}) + w_g d_g(g_{p_i}, g_{p_j}) + w_E d_E(E_{p_i}, E_{p_j}) \tag{3.1}$$

where $d_a(a_{p_i}, a_{p_j})$, $d_g(g_{p_i}, g_{p_j})$, and $d_E(E_{p_i}, E_{p_j})$ measure the distances between patients with respect to their age, gender, and examination histories, respectively. The w_a, w_g, and w_E parameters weight the relevance of the three contributions when comparing patients.

Different metrics have been used for computing distances in (3.1), based on the type of attributes describing patients. In the $(a_{p_i}, g_{p_i}, E_{p_i})$ triplet representing an arbitrary patient p_i, age a_{p_i} is an integer number, gender g_{p_i} a boolean value, and E_{p_i} is a vector of real numbers representing TF-IDF values. Consequently, the following distance metrics have been selected. We adopted the Euclidean metric [2] for evaluating the distance on the age attribute (i.e., $d_a(a_{p_i}, a_{p_j})$). The Hamming distance [2] has been used to check if the two patients have the same gender (i.e., $d_g(g_{p_i}, g_{p_j})$). The distance between two weighted examination frequency vectors (i.e., $d_E(E_{p_i}, E_{p_j})$) is evaluated using the cosine distance measure [2], which has often been used to compare documents in text mining [4]:

$$d_E(E_{p_i}, E_{p_j}) = \arccos\left(cos(E_{p_i}, E_{p_j})\right) \tag{3.2}$$

where $cos(E_{p_i}, E_{p_j})$ represents the cosine similarity between E_{p_i} and E_{p_j}, and is computed as

$$cos(E_{p_i}, E_{p_j}) = \frac{\sum_{1 \leq k \leq |\Sigma|} E_{p_i}[e_k] E_{p_j}[e_k]}{\sqrt{\sum_{1 \leq k \leq |\Sigma|} E_{p_i}[e_k]^2} \sqrt{\sum_{1 \leq k \leq |\Sigma|} E_{p_j}[e_k]^2}} \tag{3.3}$$

The cosine distance in (3.2) verifies the triangle inequality [2]. The cosine similarity $cos(E_{p_i}, E_{p_j})$ is in the range [0,1]. $cos(E_{p_i}, E_{p_j})$ equal to 1 describes the exact similarity of examination histories for patients p_i and p_j, while $cos(E_{p_i}, E_{p_j})$ equal to 0 points out that patients have complementary histories.

All the distances in (3.1) have been normalized in the range $[0, 1]$. Parameters w_a, w_g, w_E are in the range [0,1], and $w_a + w_g + w_E = 1$. Lower values of $d(p_i, p_j)$ denote a higher similarity between the two patients, and higher values of $d(p_i, p_j)$ denote a lower similarity.

3.1.3.2 Cluster evaluation

The discovered cluster set is evaluated using the Silhouette index [22]. Silhouette allows evaluating the appropriateness of the assignment of a data object to a cluster rather than to another by measuring both intra-cluster cohesion and inter-cluster separation (see Section 2.1.4).

Negative silhouette values represent wrong patient placements, while positive silhouette values a better patient assignments. Clusters with silhouette values in the range [0.51,0.70] and [0.71,1] respectively show that a reasonable and a strong structure have been found [20]. Distances between patients for silhouette evaluation have been computed as described in Section 3.1.3.1.

3.1.4 Patient classification

Clusters computed as described in Section 3.1.3 have been analyzed with the support of a domain expert to describe the cluster content from a medical perspective. Then, a class label has been assigned to each cluster, and the cluster set has been analyzed using classification techniques, as part of the MLDA framework.

Classification is the task of learning a classification model that maps each data object to one of the predefined class labels [2]. A classification model is typically used to predict the class label for a new unlabeled data object. Besides, it can also serve as descriptive model to explain what features characterize objects in each class. As described in Section 2.1.6.1, among various classification methods, decision tree classifiers have been selected in this study to analyze the result of the clustering process.

For the patient representation considered in this work, the process creating the decision tree considers tests on the following attributes: patient age, gender, and the examinations done by patients weighted through the TF-IDF weighting score. Decision trees have been previously applied in text mining to classify documents weighted through the TF-IDF scheme [24].

In the analysis, we considered *Gini index* impurity-based criterion to split the record set for growing the tree. The Gini index [2] measures how often a randomly chosen instance from the set would be incorrectly labeled if it were randomly labeled according to the distribution of labels in the subset.

Three different metrics have been used to evaluate the quality of constructed classification model, i.e., accuracy, precision and recall [2]. While the accuracy measures the overall quality of classifier, precision and recall analyse the performance of the classifier with respect to a given class.

3.1.5 Experimental results

This section discusses the results obtained when analyzing a real collection of diabetic patients with the proposed approach. The open source RapidMiner toolkit [26] has been used for the cluster and classification analysis. The procedures for data transformation into the VSM TF-IDF scheme and cluster evaluation have been developed in the Python programming language [68]. Experiments were performed on a 2.66-GHz Intel(R) Core(TM)2 Quad PC with 8 GBytes of main memory.

3.1.5.1 Dataset

The dataset considered in this study was collected by an Italian Local Health Center. Raw data contain examinations performed by 5,622 patients with overt diabetes in year 2007, and data on patient characteristics such as patient age and gender. Examinations contain both routine and more specific tests to analyze diabetes complications on various degrees of severity. The dataset includes both male and female patients (2,756 and 2,866 patients, respectively) in a wide age range (between 4 and 101 years). The diagnostic and therapeutic procedures are defined using the ICD 9-CM disease classification [25].

3.1.5.2 Clustering results

The multiple-level DBSCAN approach, iterated for two levels, generates the cluster set reported in Tables 3.2 and 3.3. Parameters weighting the relevance of the patient age,

TABLE 3.1: Patient conditions for discovered clusters

Cluster name	Patient condition
C_3, C_8, C_{15}, C_{17}	Diabetic patients with retinopathy
C_{10}, C_{11}	Diabetic patients with possible coronary artery disease
C_4, C_6	Diabetic patients with retinopathy and possible coronary artery disease
C_9, C_{12}, C_{16}	Diabetic patients not in compensation
C_{14}	Diabetic patients with carotid vascular damage
C_{13}, C_{20}	Diabetic patients with multiple organ damage (retinopathy, coronary heart disease and cardiovascular risk, renal and hepatic impairment)
Other clusters	Diabetic patients (probably) without complications

TABLE 3.2: Exam frequencies in first level clusters ($MinPts$=30, Eps=0.04, $w_a = 0.3$, $w_g = 0.05$, $w_E = 0.65$)

Category	Exam	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
Routine	Glucose level	99.9	100	85.2	89.7	91.4	92.3	92.9	90.9	62.5	50	77.7	71
	Urine test	100	100	85.2	89.7	91.4	92.3	92.9	90.9	53.1	50	77.7	64.5
	Venous blood	62.4	62	59.1	70.1	19.7	60.3	11.6	63.6	100	26.3	50	100
	History and assessment	100	99.7	86.4	89.7	78.4	92.3	78.6	92	57.3	57.9	80.5	64.5
	General visit	0.12	-	-	-	100	-	100	-	-	-	-	-
	Glycated hemoglobin	-	-	-	-	0.62	-	-	-	100	-	-	100
	Capillary blood	99	98.2	85.2	88.7	87.6	92.3	88.4	90.9	53.1	50	77.7	64.5
Cardiovascular	Electrocardiogram	-	-	-	100	-	100	0.89	-	-	100	100	-
Eye	Fundus oculi	-	-	100	100	1.23	100	-	100	-	-	-	-
Patients		827	602	88	97	162	78	112	88	96	38	36	31
Minimum age		22	28	59	48	53	53	44	57	57	61	59	64
Maximum age		101	94	88	88	98	85	90	85	96	81	82	82
Average age		71.76	70.2	72.9	68.8	75.5	70.8	68.8	72.2	75.5	70.4	71.7	73.8
Num of males		0	602	0	97	0	0	112	88	0	0	36	31
Num of females		827	0	88	0	162	78	0	0	96	38	0	0
Silhouette		0.73	0.77	0.70	0.78	0.87	0.76	0.90	0.74	0.82	0.84	0.82	0.90

gender and examination history for patient comparison (i.e., w_a, w_g, and w_E) have been set with the support of a clinical domain expert. DBSCAN parameters (i.e., $MinPts$ and Eps) have been selected to discover cohesive and well separated groups of patients with similar disease severity.

All discovered clusters show good cohesion and separation as they are characterized by high silhouette values. The clustering results were also evaluated by a domain expert to describe the cluster content from a medical perspective. Table 3.1 reports the list of patient conditions represented by each cluster as defined by the domain expert.

First-level clusters contain patients mainly undergoing routine tests to monitor diabetes conditions (for example clusters C_1 and C_2) or routine tests coupled with basic examinations to diagnose disease complications. Besides routine examinations, patients had

TABLE 3.3: Exam frequencies in second level clusters ($MinPts$=25, Eps=0.07, w_a = 0.3, $w_g = 0.05$, $w_E = 0.65$)

Category	Exam	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}	C_{18}	C_{19}	C_{20}
Routine	Glucose level	98.3	57.7	96.96	90.9	25.93	73.1	100	68.7
	Urine test	89.9	57.7	96.96	93.18	25.93	73.1	100	65.7
	Venous blood	100	38.5	27.27	100	14.81	53.8	81.81	100
	Complete blood count	9.7	-	-	-	-	-	-	-
	History and assessment	90.7	57.7	96.96	85.23	25.93	73.1	100	65.7
	General visit	2.95	1.92	100	100	-	100	-	13.4
	Glycated hemoglobin	99.2	5.77	-	100	-	3.85	3.03	98.5
	Capillary blood	89.9	57.7	96.96	90.9	25.93	73.1	100	65.7
Cardiovascular	Total cholesterol	98.7	-	-	-	-	-	-	100
	HDL cholesterol	100	-	-	-	-	-	-	100
	Triglycerides	98.7	-	-	-	-	-	-	100
	Electrocardiogram	65.4	5.77	100	5.68	-	100	100	20.9
	(Echo) Color Doppler of the supraaortic vessels	-	100	-	-	-	-	-	-
Liver	Alanine aminotransferase enzyme (ALT)	99.6	-	-	-	-	-	-	2.98
	Aspartate aminotransferase enzyme (AST)	99.2	-	-	-	-	-	-	1.49
	Gamma GT	2.11	-	-	-	-	-	-	100
Kidney	Microalbuminuria	59.1	-	-	-	-	-	-	100
	Creatinine	12.2	-	-	-	-	-	-	100
	Creatinine clearance	97.5	-	-	-	-	-	-	-
	Uric acid	97.9	-	-	-	-	-	-	-
	Culture urine	96.2	1.92	-	-	-	-	-	92.5
	Microscopic urine analysis	73.4	-	-	1.14	-	-	-	1.49
Eye	Fundus oculi	67.9	28.8	100	5.68	100	-	-	29.8
Patients		237	52	33	88	27	26	33	67
Minimum age		41	56	60	47	33	65	51	48
Maximum age		88	90	87	100	66	89	90	82
Average age		63.47	75.14	73.2	75.09	53.11	75.68	72.01	69.38
Num of males		148	35	23	39	23	3	4	25
Num of females		89	17	10	49	4	23	29	42
Silhouette		0.63	0.86	0.58	0.66	0.86	0.56	0.77	0.73

additional basic examinations to diagnose eye problem (C_3), risk for cardiovascular disease (C_{10}) or both of them (C_4, C_6). Each first-level cluster contains patients with the same gender.

Second-level clusters are more diversified. They contain patients tested using an increasing number of examinations to diagnose several diabetes complications. These patients can be seriously affected by a particular disease complication or by more than one disease complication at the same time. Moreover, each cluster contains patients of both genders, but showing similar examination histories. For example, clusters contain (both female and male) patients with cardiovascular complications (C_{18}) or with possible multiple organ damage (C_{13}, C_{20}).

By stopping the multiple-level approach after two iterations, 2,800 patients are labeled as outliers and remain unclustered. Note that these patients can be additionally clustered

by iterating the approach for more steps. In our analysis, only clusters with quite good silhouette are extracted out. By trading-off between cluster cohesion and separation, and number of unclustered patients, more clusters and less outliers can be obtained.

3.1.5.3 Classification results

Starting from the cluster set reported in Tables 3.2 and 3.3, a classification model was built to characterize patients contained in each cluster as well as to measure the quality of clusters. The decision tree method has been used to create the classification model.

To preserve the characteristics of the discovered clusters, where patients with similar disease severity and/or age and/or the same gender have been grouped together, each cluster has been labeled with a different class label. The cluster name C_i ($1 \leq i \leq 20$) has been used as the class label for each cluster. The 7-fold cross validation method has been adopted for evaluating the classification model.

In the resulting decision tree, each node represents a patient characteristic (the patient age, gender or one examination undergone by the patient), while each branch descending on a node represents a possible value, or a range of values, for that characteristic. For example, it can represent a range of values on the age attribute or on the TF-IDF weight associated with each examination. The computed decision tree contains 50 nodes, 51 paths with average length 7, and leaf nodes with quite good degree of purity. Due to space constraints, only a portion of the tree, specifically the upper part, is reported in Fig. 3.3.

Consider, as an example, the first two tree paths in Fig. 3.3. A patient with a higher weighted frequency of the HDL cholesterol examination (> 0.012) is likely to be a diabetic patient with multiple organ damage. She/He is labeled with class label C_{20} or C_{13} based on the weighted frequency of the Gamma GT examination. Instead, patients with lower weighted frequency of HDL cholesterol (≤ 0.012) are labeled by traversing the right-end side of the tree root.

The experimental result showed the goodness of the constructed model. The accuracy value is about 98.6%. The average recall value is around 98.5%, except for clusters C_{14} (69%) and C_{19} (76%), and the precision value has an average of 97%, apart from clusters C_{10} (81.4%) and C_{19} (86.2%). The values are all very high, which guarantees the quality of the classification model.

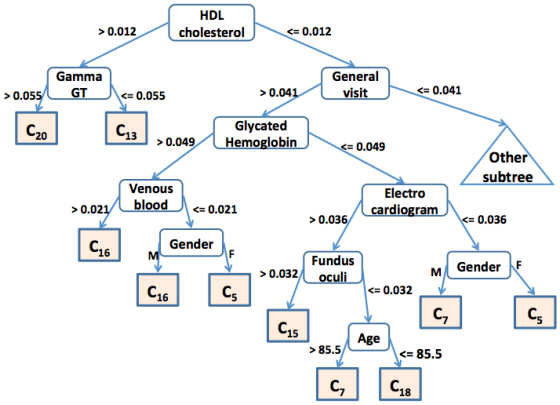


FIGURE 3.3: Portion of the classification tree

3.1.5.4 Execution time

For the multiple-level DBSCAN algorithm, the execution time is about 3min 34s, 3min 8s at the first and second level, respectively. The classification process to generate a decision tree lasts about 2 seconds.

3.2 Analysis of User-Generated Content from Twitter with spatio-temporal information

Tweet messages are timely spatio-temporal messages posted by a large variety of users actively involved in discussions addressing diverse topics. The potential business impact of mining social data is still largely unexplored. Service providers (e.g., TV channels, radio stations) as well as product placement agencies may explore and analyze Twitter posts along all three dimensions (spatial, temporal and textual content) to improve service/product provision according to the knowledge hidden in Twitter data. From a business point of view, it is worth profiling Twitter user trends and message topics along all three dimensions (i) to plan targeted promotions or identify exceptional events, (ii) to understand different user perceptions of an event in different geographical areas, (iii) to characterize the different facets of user involvement in different geographical areas and time frames. Innovative analytics solutions are needed to effectively and efficiently support the wide range of service profiling and interesting analysis to discover actionable knowledge from tweet data collections. The design of effective solutions for Twitter data analysis imposes new challenges as

(i) *Analysing large data collections with an inherent sparseness along all three tweet dimensions.* Tweet collections are characterized by a variable data distribution with a high degree of sparseness. The textual content is intrinsically sparse because posts range

over many different topics and use a wide vocabulary. The temporal aspect may include a wide range of time frames and the geographical locations may be distributed across a large number of countries. The variability in data distribution grows with data volume, thus increasing the complexity of mining such data. Different analytics strategies need to be combined to effectively extract actionable knowledge.

(ii) *Mining heterogeneous kind of data.* Since a single data mining technique may not fit the heterogeneous characteristics of rather different data types available in tweet messages, the integration of different types of algorithms is needed to reduce the complexity of the mining process.

(iii) *Scalable approaches to deal with large data volumes.* Since Twitter messages are being posted at an ever increasing rate, new and distributed solutions toward big data issues need to be designed.

Aimed at addressing the above issues, this section presents the application of MLDA framework (see Chapter 2), based on cluster analysis and pattern discovery, to gain interesting knowledge (e.g., topic and sentiment detection, people involvement) from large complex social data collections. The data analysis framework presented in this section enhances the methodology proposed in [30] by providing a more general approach which (i) integrates a novel combined distance measure to group twitter messages based on their spatio-temporal features and textual content, (ii) proposes a rule categorization into few reference classes according to their semantics, and (iii) exploits the computational advantages of distributed computing frameworks (i.e., Spark). The main components of the proposed approach are shown in Figure 3.4. Cluster analysis is driven by a novel combined distance measure to group twitter messages based on their spatio-temporal features and textual content, for discovering groups of tweets with similar textual content, but posted in nearby geographical areas and time. Each computed cluster is then locally characterized through a set of association rules to model correlations among words appearing in messages and the spatial and temporal information of tweets. To easy the analysis of the discovered patterns, we proposed a categorization of the rules into few groups according to their semantics, determined by the tweets characteristics appearing in the rule. The proposed methodology exploits the computational advantages of distributed computing frameworks, as the current implementation runs on Apache Spark.

As a case study, we focus on tweets posted on Twitter during the 2014 FIFA World Cup. Since each event takes place over a short defined period of time and a large number of persons are strongly involved during the football matches, a substantial number of tweets are collected about each event. The performance evaluation demonstrates the effectiveness of the proposed methodology in discovering interesting and actionable

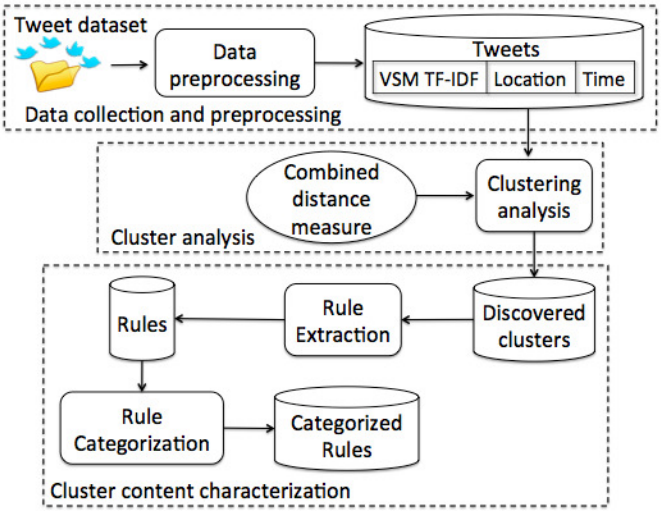


FIGURE 3.4: The proposed architecture

knowledge. Furthermore, the performance of the novel combined distance measure have been evaluated on large tweet datasets.

3.2.1 Related work

The application of data mining techniques to discover relevant social knowledge from tweet collections has become an appealing research topic. Many works proposed different strategies to address a joint analysis of tweet textual content and the corresponding spatial-temporal information. The targeted analyses are: (i) discovering regional social activities or nearby events using geo-tagged tweets [69, 70, 71], (ii) studyng social dynamics of Twitter activity [72], (iii) event detection based on cluster analysis [73] or on the SVM classifier [74], (iv) extracting insightful summaries of citizen perceptions from tweets [75], (v) discovering contrasting situations by means of generalized itemsets [76], (vi) detecting the period in which a burst of information diffusion took place from an observed information diffusion sequence of tweets [77], (vii) tackling the opinion retrieval [78].

To perform spatial analysis, works in [69, 70, 71] identify groups of tweets analysing GPS coordinates associated to tweets by driving the cluster analysis (based on K-means [69, 70]) through the Euclidean distance. A topic detection algorithm has been also used to extract from each cluster the main targeted topic. Works in [72, 74] focused on analyzing tweets related to a specific event, from a given geographical area. Conversely, the PhD study in the research work automatically detects different events occurred in any place, preferring flexibility for users to be informed of any event. Authors

in [73] proposed a framework to cluster tweets based on textual content and temporal information into thematic topics. A new distance measure based on temporal text similarity combines content and temporal dimensions in the IncrementalDBSCAN algorithm. Then, the spatial analysis block associates topic centric messages to appropriate locations in the map. Differently from the above works addressing a joint analysis of tweet textual content and the corresponding spatial-temporal information, we ehnanced previous approaches because we aimed at discovering similar events with similar time and location *in one step*. The proposed mixed distance measure combines the distance calculation of textual content with the spatial and temporal information. In this way, instead of estimating the temporal and spatial information after the event detection, the spatio-temporal information also participates in detecting the events. Moreover, we implemented the framework by exploiting the computational advantages of distributed computing frameworks.

3.2.2 Twitter data preparation

In this study, tweets have been characterized based on the following three main components: (i) *tweet textual content*, (ii) *tweet temporal feature*, and (iii) *tweet spatial feature*. The last two components are the contextual features describing the time and location when tweet has been posted. An example tweet for the reference case study addressed this work is reported in Table 3.4. Tweet components are described below.

Tweet textual content. Tweets are short, user-generated, textual messages of at most 140 characters long and publicly visible by default. Due to their limited size, these messages are inherently sparse. Moreover, they are usually extremely impure because they include a wide variety of Unicode data, symbols, numbers and links. Thus, tweets messages should be properly cleaned and prepared before applying the data analysis phase. In this study, the textual content has been represented using the *Bag-of-Word* (BOW) representation as described in Section 2.2.2.

Tweet spatial feature can be acquired as geo-coordinates, location specified in user profile, and location mentioned in the tweet textual content. Geo(graphical)-coordinates (i.e., latitude and longitude) are available when GPS enabled devices were used in accessing Twitter. They specify the spatial position of people when posting the tweet. Instead, the location specified in the user profile is free-text information provided by posters. It usually corresponds to the place (as city, state or country) where people come from. Since our aim is discovering tweets with similar textual content but posted in nearby geographical areas (and time periods), we focused on the spatial information provided by geo-coordinates.

Geo-coordinates	52.076171,-1.363145
Creation time	Fri Jun 20 09:26:53 +0000 2014
Textual content	England 2-0 I still believe

TABLE 3.4: Tweet example

Tweet temporal feature corresponds to the *timestamp* including date and hour when tweet was posted. In this study, we neglect the temporal information possibly appearing in the tweet message because less relevant for discovering tweets posted in nearby time.

3.2.2.1 Twitter Data Collection and Preprocessing

Tweet posts are retrieved from twitter.com via Twitter's Streaming Application Programming Interfaces (APIs). The Streaming APIs provide low latency access to Twitter's global stream of Tweet data, by establishing and maintaining a continuous connection with the stream endpoint. A java crawler has been used to collect and parse tweets in real time based on a predefined set of keywords (e.g., "worldcup2014", "fifaworldcup" in our case study), ignoring case considerations. Among the crawled tweets, we extracted English tweets only.

To suit the raw tweet textual content to the subsequent mining process, some preliminary data cleaning and processing steps have been applied. First, numbers, usernames and URLs mentioned in the content have been removed. Then after converting the letters into lowercase, tweet messages are purified by eliminating stop words (such as "is", "at" and "the"), and represented according to the Bag-of-Word representation. Collected tweets may be posted from different time zones. For computing the temporal distance between messages, the tweet time information has been preliminarily reported to a reference time zone. For example, in the use case considered in this study, the Brazil (America/Sao_Paulo) time zone, where the 2014 FIFA World Cup was held, has been selected as a reference.

Tweets may spread out over a vast geographical area and/or time frame. In this case, the tweet collection can be preliminarily partitioned based on same geographical areas and/or time periods. Then, the cluster analysis is locally performed in each segment.

3.2.2.2 Twitter Data Representation

This section formalizes the adopted data representation for the textual, temporal and spatial information of tweets.

Definition 3.1. Tweet data collection. Let $\mathcal{D}$ be a collection of tweets and $\Sigma = \{w_1, \ldots, w_k\}$ the set of words appearing in at least one tweet in $\mathcal{D}$. An arbitrary tweet $tw_i \in \mathcal{D}$ is represented as a triplet $tw_i = (t_{tw_i}, s_{tw_i}, W_{tw_i})$ where t_{tw_i} and s_{tw_i} are respectively the temporal and spatial features of tw_i, while W_{tw_i} is the tweet textual content.

According to the tweet characterization in Section 3.2.2, the temporal feature t_{tw_i} is the *timestamp* on *when* tweet tw_i was posted, while the spatial feature s_{tw_i} is the pair of *geo-coordinates* (latitude, longitude) reporting *where* tweet tw_i was posted. W_{tw_i} represents the *set of words* w_j, $w_j \in \Sigma$, appearing in tweet tw_i.

The Term Frequency (TF) - Inverse Document Frequency (IDF) scheme [2] usually used in text mining has been adopted to highlight the relevance of specific words for each tweet. This scheme reduces the importance of common terms in the collection. It allows focusing the tweet matching in the subsequent clustering phase on words specific for each set of tweets instead of words appeared in most tweets.

To weight word relevance based on the TF-IDF scheme, the tweet textual content is transformed using the Vector Space Model (VSM) representation [6]. Each tweet is a vector in the word space. Each vector element corresponds to a different word and is associated with the Term Frequency(TF)-Inverse Document Frequency(IDF) weight describing the word relevance for the tweet.

Definition 3.2. Tweet textual content representation. Let $tw_i = (t_{tw_i}, s_{tw_i}, W_{tw_i})$ be an arbitrary tweet in collection $\mathcal{D}$. The tweet textual content W_{tw_i} is a vector of $|\Sigma|$ cells in the word space Σ in $\mathcal{D}$. Each vector element $W_{tw_i}[w_j]$ contains the TF-IDF weight of word w_j for tweet tw_i. $W_{tw_i}[w_j]$ is computed as $W_{tw_i}[w_j] = TF_{tw_i,w_j} * IDF_{w_j}$, where terms TF_{tw_i,w_j} and IDF_{w_j} are defined as follows.

1. TF_{tw_i,w_j} is the relative frequency of word w_j for tw_i. $TF_{tw_i,w_j} = f_{tw_i,w_j} / \sum_{1 \leq k \leq |\Sigma|} f_{tw_i,w_k}$, where f_{tw_i,w_j} is the number of times word w_j appeared in tweet tw_i and $\sum_{1 \leq k \leq |\Sigma|} f_{tw_i,w_k}$ is the total number of words contained in tw_i.
2. IDF_{w_j} is the frequency of word w_j in Σ. $IDF_{w_j} = Log[|\mathcal{D}| tw_k \in \mathcal{D} : f_{tw_k,w_j} \neq 0|]$ where $|\mathcal{D}|$ is the number of tweets in $\mathcal{D}$ and $|tw_k \in \mathcal{D} : f_{tw_k,w_j} \neq 0|$ is the number of tweets in $\mathcal{D}$ which contain (at least once) word w_j.

Mathematically, the base of the log function for IDF computation in Definition 3.2 does not matter and constitutes a constant multiplicative factor towards the overall result. The TF-IDF weight $W_{tw_i}[w_j]$ for word w_j in tweet tw_i is high when w_j appears with high frequency in tweet tw_i but low frequency in tweets in the collection $\mathcal{D}$. When word

w_j appears in more tweets, the ratio inside the IDFs log function approaches 1, and the IDF(w_j) value and TF-IDF weight $W_{tw_i}[w_j]$ become close to 0. Hence, the approach tends to filter out common words. In short-messages as tweets, the TF-IDF weighting score could actually build down to a pure IDF due to the limited word frequency within each tweet. Nevertheless, we preserved the TF-IDF approach to consider also possible word repetitions.

3.2.3 Clustering analysis through a new distance measure

In this section, we describe the technical details of the proposed clustering method. Even if DBSCAN outperforms K-means in some example cases, we have adopted the K-means algorithm since it's currently available in Spark with the support for multiple parallel runs, and it has been also widely used in various applications domains, including tweets analysis, providing good quality solutions.

The K-means algorithm [2] discovers K clusters modeled by their representatives, named *centroids*, calculated as the mean value of the objects in the clusters. Initially, K tweets of the tweet collection $\mathcal{D}$ are randomly chosen as centroids. Then, each tweet tw_i in the collection $\mathcal{D}$ is assigned to the cluster with nearest centroid. Finally, centroids are relocated by computing the mean of the tweets within each cluster. The process iterates until the centroids do not change, or some objective functions are achieved. To discover clusters with "similar" content and posted in nearby area and time, we propose a novel distance measure evaluating tweet distance based on all the three aspects characterizing tweets.

3.2.3.1 The combined distance measure

The proposed distance measure evaluates tweet distance according to the three aspects characterizing tweets, i.e., timestamp, location and tweet textual content. Specifically, the distance between two arbitrary tweets tw_i and tw_j in $\mathcal{D}$ is computed as follows.

Definition 3.3. Tweet distance measure. Let $tw_i = (t_{tw_i}, s_{tw_i}, W_{tw_i})$ and $tw_j = (t_{tw_j}, s_{tw_j}, W_{tw_j})$ be two arbitrary tweets in collection $\mathcal{D}$. The distance $d(tw_i, tw_j)$ between tweets tw_i and tw_j is computed as

$$d(tw_i, tw_j) = \frac{d_W(W_{tw_i}, W_{tw_j}) \times e^{p_s d_s(s_{tw_i}, s_{tw_j})} + d_W(W_{tw_i}, W_{tw_j}) \times e^{p_t d_t(t_{tw_i}, t_{tw_j})}}{2} \quad (3.4)$$

where terms $d_t(t_{tw_i}, t_{tw_j})$, $d_s(s_{tw_i}, s_{tw_j})$, and $d_W(W_{tw_i}, W_{tw_j})$ measure the distance on temporal and location information, and tweet textual content, respectively. Parameters

p_t and p_s weight the relevance of the temporal ($d_t(t_{tw_i}, t_{tw_j})$) and spatial ($d_s(s_{tw_i}, s_{tw_j})$) distances in comparing tweet messages. Lower values of $d(tw_i, tw_j)$ denote a higher similarity between the two tweets, and higher values of $d(tw_i, tw_j)$ denote a lower similarity. All the distances in (3.4) have been normalized in the range $[0, 1]$ using the min-max normalization method [2]. Parameters p_t and p_s have been experimentally tuned in order to discover cohesive clusters.

In our proposed distance measure, the distance on the textual content between two tweets is weighted through their distance in time and space. It allows reducing the similarity between tweets with long time and/or large spatial distances. To significantly penalize distant tweets, the contribution of distances on time and space is given through an exponential function. In this study, we averaged the two contributions because we assumed that they have a similar relevance.

Different metrics have been used for computing the three distances in Equation 3.4 based on the type of attributes describing tweets.

Content distance evaluation ($d_W(W_{tw_i}, W_{tw_j})$). The distance between two weighted word frequency vectors is evaluated using the cosine distance measure [2], which has often been used to compare documents in text mining [4]:

$$d_W(W_{tw_i}, W_{tw_j}) = \arccos\left(cos(W_{tw_i}, W_{tw_j})\right) \tag{3.5}$$

where $cos(W_{tw_i}, W_{tw_j})$ represents the cosine similarity between W_{tw_i} and W_{tw_j}, and is computed as

$$cos(W_{tw_i}, W_{tw_j}) = \frac{\sum_{1 \le k \le |\Sigma|} W_{tw_i}[w_k] W_{tw_j}[w_k]}{\sqrt{\sum_{1 \le k \le |\Sigma|} W_{tw_i}[w_k]^2} \sqrt{\sum_{1 \le k \le |\Sigma|} W_{tw_j}[w_k]^2}} \tag{3.6}$$

The cosine distance in (3.5) verifies the triangle inequality [2]. The cosine similarity $cos(W_{tw_i}, W_{tw_j})$ is in the range [0,1]. $cos(W_{tw_i}, W_{tw_j})$ equal to 1 describes the exact similarity of textual contents for tweets tw_i and tw_j, while $cos(W_{tw_i}, W_{tw_j})$ equal to 0 points out that tweets have complementary texts.

Temporal distance evaluation ($d_t(t_{tw_i}, t_{tw_j})$). The tweet temporal feature (t_{tw_i}) is an integer number storing the timestamp on when tweet has been posted. The *Euclidean distance* [2] has been adopted for evaluating the temporal distance between tweets.

Spatial distance evaluation ($d_s(s_{tw_i}, s_{tw_j})$). Both Haversine and Euclidean distance measures have been used to calculate geo-coordinates distances [79, 80]. However, the

Haversine distance is usually considered as more appropriate and precise than the Euclidean distance especially when the distance between two points gets larger and it can not be approximated as a straight line. For this reason, in this study the *Haversine distance* has been adopted for computing the spatial distance bewteen tweets. The Haversine distance corresponds to the great-circle distance between two points, i.e., their shortest distance over the earth's surface. The *Haversine distance* $d_s(s_{tw_i}, s_{tw_j})$ is defined as follows

$$d_s(s_{tw_i}, s_{tw_j}) = 2 \cdot R \cdot \arcsin(\sqrt{h}) \tag{3.7}$$

$$h = \sin^2(\Delta\varphi/2) + \cos\varphi_1 \cdot \cos\varphi_2 \cdot \sin^2(\Delta\lambda/2) \tag{3.8}$$

where φ and λ are latitude and longitude values of the tweet and R is a constant value equal to the Earth's radius (mean radius = 6,371km).

Cluster evaluation. For the (internal) validation of clustering results, the framework adopts the *SSE* quality index (see Section 2.1.4), usually used for evaluating the performance of clusters.

3.2.4 Cluster content characterization

Association rules represent underlying correlations among the analyzed data items [81]. In the framework, each computed cluster is locally analysed using association rules to discover correlations in the textual content, and correlations between textual content and spatial-temporal features characterizing tweets.

3.2.4.1 Association rule extraction

To enable the mining process, tweet data contained in the cluster under analysis is tailored to a transactional data format. A transactional tweet dataset is a set of transactions in which each transaction corresponds to a tweet; it consists of a set of tweet features called items. Items are represented in the form *attribute* = *value*. Items can be related to words appearing in the tweet textual content, tweet spatial data or tweet temporal data. Items are represented in the form *attribute* = *value*. Specifically, let us consider a cluster C computed in the tweet collection $\mathcal{D}$. An item on tweet in cluster C is given by each different word appearing in the tweet textual content, value of the spatial feature s_{tw_i}, and value of the temporal feature t_{tw_i}.

The spatial and temporal features need to be discretized to discover a limited and interesting number of rules. To this aim, the time and location of tweets are described

with a coarser granularity. For example, the geographical location of the user when posting the tweet can be specified in terms of city, region and country. The information about when the tweet has been posted can be described with the hourly, daily time slot or daily granularity.

An association rule is an implication in the form $X \rightarrow Y$, where X and Y are disjoint itemsets (i.e., sets of data items). X and Y are also denoted as antecedent and consequent of the rule. Association rule extraction is commonly driven by rule support and confidence quality indexes. Whereas the support index represents the observed frequency of occurrence of the rule in the source dataset, the confidence index represents the rule strength. Rule support (*supp*) is the percentage of tweets containing both X and Y. Rule confidence ($conf$) is the percentage of tweets with X that also contain Y. To rank the most interesting rules, we also used the lift index [2], which measures the (symmetric) correlation between sets X and Y.

For example, the association rule *{start, love, world, cup → game}* (supp = 1.1%, conf = 75%) model correlations among representative words in tweets contained in a cluster. It talks about people's feeling on the World Cup game. 1.1% of tweets in the analysed cluster contain words *{start, love, world, cup, game}*, but the word *game* appears in all tweets including *{start, love, world, cup}*. In this work, to mine association rules representing strong correlations, rules with high confidence value and lift grater than 1 have been selected.

3.2.4.2 Association rule categorization

Nevertheless association rule technique is a powerfull method to discover data correlations, analyzing the (usually) large number of extracted rules is not a trivial task. To address this issue, we propose a categorization of the rules into few groups according to their semantics. The semantics of a rule is determined by its template including the attributes characterizing Twitter data. Specifically, classes are built upon the following attributes: tweet *spatial information* (abbreviated *location*), tweet *temporal information* (*time*), and *words* appearing in the tweet message (*word* w_i). Given this information, the following basic classes of rules can be defined:

1. *Correlations among words (WC)* in tweet messages included in the cluster. This class (denoted W(ord)C(orrelation)) mainly focuses on the tweet textual content, aimed at capturing the peculiar characteristics of messages posted in the cluster (i.e., which topic attract/involve users), neglecting both spatial and temporal details on *when* and *where* the tweet was posted. Instead, this information is

Topic ID	Description/Theme
T1	emotional states
T2	event location
T3	specific aspects of the event
T4	point of interest
T5	celebrities

TABLE 3.5: List of some topics

concisely represented by location and time values of the cluster centroid, selected as representative point in the cluster.

2. *Correlations among location and words (LWC).* This class (denoted L(ocation)W(ord)C(orrelation) analyses the correlations between words in tweet messages and the location where tweets have been posted. It allows identifying the topic attracting/involving users in a given location.
3. *Correlations among time and words (TWC).* This class (denoted T(ime)W(ord)C(orrelation)) analyses the correlation between words in tweet messages and the time when tweets have been posted, for discovering the topic attracting/involving users in a given time frame.
4. *Correlations among location, time, and words (LTWC).* This class (denoted LTWC) considers all properties characterizing tweets to analyse the correlation between words in tweet messages and both the time and the location tweets have been posted. It allows discovering the topic attracting/involving users in a given time frame and location.

The four classes progressively provide more detailed information. To ease the manual inspection of the discovered rules, we defined a short list of reference example topics reported in Table 3.5. For each topic a data dictionary of the characteristic words can be exploited to automatically identify the topic of each rule based on the word set appearing in the rule.

3.2.5 Experimental results

This section presents the results of the experiments with the proposed framework regarding (i) *quality evaluation* for the computed cluster sets, (ii) *cluster characterization* through association rules, and (iii) impact of the *parameters* (such as K) on the quality of the cluster set. The framework has been validated on a real collection of Twitter data related to 2014 FIFA worldcup held in Brazil used as a reference case study in this paper.

3.2.5.1 Evaluation setup and parameter configuration

The framework has been implemented as follows. Tweet textual messages have been cleaned in Python by removing stop words, links, and etc. The Machine Learning Library (spark.ml and spark.mllib packages) in Apache Spark has been used for all the other steps, i.e., the preprocessing phase containing VSM representation with TF-IDF weighting score calculation, the clustering analysis through K-means algorithm, and the cluster content characterization by using FP-growth to generate association rules.

The proposed combined distance measure has been implemented in Scala programming language and integrated in K-means in Spark. To evaluate the quality of the computed cluster set, the Sum of Squared Error (SSE) has been also implemented in K-means based on the new distance measure in Spark. To discover interesting correlations in the discovered cluster set through various indices, the calculation of lift value has been implemented in the association rule generation in Spark. The entire process has been implemented as an application in Scala language in the Apache Spark platform.

Based on the experimental evaluation discussed in Section 3.2.5.5, parameter setting (K=200, p_s=3 and p_t=6) has been used as reference default configuration for cluster analysis. The usual approach has been adopted for the K-means algorithm to address the problem of centroids initialization. Multiple runs, each with set of randomly chosen initial centroids have been performed, and then the cluster set with minimum SSE has been selected.

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

3.2.5.2 Datasets

The framework exploits a crawler to efficiently access the Twitter global stream (http://twitter.com). The public stream endpoint offered by the Twitter APIs was monitored over a time period of 27 days from June 18th to July 14th 2014 by tracking a selection of keywords related to the 2014 FIFA worldcup (as “worldcup2014”, “fifaworldcup”). Both textual content and the most relevant contextual features were collected. Tweets in English language were extracted and used in the subsequent analysis phase. Collected tweets have been then preprocessed and tailored to VSM representation using the TF-IDF weighting scheme (see Sections 3.2.2.1 and 3.2.2.2).

Dataset name	Time window	Partition	Number of transactions	Average transaction length
Dataset a_1	1	UK	29,864	8.10
Dataset b_1	1	USA	26,447	8.02
Dataset c_1	1	Central America	4,555	7.76
Dataset a_2	2	UK	15,175	8.43
Dataset b_2	2	USA	19,828	8.27
Dataset c_2	2	Central America	3,033	7.90
Dataset a_3	3	UK	34,392	8.46
Dataset b_3	3	USA	50,028	8.06
Dataset c_3	3	Central America	8,541	7.89

TABLE 3.6: Main characteristics of dataset partitions

In this study, a dataset containing 302,052 tweets with average tweet length 8.19 has been analysed. It includes the geo-coordinates on the position where tweets were posted, and timestamp to analyse the clustering performance in discovering groups of tweets not only with similar content but also posted in nearby area and limited time.

To analyze how tweet textual content unfolded over time, the tweet collection has been partitioned into three time segments based on the official time schedule of football matches on the 2014 FIFA WorldCup website. Specifically, *time-window #1* and *time-window #2* cover respectively the first and the second stage time period (i.e., June 18th ÷ June 27th and June 28th ÷ July 3rd), while *time-window #3* covers the remaining time period from quarter-finals to World Cup end (i.e., July 4th ÷ July 14th). The number of tweets is comparable in the three windows, but slightly higher in the last one including the most exciting football matches (i.e., semi-finals and finals).

The tweet spatial distribution has been then locally analysed within each time window based on tweet GPS geo-coordinates. In all the three time windows, tweets are widely dispersed and geographically separated in different areas. English speaking countries (as UK and USA) show higher tweet concentrations, even though tweets are also widely distributed in other areas (as Italy and Indian). Following this evaluation, three spatial segments with higher tweet concentrations have been selected for the subsequent data analysis, i.e., *UK*, *USA*, and *Central America*.

3.2.5.3 Analysis of the clustering results

Here we discuss the clustering results discovered with the framework. As a representative dataset for this analysis, we considered tweets posted in the first time window in the UK partition (i.e., Dataset a_i in Table 3.6). The first time window has been selected because a larger number of football events (as football matches) occurred in this time

frame. Consequently a more varied collection of tweets is possibly posted, in terms of textual content and wide geographical and temporal distribution.

Figure 3.5 provides an overview of the cluster cardinality in terms of number of tweets. Clusters are sorted on the x axis based on their number of tweets. Most clusters include a medium number of tweets (from 100 to 200 tweets). Figures 3.6 and 3.7 show the spatial and temporal distribution of the cluster set, respectively. In both figures, each cluster has been concisely described with the spatial and temporal features of its centroid.

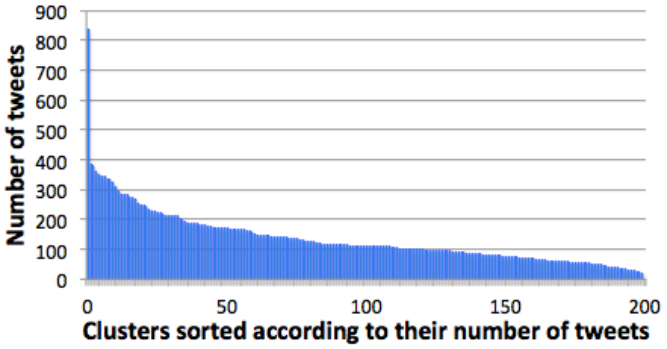


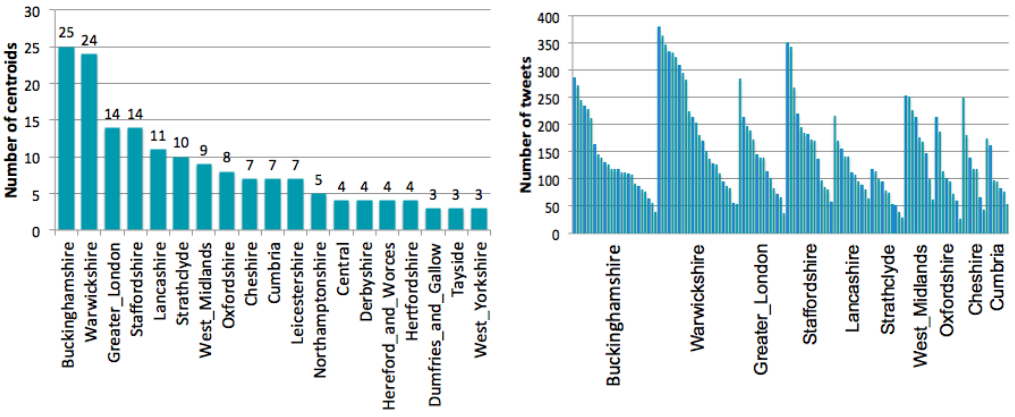
FIGURE 3.5: Distribution of number of tweets in the cluster set

To ease the comprehension of the results, the county in UK where the centroid is located is used to represent the geographical location of the centroid. For each tweet, the county information is retrieved by comparing the centroid geo-coordinates with the geo-coordinates of the boundaries in each county[1]. Figure 3.6(a) plots the number of centroids (i.e., clusters) for UK counties having at least three centroids (clusters). For each county, Figure 3.6(b) reports the number of tweets for each cluster in the county. Counties with a large number of medium-sized clusters (such as Buckinghamshire, Warwickshire, Greater London) also include a large number of tweets. They represent locations where people were strongly involved in the World Cup 2014.

Figure 3.7(a) plots the temporal distribution of centroids (i.e., clusters) considering a two-days time frame as a reference period (from June 19th to June 20th). Figure 3.7(b) shows the total number of tweets in the corresponding clusters. Similarly to the analysis on spatial distribution, the temporal information for centroids has been represented with a coarser granularity given by the hourly time slot.

The number of discovered centroids (clusters) increases in correspondence of two events occurred in the selected time period, i.e., the football matches Colombia−Cote D'Ivoire and Costa Rica−Italy (see Figure 3.7). For both matches, the starting hour is highlighted in the figure. The number of centroids (clusters) increases at the time frame preceding the match starting hour, while it reaches a peak during the match. The number of

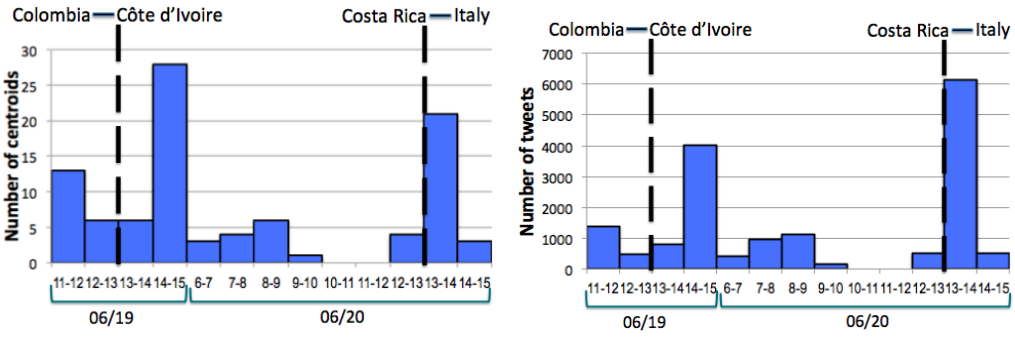
[1]http://www.nearby.org.uk/downloads.html



(a) Spatial distribution of the number of centroids

(b) Clusters grouped according to their centroids defined as county

FIGURE 3.6: Spatial characterization of the cluster set



(a) Temporal distribution of the number of centroids (clusters)

(b) Clusters grouped according to their centroids defined as hourly time

FIGURE 3.7: Temporal characterization of the cluster set

tweets in clusters shows an analogous trend. These results point out that the evaluation of temporal distribution of the discovered clusters (and their size) can support in the detection of the occurred events.

The characteristics of five example clusters in Figure 3.5 are detailed in Table 3.7 in terms of textual content (number of tweets), geo-coordinates (city of the cluster centroid, maximum and average GPS distances among tweets in the cluster), temporal information (date of the cluster centroid, maximum time period in the cluster). Results show a good balance among the three facets characterizing tweets. The cluster with larger number of tweets represents common topics discussed by users located in a wide geographical area and within a relatively long time period. Messages in smaller clusters may spread out in a smaller geographical area and in a shorter time period.

Cluster ID	# of tweets	City of centroid	MAX GPS distance (KM)	AVG GPS distance (KM)	Date of centroid	Max Time distance (Hour)
A	332	Brandon and Bretford	684.70	173.60	2014/06/20	2.96
B	285	Chartridge	504.75	119.45	2014/06/20	2.87
C	283	Uxbridge	474.01	99.69	2014/06/20	2.96
D	212	Uxbridge	472.99	80.52	2014/06/25	1.3
E	197	London	333.62	62.69	2014/06/25	1.55

TABLE 3.7: Characterization of five example clusters in Figure 3.5

We also detailed cluster C with centroid in the Greater London county. For this cluster, Figure 3.8 reports the distribution of the number of tweets in the top ten counties. Figure 3.9 shows the number of tweets per hour. The majority of tweets are located in Greater London county, while the others are mainly spread out in 4 counties. This result shows that the clustering algorithm has been able to group in the same cluster tweets on a similar topic written by users spread out in the wide area, including different counties. The distribution in Figure 3.9 highlights an increasing number of tweets, which grows significantly when the time becomes close to the football match starting time (Italy-Costa Rica in this case).

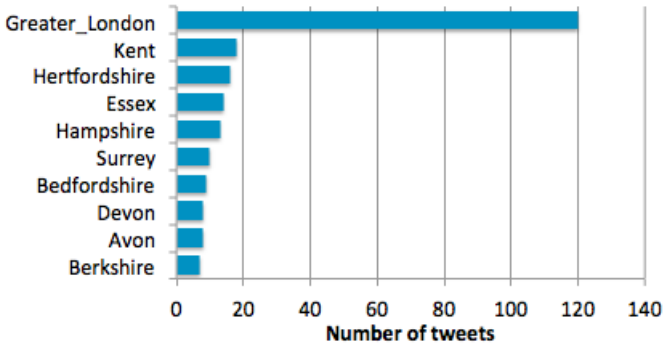


FIGURE 3.8: Cluster located in the Greater London county: distribution of the number of tweets w.r.t. the top ten counties

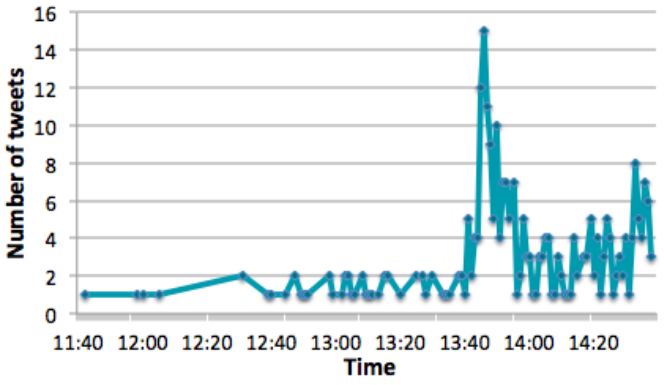


FIGURE 3.9: Cluster located in the Greater London county: distribution of the number of tweets w.r.t. hourly time frame

Despite Apache Spark, the experiments have been performed also in Apache Mahout MapReduce platform by implementing the new distance measure in java programming language and integrating it in K-means algorithm. Results show that K-means in Apache Mahout is able to generate more cohesive clusters, with much more performance time instead. In addition, the FP-Growth algorithm has been deprecated in the newest version of Mahout. For these reasons, Apache Spark has been used for the entire data analysis process in this study.

3.2.5.4 Cluster characterization using association rules

The cluster content has been concisely described using association rules. Rules have been extracted according to the class template defined in Section 3.2.4.2. However, since currently in Apache Spark the parallel rule generation algorithm is able to construct rules with a single item as the consequent, in this reference case study we mainly focus on rules that have a single item as the consequent.

Rules analysis on an example cluster. Rules have been extracted to model correlations among the analyzed data items (words, location and time) within each cluster. As an example of the type of information mined using these patterns, Table 3.8 reports some association rules extracted from cluster C on Dataset a_1 (see Table 3.6). For each rule class template defined in Section 3.2.4.2 some rules are presented in Table 3.8. In cluster C, rules are related to a variety of topics as event, emotional state and celebrity mainly on the football match Italy-Costa Rica on June 20 2014.

Tweet textual contents are usually quite sparse due to the short length of messages but the large variety of possible words. Thus, the selection of representative rules has been mainly driven by the relevance of the lift value instead of other quality indices as support and confidence. Rules with lift values greater than 1 represent relevant correlations among the analyzed data items. For association rule extraction, the tweet geo-coordinates have been discretized into the corresponding counties, while the tweet posting timestamp into a 2-hours time frame. Rules have been extracted by setting $supp = 1\%$ but without enforcing lift and confidence thresholds.

Correlations among words in tweet messages (class WC) are modeled in rules R_1 and R_2. These rules are enriched with spatial and temporal detail on the cluster centroid. Rule R_1 captures people's emotional state on the football match involving Costa Rica. Instead, in rule R_2 people talked about a celebrity Gary Lineker, a retired English footballer and current sports broadcaster, who wore an Italy shirt, since England needed Italy to beat Costa Rica to keep their World Cup hopes alive.

Rule id	Class	Topic description	Rule	supp	conf	lift
R_1	WC	Emotional state	centroid(time=2014/06/20 [12:00-2:00p.m.], location=Greater London): {fancy, costa, rica} ⇒ {chances}	1.1%	75%	53.25
R_2	WC	Event	centroid(time=2014/06/20 [12:00-2:00p.m.], location=Greater London): {shirt, italy} ⇒ {lineker}	1.1%	100%	56.8
R_3	LWC	Event	{watching, costa, rica} ⇒ {location=Greater_London}	1.4%	66.7%	1.58
R_4	LWC	Emotional state	{bad, england} ⇒ {location=Greater_London}	1.1%	100%	2.37
R_5	TWC	Emotional state	{bad, italy} ⇒ {time_frame=[2:00-4:00p.m.]}	1.1%	50%	1.23
R_6	TWC	Emotional state	{need, win, england} ⇒ {time_frame=[2:00-4:00p.m.]}	1.1%	100%	2.47
R_7	TWC	Celebrity	{robbiesavage, playing, italy, costa} ⇒ {time_frame=[12:00-2:00p.m.]}	1.1%	100%	1.71
R_8	LTWC	Event	{lose, italy, time_frame=[12:00-2:00p.m.]} ⇒ {location=Greater_London}	1.1%	60%	1.42
R_9	LTWC	Event	{location=Greater_London, england, time_frame=[12:00-2:00p.m.], costa, rica} ⇒ {goal}	1.1%	15.0%	10.65

TABLE 3.8: Rules characterizing cluster C extracted from Dataset a_1

Correlations between location where tweets have been posted and words in the messages (class LWC) are reported in rules R_3-R_4. While correlations between time when tweets have been posted and words in the messages (class TWC) are shown in rule R_5-R_7. These rules refer to the event, people's emotional states and specific celebrity related to the football match Italy-Costa Rica. Correlations between time, location and words (class LTWC) are in rule R_8-R_9. For example, rule R_9 shows that the event on Costa Rica was discussed in the Greater London county in a given time frame.

Rules analysis across the three partitions Considering as time window #1 as a reference example, we compared rules extracted from UK, USA and Central America partitions (Datasets a_1, b_1, and c_1). Rules representing correlations among words are discussed (i.e., class WC). For association rule extraction, the tweet geo-coordinates have been discretized into the corresponding city, while the tweet posting timestamp into a daily time frame. Rules have been extracted by setting $supp = 1\%$ but without enforcing lift and confidence thresholds. Some example rules related to a variety of topics as event, emotional state and celebrity are reported in Table 3.9.

Some events have been widely discussed in all three partitions. For example, the match between Costa Rica and Italy (on June 20th) was discussed in both in UK and USA (rules R_1 in Table 3.8 and R_3 in Table 3.9). Moreover, people may show a higher interest for events related to their geographical area. For example, on June 18, in Central America people still focused on the Brazil-Mexico match (rule R_6) held on June 17.

Rule id	partition	Topic description	Rule	supp	conf	lift
R_1	UK	Celebrity	centroid(time=2014/06/25, location=Rugeley): {suarez, someone} ⇒{bite}	3.0%	80%	26.9
R_2	UK	Celebrity	centroid(time=2014/06/25, location=Rugeley): {free, kick} ⇒{messi}	1.9%	71.4%	3.5
R_3	USA	Event	centroid(time=2014/06/20, location=Whittier): {costa, rica} ⇒{italy}	8.3%	64.3%	1.67
R_4	USA	Emotional state	centroid(time=2014/06/20, location=Whittier): {better} ⇒{italy}	1.6%	64.3%	1.67
R_5	Central America	Event	centroid(time=2014/06/18, location=Plantation): {england, stop, tomorrow, watch} ⇒{uruguay}	1.5%	100%	34
R_6	Central America	Event	centroid(time=2014/06/18, location=Houston): {mexico, last, night, outside, europe, cup} ⇒{brazil}	1.5%	100%	68

TABLE 3.9: Rules characterizing clusters in Datasets a_1, b_1, and c_1 (rule class WC)

3.2.5.5 Impact of parameters K, p_s and p_t

This section analyses the impact of the parameters K, p_s and p_t on the quality of the cluster set. Cluster quality is evaluated as the total SSE in the cluster set, average, minimum, and maximum SSE among the clusters, for a range of K. This evaluation is also known as *elbow method*. As a reference example, the tweet collection for the UK partition within the first time window (TW1_UK) has been considered. This data partition includes a large number of tweets covering a vast geographical area.

Figure 3.10 plots how the cluster quality varies when increasing the value of K, corresponding to the number of clusters in the final set. Parameters p_s and p_t were set to standard values p_s=3 and p_t=6. When growing K, and thus the number of clusters, the SSE value progressively decreases. K can be selected by trading-off the desired number of clusters and the expected cluster quality. Specifically, when K=200 the SSE value decreases abruptly, the average SSE of the cluster set become close to the minimum and maximum SSE values. The number of clusters in the set is also acceptable since a vast geographical area is covered. Thus, K=200 has been selected as default value.

Similar to the selection of parameter K, the performance of different pairs of p_s and p_t has been compared based on the SSE values when K is fixed as 200. Results show that configuration ($p_s = 3$ and $p_t = 6$) provides optimal results for the experiments on tweets in this study.

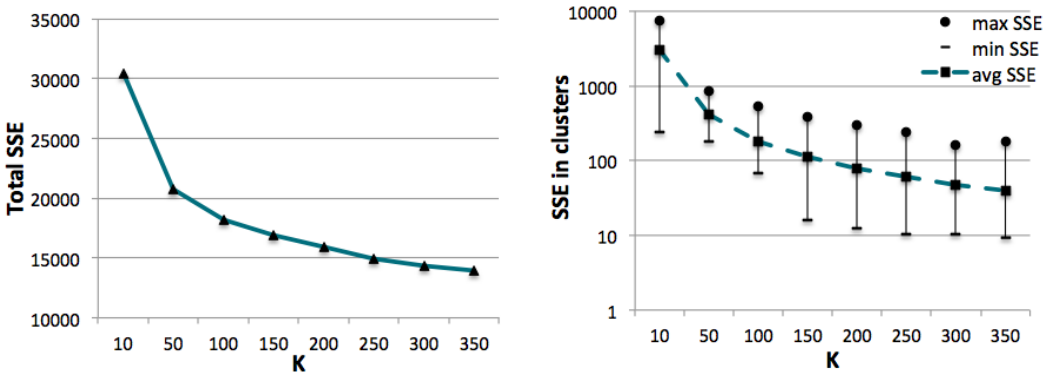


FIGURE 3.10: Cluster quality by varying K for TW1_UK ($p_s = 3$, $p_t = 6$)

3.3 Analysis of air pollution data

This section presents a new data mining system, named GEneralized Correlation analyzer of pOllution data (GECKO), to discover interesting and multiple-level correlations among a large variety of open air pollution-related data. Specifically, correlations among pollutant levels and traffic conditions representing User-Generated Data on people mobility as well as climate conditions are analyzed at different abstraction levels. The knowledge extraction process is driven by a taxonomy to generalize low-level measurement values as the corresponding categories. To ease the expert-driven rule inspection process, the extracted correlations are classified into few classes based on the semantics of underlying data. The GECKO system was validated on real data collected in a major Italian city.[2] This work has been carried out in collaboration with another PhD student Giuseppe Ricupero. The results have been summarized in paper titled "Modeling correlations among air pollution-related data through generalized association rules", which has been accepted for publication in the Second International Workshop on Sensors and Smart Cities (SSC) co-located with the 2nd IEEE International Conference on Smart Computing, St. Louis (Missouri), 18-20 May 2016.

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 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 and enriched with a taxonomy aggregating concepts into higher-level ones; (iii) *Data analyses*, in which generalized association rules are extracted from the prepared data to support domain experts in performing advanced analyses.

[2]The research has received funding from the Italian Ministry of Research (MIUR) under the "Cluster Tecnologie Smart Communities Progetto MIE - Mobilità Intelligente Ecosostenibile".

A more detailed description of each block is given later.

3.3.1 Related work

The evaluation of how the different factors, e.g., weather conditions and human activities, impact on the air quality is currently a relevant research issue. Previous works have already studied the correlation between different pollutants through statistics-based methods such as one-way ANOVA analysis [82]. Furthermore, Principal Component and Canonical Correlation analyses [83] have been exploited to analyze the correlation between pollutants and meteorological data [84]. A parallel effort has been devoted to exploiting data mining techniques to analyze the air quality levels in urban environments [85, 86]. Classification algorithms have been exploited to predict the air quality level in areas not equipped with monitoring stations [85]. 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 [86] 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 [87], social network data analysis [76]) 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. In the research work, to deal with high dimensionality due to the various factors, taxonomy has been integrated with generalised association rules to discover correlations between items from different abstraction levels.

3.3.2 Data collection and representation

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 the following we first describe

the considered data types, and then the data integration strategy currently adopted in GECkO.

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 $PM_{2.5}$, 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.

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, we considered traffic data separately for each category of vehicles. Specifically, vehicles are categorized based on their fuel type (e.g., gasoline, diesel, electric).

Meteorological data. To analyze the climate conditions of the urban area, the GECkO 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.

To allow the analysis of the correlations between pollutant levels and traffic conditions, taking into account also the impact of weather conditions, the three different types of data described above are integrated into a unique repository. Traffic and meteorological 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) traffic and meteorological data are aligned to the closest timestamp available in pollutant data through an approximate join. 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.

3.3.2.1 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, and enriched with a taxonomy, which generalized the relational model to a multiple-level model.

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). In our context of analysis, the considered attributes are shown in Table 3.10.

TABLE 3.10: Dataset attributes

Pollutant levels	*Traffic indicators*	*Meteorological levels*	*Time*
Particulate matter PM_{10}	# gasoline engine vehicles	Wind direction (degrees)	Hourly timeslot
Particulate matter $PM_{2.5}$	# diesel engine vehicles	Wind speed ($\frac{km}{h}$)	Date
Ozone O_3	# electric engine vehicles	External temperature (Censius degrees)	
Nitrogen dioxide NO_2	# natural gas engine vehicles	External humidity ($\frac{kg}{m^3}$)	
Carbon Monoxide CO	# hybrid vehicles	UV radiations ($\frac{W}{m^2}$)	
Benzene C_6H_6		Pressure (Pa)	
		Precipitations ($\frac{l}{m^2}$)	

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 [88] (e.g. *blue* and *green* imply non-critical levels, while *orange* and *red* indicate highly critical levels).

TABLE 3.11: Discretized humidity values and UV radiations

Attributes	Categories	Ranges
Humidity ($\frac{kg}{m^3}$)	very low	[0, 20]
	low	(20, 40]
	medium	(40, 60]
	high	(60, 80]
	very high	(80, 100]
UV ($\frac{W}{m^2}$)	very low	[0, 0.9]
	low	(0.9, 2.9]
	medium	(2.9, 5.9]
	high	(5.9, 7.9]
	very high	(7.9, 10.9]
	extremely high	(10.9, infinity)

The *traffic indicator* values are uniformly discretized by using the equal-width discretization algorithm available in the RapidMiner suite [26].

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, Table 3.11 shows the discretization levels of humidity values and UV radiations. 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*).

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. In the experiments, the taxonomy reported in Table 3.12 is considered. 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).

TABLE 3.12: Example taxonomy

Attrib.	*PM_{10}, $PM_{2.5}$*	*Traffic*	*Wind direction*	*Hourly timeslot*	*Date*
Level-4					season
Level-3				8-hour timeslot	month of year
Level-2	non-critical | critical | highly-critical	total range	east-side | west-side	4-hour timeslot	week of month
Level-1	blue, green | yellow, orange | red	equal-width sub-ranges	north-east, east, south-east | south-west, west, north-west | south | north	1-hour timeslot	Date

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 [76], 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.

3.3.3 Data analysis through generalised association rules

This block aims at discovering interesting associations between pollutant levels and traffic conditions describing people's mobilities, taking into account also the impact of meteorological conditions, in the form of generalized association rules. Association rule mining [89] is an exploratory data mining technique that has largely been used to extract hidden correlations among data items from large datasets.

To introduce the concept of association rule, we first recall the notion of itemset. 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.

To analyze pollutant data at different granularity levels, the itemset definition can be straightforwardly extended to the case in which data are enriched with a taxonomy. A *generalized itemset* [90] 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 3.3.2.1).

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* [90] is an implication $A \rightarrow B$, where A and B are disjoint generalized itemsets, i.e., generalized itemsets having no attributes in common. A and B are denoted as the antecedent and consequent of rule $A \rightarrow B$, respectively. Generalized rules are characterized by three main quality index, i.e., support, confidence, and lift.

The *support* of a rule $A \rightarrow B$ ($s(A \rightarrow B)$) corresponds to the support of itemset $A \cup B$ in the analyzed dataset. It indicates the observed frequency of occurrence of the rule. High-support rules represent recurrent patterns that are likely to occur in the analyzed data not by chance.

The *confidence* of a rule $A \rightarrow B$ ($c(A \rightarrow B)$) is the conditional probability of occurrence of generalized itemset B given generalized itemset A. It indicates the strength of the implication ($\rightarrow$) and it is computed as the percentage of records matched by all the items in A and B (i.e., the support of the rule) over the number of records matched by items of the rule antecedent (i.e., the support of A). High-confidence rules are more likely to be reliable than low-confidence ones, because the implication is true in the majority of the cases.

The *lift* of a generalized association rule $A \rightarrow B$ is defined as $lift(A,B) = \frac{c(A \rightarrow B)}{s(B)} = \frac{c(A \rightarrow B)}{s(A)s(B)}$ [2], where $s(A \rightarrow B)$ and $c(A \rightarrow B)$ are respectively the rule support and confidence, and $s(A)$ and $s(B)$ are the supports of the rule antecedent and consequent. If lift(A,B)=1, the itemsets A and B are not correlated, i.e., they are statistically independent. Lift values below 1 show negative correlation, while values above 1 indicate a positive correlation between itemsets A and B.

The mining problem 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 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), the GenIO algorithm 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 [87] and [89], respectively.

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-Traffic (PT) 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*) $\rightarrow$ (*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.

Rules representing correlations between pollutants and other factors such as meteorological conditions and temporal attributes have been also extracted. These rules include (i) *Pollutant-Pollutant (PP)*, comprising all the rules that contain only items belonging to attributes related to pollutant concentration levels. (ii) *Pollutant-Meteo (PM)*, comprising all the rules that contain items related to pollutant concentration levels and meteorological conditions (e.g., temperature, humidity). (iii) *Pollutant-Date (PTE)*, comprising all the rules that contain items related to pollutant concentration levels and temporal attributes (e.g., date, time).

More complex rules, e.g., class *Pollutant-Meteo-Traffic (PMT)*, can be extracted as well. They represent implications between pollutant levels and traffic conditions, infected also by other factors such as meteorological conditions (e.g., rule (PM_{10}, *red*) $\rightarrow$ {(*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.

3.3.4 Experimental results

The proposed approach was validated on real data acquired in Milan, which is one of the largest and most important Italian Smart Cities. To perform our analyses, we considered two open datasets collecting the sensor measurements 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 [87] provided by the respective authors. We extracted frequent and high-confidence rules, 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.

3.3.4.1 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 first dataset, hereafter denoted as *DailyMeasures*, collects the daily pollutant levels measured on a daily basis and traffic conditions describing people's mobilities, as well as the environmental information about meteorological conditions. The second dataset (*HourlyMeasures*) collects the hourly pollutants levels and traffic conditions, together with the corresponding meteorological conditions.

Pollutant data were gathered by the ARPA Lombardia [88]. 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 [91], which gathers data from a geo-referenced network of Personal Weather Stations (PWSs) registered by users. We considered three PWSs located in the city center. Traffic data were provided by the Municipality of Milan[3]. They consist of the counts of the number of vehicles entering in the central area of Milan, separately for each category of vehicles.

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

3.3.4.2 Knowledge discovery

The extracted rules were categorized, according to the type of item correlations they represent, into the classes described in Section 3.3.3. Classes PT and PMT represent rules describing correlations between pollutants and traffic conditions, without or with the impact of meteorological conditions. For each class, a subset of the most interesting rules extracted from both datasets is reported in Table 3.13. The mined generalized association rules provide insightful information, indicating the levels at which pollutants and traffic conditions are actually influenced with each other, affected or not by the climate factors.

TABLE 3.13: Rule examples.

ID	Dataset	Rules	Sup (%)	Conf (%)	Lift
		Class PT			
R_1	DailyMeasures	(*num.diesel engine vehicles,medium*) → (PM_{10},*fairly high*)	9.7	70.0	1.8
R_2	DailyMeasures	(*num.diesel engine vehicles,high*) → (PM_{10},*green*)	14.7	34.4	0.9
R_3	DailyMeasures	(*num.gasoline engine vehicles,high*) → (CO,*low*)	9.2	73.3	1.4
		Class PMT			
R_4	DailyMeasures	{(*pressure,low*), (*num.petrol engine vehicles,medium*), (PM_{10},*blue*)} → ($PM_{2.5}$,*blue*)	20	95	1.96
R_5	DailyMeasures	{(*uv,very low*), (*num.gasoline engine vehicles,high*)} → {(*windspeed,light air*), (O_3,*non-critical*)}	21	72	1.94
		Other rules			
R_6	DailyMeasures (**PP**)	(PM_{10},*yellow*) → ($PM_{2.5}$,*yellow*)	9.7	72.9	5.4
R_7	HourlyMeasures (**PP**)	(O_3,*highly critical*) → (NO_2,*non-critical*)	5.9	51.6	1.5
R_8	DailyMeasures (**PM**)	*drizzling*), (PM_{10},*orange*), ($PM_{2.5}$,*red*)} → {(*precipitations*, {(*temperature,very cold*), (*CO,fairly high*)}	1.1	40.0	20.1
R_9	DailyMeasures (**PTE**)	(*date,spring*) → (PM_{10},*green*)	5.6	55.6	1.4

Correlations between pollutant levels and traffic conditions, without or with the impact of meteorological conditions (Class PT and PMT). 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_1-R_3 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. When considering also the impact of meteorological conditions, more complex rules can be extracted. For example, R_4 shows a strong correlation among pressure, number of petrol engine vehicles and pollutants as PM_{10} and $PM_{2.5}$, with a very high confidence. R_5 shows when uv radiation is very low, and the number of gasoline engine vehicles is high, the pollutant O_3 become non-critical if the wind speed is light air. The high confidence values indicate

that there are strong correlations between pollutants and traffic conditions describing people's mobilities.

Correlations between other factors. As shown in Table 3.13, rules representing correlations between other factors can be also extracted out. According to R_6, there's a positive correlation between two pollutants PM_{10} and $PM_{2.5}$, with approximately 73% probability. R_7 shows the inverse relationship between the levels of Nitrogen dioxide (NO_2) and Ozone (O_3). Hence, a high concentration of Ozone is often associated with a low concentration of Nitrogen dioxide. According to Rule R_8, 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). Rule R_9 shows that there's also correlation between pollutants and season.

Chapter 4

Analysis of historical data

This chapter describes data mining techniques developed in the PhD activity for the analysis of historical User-Generated Data (UGD). Historical data is data collected in past-periods, used usually as a basis for forecasting the future data values or trends. Historical data is often represented as time series records, which has been useful in helping predict the future of a company and a market through predictive analyses. In the research work presented in this thesis, historical User-Generated Data, such as patient physiological data and user exploitation of service in bike sharing system, has been considered for prediction analysis (see Figure 4.1).

In this thesis we considered some reference example of historical UGD coming from the health care domain and the urban scenario. In health care domain, an example of historical UGD is the collection of physiological signal values describing the cardiac and respiratory response of patients during a cardiopulmonary exercise test [92]. The physiological signal values are multivariate time series where all signals are continuously monitored during the test to analyze the body response to increasing workload. These signals include for example heart rate, oxygen consumption, and inspired/expired carbon dioxide. Since the test is physically very demanding, innovative data analysis techniques are needed to predict patient response thus lowering body stress and avoiding cardiopulmonary overload.

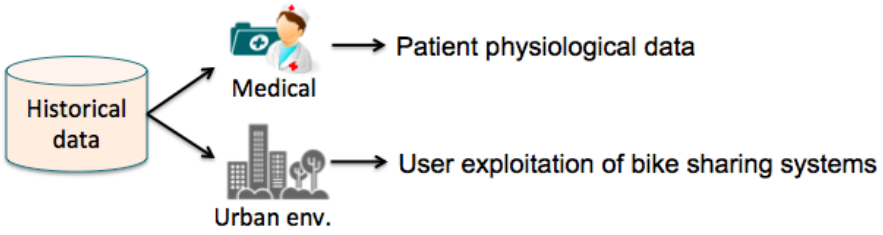


FIGURE 4.1: Historical data analysis

In Smart urban environments, different services are currently provided to citizens. Monitoring and analyzing historical User-Generated Data reporting how these systems have been used by citizens can provide useful feedbacks for improving the quality of the offered services. Among these services, the analysis of data acquired from bicycle sharing systems represents a key challenge in the human mobility scenario [93]. Bicycle sharing systems are eco-friendly transportation systems that have found wide application in Smart urban environments. They allow citizens to rent bicycles from a station equipped with a fixed number of slots, and return them in any other station with any free slot. The benefits of using these systems comprise: (i) environment safeguard, because bicycles are eco-friendly ways of transport, which preserve the air quality of the city, (ii) prevention of traffic congestion, because people who need to move from short trips may avoid using cars or other vehicles, and (iii) people wellness, because riding a bicycle is a good way to maintain or improve their fitness. In bicycle sharing systems, monitoring and analyzing when users rent and return bikes from/to stations allows assessing the occupancy levels of the system's stations. This aspect is crucial for guaranteeing the quality of the offered service. In this example scenario, the stations' occupancy value is continuously monitored over time. The occupancy value in both current and past instants should be analyzed for the prediction.

To deal with the above issues, in the research activity, we proposed a framework to predict future values based on historical User-Generated Data collected in a time window. The main architecture blocks, depicted in Figure 4.2, are (i) *Data collection and preparation*, where the historical data is acquired at different sampling time instants, collected and prepared to the next data mining step. (ii) *Data modelling*, where a windowing approach has been exploited to consider not only the current value but also a snapshot of values acquired at the previous instants. Moreover, the original dataset is enriched with some additional features to better characterize the considered context and thus support a more accurate prediction result. (iii) *Prediction analysis*, where a prediction model based on regression or classification techniques has been applied to predict the future data values. The extracted knowledge can help domain expert to make decisions in advance and improve the quality of service.

In this thesis, the framework has been exploited to predict patient physiological data in health care and to predict occupancy levels of bike sharing service in urban environment. More specifically, in the health care domain, regression techniques have been exploited in the framework to early predict the physiological signal values that can be reached during an incremental cardiopulmonary exercise test. The prediction allows physicians to decide when to prematurely interrupt test execution and avoid cardiopulmonary overload of patients. In the urban environment, Bayesian and associative classifiers have been applied to predict the occupancy levels (i.e., critical or non-critical) of the stations in

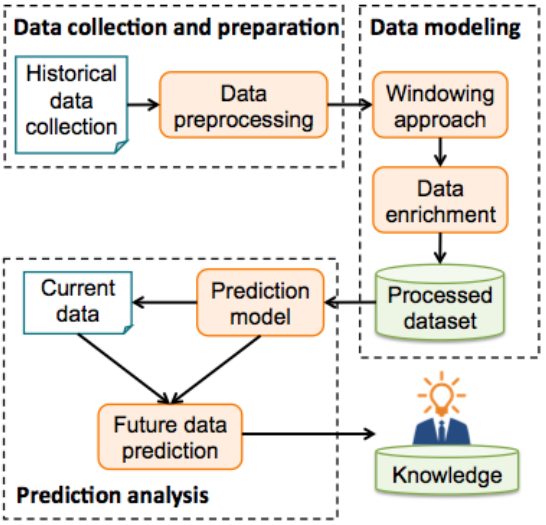


FIGURE 4.2: Historical data analysis framework

the near future, based on the analysis of (time series) historical data on the station occupancy values enriched with additional temporal information to fully characterize the predicted time instant. The model allows predicting short-term station occupancy levels and characterizing system usage to support planning maintenance activities in the medium term as well. More detailed description of each activity is given later.

This chapter is organized as follows. Section 4.1 presents the prediction analysis of patient physiological data. Section 4.2 describes the prediction of stations' occupancy levels in bike-sharing systems.

4.1 Analysis of patient physiological data

Cardiopulmonary exercise testing is a non-invasive method widely used to monitor various physiological signals, describing the cardiac and respiratory response of the patient to increasing workload. Incremental tests are commonly used to progressively increase the mechanical demand that the individual cardiopulmonary system has to match until she/he can no longer maintain the applied workload. Various physiological signals, mainly describing the patient cardiac and respiratory functions, are monitored during the test to analyze the body response to increasing strain. This cardiopulmonary response, when skeletal muscles transform chemical energy into mechanical output, has been shown to be best described by patient's peak aerobic power (VO_{2peak}), i.e., the oxygen consumed by exercising muscles per unit of time at peak incremental effort.

Since cardiopulmonary tests are physically very demanding, long test durations can significantly increase the body stress on the monitored individual and may cause cardiopulmonary overload. It follows that the capability to early predict the patient body response to the exercise during the test execution is a challenging issue. The aim is lowering the body stress, by prematurely interrupting the test and by avoiding its entire execution, without missing the information on the cardiopulmonary adaptation for the monitored individual.

This section proposes the Cardiopulmonary Response Prediction (CRP) framework for early predicting the physiological signal values that can be reached during an incremental test. During the test execution, CRP analyses various vital signals for the patient executing the test, and automatically predicts the signal values achievable at different subsequent steps of the test (i.e., at the next step, when the test ends or at an intermediate step of the test). Through the periodical prediction of the individual cardiopulmonary response to the test, physicians can decide when to prematurely stop the test execution, thus lowering the body stress. To obtain an accurate prediction, the learning phase creates different models tailored to specific conditions (i.e., *single-test* and *multiple-test* models). Each model can be exploited in the real-time stream prediction phase to periodically predict, during the test execution, signal values achievable by the patient. The research activity presented in this section has been published in [92].

In the field of cardiopulmonary signal analysis, many research efforts have been devoted to analysing the patient cardiopulmonary response through: (i) the analysis of signal patterns collected during exercise tests [94, 95, 96, 97], such as CRP or the (ii) electrical simulation models [98, 99, 100]. Cardiopulmonary exercise testing can be used by analyzing accessible physiological signal patterns collected during exercise, such as ventilation, VO_2, HR, blood pressure and body temperature [101]. VO_{2peak} has been estimated in both normal subjects and several patient populations from submaximal signals, such as rating of perceived exertion, workload, and heart rate [94, 95, 96, 97]. Most of the above work is based on statistical analysis of periodic snapshots of physiological parameters on a weekly or monthly basis. Instead, the CRP framework using data mining techniques allows to early predict the signal values achievable at different subsequent steps of the test *during the test execution*. Signals prediction during the test execution allows to indirectly assess the key physiological information on the patient's response to the test (e.g., VO_{2peak}), thus reducing the test duration and lowering the body stress of patients.

Data mining techniques have been widely used both in the healthcare and sports domain

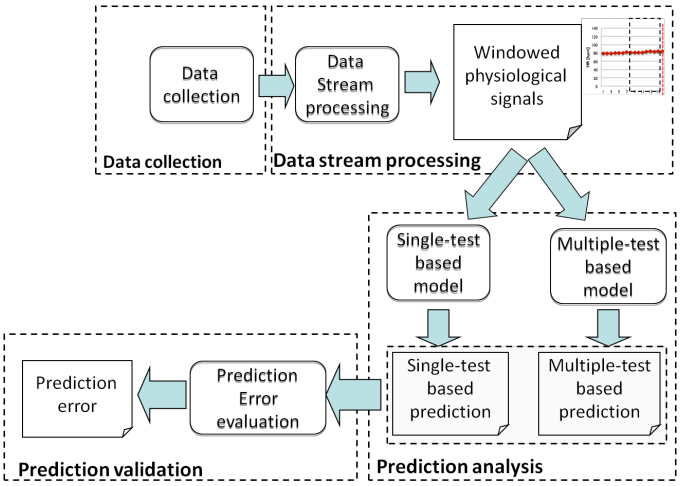


FIGURE 4.3: The CRP framework

to analyze physiological signals and support clinicians and exercise physiologists. Common techniques as support vector regression, artificial neural network and other classifiers have been used for blood glucose level prediction [102], emotion recognition [103], and other kinds of disease-related predictions through physiological signal analysis [104]. The early prediction of maximum workload value reached by the individual in the incremental test was first studied in [105, 106] for endurance sports testing. Different from these works, in the research activity, the proposed CRP framework addresses the early prediction of different physiological signal values relevant in the clinical domain (e.g., the heart rate and the oxygen consumption). CRP is also a more general approach because it supports the body response prediction both at the final and next step of the test.

In this study we present a implementation of the CRP framework focusing on the prediction of the heart rate (HR) and oxygen consumption (VO_2) values, that are important indicators of the individual body response to the test. Specifically, in the current implementation CRP allows predicting (i) the HR and VO_2 values reached at the test end (HR_{peak} and VO_{2peak}, respectively) and (ii) the VO_2 value reached at the step following the prediction step (VO_{2next}). The experimental evaluation of the proposed approach has been performed on a real dataset containing incremental tests for diverse patients. The building blocks of the framework are shown in Figure 4.3 and detailed below.

4.1.1 Data collection and preprocessing

The test execution is characterized by the test protocol and the various physiological signals that are continuously monitored during the test. The CRP framework collects

Signal name	**Abbreviation**	**Measurement unit**
Fraction of inspired oxygen	FIO_2	%
Fraction of expired oxygen	FEO_2	%
Fraction of inspired carbon dioxide	$FICO_2$	%
Fraction of expired carbon dioxide	$FECO_2$	%
Fraction of end-tidal carbon dioxide	$FetCO_2$	%
Fraction of end-tidal oxygen	$FetO_2$	%
Ventilation	VE	l/min
Respiratory rate	RR	$breaths/min$
Inspiratory time	IT	sec
Expiratory time	ET	sec
Heart rate	HR	$beats/min(bpm)$

TABLE 4.1: Monitored physiological signals

data on both the monitored signals and the workload progressively assigned in the text execution.

In the incremental tests considered in this study, the workload is a step signal defined by two parameters: The increment of workload at each step (W_{step}) and the duration of each step (t_{step}) in which the workload is kept constant. These two parameters define the test protocol, denoted $W_{step} \times t_{step}$, meaning that every t_{step} seconds the workload is increased by W_{step} Watt. The protocol is set before the test starts and it is kept constant during the test. The test ends when the individual cannot sustain the current workload. The HR_{peak} and VO_{2peak} values represent the highest values achieved by the individual in the test, for the heat rate and oxygen consumption signals respectively.

During the test execution, various physiological signals are sampled to analyze the patient body response under increasing strain. More specifically, the patient is monitored by means of a set of sensors and a spirometer. Besides the cardiovascular parameters (e.g., the heart rate), the majority of monitored signals describes the patient ventilatory function (e.g., fraction of inspired and expired oxygen). Table 4.1 reports the subset of physiological signals collected in CRP to support the prediction analysis. Since collected signals differ both in scale and measurement unit, a *min-max* normalization step [2] has been performed. This technique is typically exploited in time series analysis [107], because it preserves the original data distribution.

According to [108], the patient oxygen consumption VO_2 (expressed in liters per minute, l/min) during the test execution has been calculated based on the oxygen and carbon dioxide inspired and expired by the patient. VO_2 is computed as

$$\begin{aligned} VO_2 = f_{STPD} \times VE \times [((1 - (FEO_2 + FECO_2)) \,/\, (1 - (FIO_2 + FICO_2))) \\ \times FIO_2 - VE \times FEO_2] \end{aligned} \tag{4.1}$$

where f_{STPD} is the factor of Standard Temperature and Pressure Dry air. f_{STPD} allows comparing values regardless of the temperature and pressure conditions at which they are collected. Based on [108], f_{STPD} can be expressed as

$$f_{STPD} = (273 \,/\, (273 + T_A)) \times (P_{BAR} - P_{H20}) \,/\, (760mmHg) \tag{4.2}$$

where P_{BAR} is the ambient barometric pressure and P_{H20} is the water vapor pressure at temperature T_A. In this study, T_A was assumed 36, and consequently $P_{H20} = 44.6mmHg$.

4.1.1.1 Data stream processing

In cardiopulmonary exercise tests, the exact test duration cannot be specified a-priori because it depends on the patient condition and his/her capability to sustain the progressively rise in workload. Consequently, physiological signals monitored during the test execution should be captured as an unbounded stream. For this reason, the CRP framework has been designed to perform the prediction task through the *data stream analysis over a sliding time window*. Specifically, at each step of the test, one single sliding time window over the original data stream is considered for the prediction task. This window contains a snapshot of the physiological signals monitored in the previous instants of the test. It allows describing the recent past response of the patient to the test, and consequently predict his/her response in the next instants of the test (e.g., the achievable HR_{peak} and VO_{2peak} values).

The sliding time window approach required the definition of the three parameters listed below. (i) The sliding time *window size* parameter (w_{length}) determines the temporal context of interest. A too short time window may focus the prediction task on an almost instantaneous evaluation of the patient condition, since only recently collected data are considered while the previous patient behavior is ignored. Instead, a too large time window allows analyzing many data on past patient behavior, but it may introduce noisy information in the prediction analysis. (ii) The *moving step* parameter (s_t, $s_t \leq w_{length}$) defines how often the window moves, and consequently the step when the prediction is performed. (iii) The *prediction horizon* parameter (h_t) defines the distance between the current sample in the time window and the value to be predicted.

4.1.2 Prediction analysis

During the test execution, the patient cardiopulmonary response to the exercise in the subsequent steps of the test can be predicted using the CRP framework.

The prediction of the physiological signal values achievable in a new ongoing test Q takes place at each time t_p in which the workload is increased. Two types of prediction models, named *single-test* and *multiple-test* model, can be created in CRP. They differ in the reference knowledge base used for model training.

More specifically, for the *single-test* approach, the prediction model is trained only using the *new test* Q currently in execution. Instead, the *multiple-test* model is trained with a *set of previous tests* run with the same protocol of test Q, and reaching a workload value at least equal to the workload of test Q at the prediction step t_p.

The *single-test* approach provides a *tightly tailored model* to the patient response in the ongoing test. The *multiple-test* approach generates an *enriched model* considering responses collected in more tests. To build a suitable model for the currently monitored patient, previous tests showing response to the exercise similar to the patient response in the ongoing test can be considered. For example, tests reaching workload values within a given range can been selected. In this study, we adopted this criterion.

For both *single-test* and *multiple-test* approaches, the prediction process entails the following two main phases.

(i) *Prediction model creation.* A different prediction model is created for each target physiological signal value (e.g., for HR_{peak} and VO_{2peak} value). The prediction model is trained with the ongoing test Q (*single-test* model) or a set of previous tests (*multiple-test* model). For both (*single-test* and *multiple-test*) approaches, the prediction model is trained by considering the physiological signals listed in Table 4.1. These signals are monitored within a sliding time window preceeding the prediction step t_p.

(ii) *Prediction of the physiological signal values.* The (*single-test* or *multiple-test*) model is used to predict the physiological signal values achievable in one subsequent step of the new ongoing test Q. It is possible to predict the signal values reached by test Q in the step following the current (prediction) step (i.e., the *next* step of the test), when the test ends (i.e., the *final* step of the test), or in an *intermediate* step of the test.

Different data mining algorithm can be chosen for the prediction analysis. Among the available techniques suited for the regression problem (i.e., the prediction of a real value as in this study) we selected Support Vector Machines (SVM) and Artificial Neural Networks (ANN) [2]. Both techniques can be used for both regression and classification problems, and they have been widely exploited in many different applications yielding good accuracy performance. The two techniques are briefly presented below while their configuration in the CRP framework is described in Section 4.1.3.

Support Vector Machines (SVM) [2] have been first proposed in statistical learning theory. SVM is able to deal with high-dimensional data and it generates a quite comprehensive (geometric) model. An SVM predictor is based on a kernel function K that defines a particular type of similarity measure between data objects. Examples of kernel functions are linear, RBF (Radial Basis Function), polynomial, or sigmoid kernel. The SVM learning problem can be formulated as a convex optimization problem, in

which different algorithms can be exploited to find the global minimum of the objective function.

Artificial Neural Networks (ANN) [2] simulate biological neural systems. The network consists of an input layer, n hidden layers, and an output layer. Each layer is made up of nodes. Each node in a layer takes as input a weighted sum of the outputs of all the nodes in the previous layer, and it applies a nonlinear activation function to the weighted input. The network is trained with backpropagation and learns by iteratively processing the set of training data objects. For each training data object, the network predicts the target value. Then, weights in the network nodes are modified to minimize the mean squared prediction error. These modifications are made in the backwards direction, that is, from the output layer through each hidden layer down to the first hidden layer.

4.1.2.1 Prediction validation

This block measures the ability of the CRP framework to correctly predict, for a new ongoing test, the physiological signal values achievable in a subsequent step of the test (e.g., when the test ends). To this aim the *absolute prediction error* is computed. It is the absolute difference between the predicted and the actual value of the signal in the test. During the test execution, the signal value is periodically predicted each time an increment of workload occurs, and the corresponding prediction error is evaluated.

In this study, the leave-one-out cross-validation method [2] is used for prediction error evaluation. At each workload increment (i.e., at each prediction step t_p), the subset of tests still running is selected from the dataset. In turn, a different test is picked out of this subset, while the remaining tests are used as knowledge base to predict the considered values. To perform the prediction, the chosen test for the prediction and the reference knowledge base used for the prediction model creation are described by their values within the sliding time window (with size w_{length}). The *Mean Absolute Error* (MAE) [2] at prediction step t_p is the average of the absolute prediction errors computed for all tests in the subset.

4.1.3 Experimental results

This section presents the experimental results for our first implementation of the CRP framework. In the current implementation, CRP allows predicting (i) the HR and VO_2 values reached at the test end (HR_{peak} and VO_{2peak}) and (ii) the VO_2 value reached at the step following the prediction step (VO_{2next}). The *multiple-test* model

Signal Name (unit)	Mean ± SD	MIN	MAX
FIO_2 (%)	0.205 ± 0.0021	0.2	0.21
FEO_2 (%)	0.17± 0.0064	0.15	0.19
$FICO_2$ (%)	0.00096± 0.00025	0.00035	0.0027
$FECO_2$ (%)	0.035± 0.0053	0.017	0.052
$FetCO_2$ (%)	0.05± 0.0058	0.032	0.066
$FetO_2$ (%)	0.152± 0.0092	0.121	0.184
VE (l/min)	28± 12	7	76
RR ($breaths/min$)	23.07± 5.71	6.05	46.75
IT (sec)	1.24± 0.35	0.60	3.48
ET (sec)	1.594± 0.52	0.69	6.58
HR (bpm)	99.61± 20.04	58.17	163.17
VO_2 (l/min)	0.796± 0.28	0.175	1.76
HR_{peak} (bpm)	128.93 ± 15.24	79.33	163.17
VO_{2peak} (l/min)	1.22± 0.18	0.72	1.76

TABLE 4.2: Characteristics of the dataset. For all signals, mean and standard deviation (SD), minimum, and maximum values are reported.

is provided to predict HR_{peak} and VO_{2peak} values, while both *single-test* and *multiple-test* models are available for VO_{2next} prediction. Both *multiple-test* and *single-test* models have been created using both SVM and ANN algorithms. The ability of the CRP framework in correctly predicting the values above is evaluated by analysing the prediction error (MAE) and its distribution. To measure the efficiency of CRP in performing the prediction analysis, the training and prediction time are also discussed.

The experimental evaluation has been performed on a real dataset including several incremental tests for diverse anonymized patients collected at "Exercise pathophysiology laboratory - Cardiac rehabilitation division - Fondazione Salvatore Maugeri IRCCS", Veruno, Italy [109]. Tests have been run using the $5W \times 30sec$ test protocol commonly adopted for the functional evaluation of cardiac patients. The dataset includes 125 tests done by cardiac patients who have reached a maximum workload in the range $[85W \div 110W]$ in the tests. Table 4.2 reports the main data distribution of the monitored physiological signals (introduced in Table 4.1), the computed VO_2 (as described in Section 4.1.1), and the peak values for VO_2 and HR (i.e., VO_{2peak} and HR_{peak}).

In the current CRP implementation, the data stream processing has been implemented by using the Windowing operator available in the RapidMiner toolkit [26]. Both SVM and ANN predictors have been implemented using the corresponding operators available in RapidMiner. The prediction validation block of CRP has been implemented in Python programming language using the X-Validation operator of RapidMiner.

For the results reported in this study, the CRP framework has been configured as follows. For the sliding time window approach, the time window size (w_{length}) has been set to 3, the moving step (s_t) to 1, and the prediction horizon (h_t) to 1 (for VO_{2next} prediction) or to the test duration (for HR_{peak} and VO_{2peak} prediction). For the SVM operator, the RBF (Radial Basis Function) kernel has been selected, and *Gamma*, *C* and *Epsilon* parameters have been set to 0, 500, and 0.001, respectively. To configure

the ANN operator, we set activation function as sigmoid function, training cycles 100, learning rate 0.3, momentum 0.2, *errorepsilon* 1.0E-5, and hidden layer 2. The effect of varying the CRP configuration parameters is discussed in [110]. Experiments have been run on a 2 GHz Intel Centrino Dual-Core PC, with 1 GB of RAM and running Linux kernel 2.6.27.

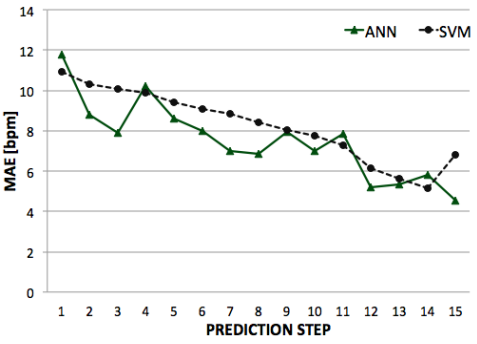
4.1.3.1 Analysis of the prediction accuracy

This section analyses the accuracy of the CRP framework in predicting physiological signal values during the test execution. In all reported charts, step=1 corresponds to the *first step* in the test, when workload 5 W is assigned. The last prediction time corresponds to the achievement of a steady-state heart rate.

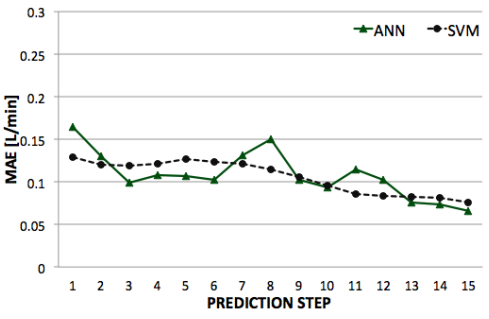
Prediction of HR_{peak} and VO_{2peak}. Figures 4.4(a) and 4.4(b) plot the mean absolute error (MAE) (see Section 4.1.2.1) for the prediction of the HR_{peak} and VO_{2peak} values reached at the test end. The results are promising, as the MAE value is always below 12 bpm for HR_{peak} and below 0.18 l/min for VO_{2peak}, for both SVM-based and ANN-based predictors. For both signals and both predictors, the MAE value decreases when postponing the prediction time and progressively tends to zero.

Experimental results show that the CRP framework would allow to prematurely end a cardiopulmonary exercise test even in the early steps of an incremental protocol, with a limited prediction error on both HR_{peak} and VO_{2peak} values. This would reduce the test duration, thus lowering the body stress of patients without loosing key physiological information (as VO_{2peak} and HR_{peak}) on their response to the test. Importantly, the prediction error of our method does not seem to affect the evaluation of the patient's response to exercise. For example, Figure 4.4(b) shows that estimating VO_{2peak} at step 10 of a cardiopulmonary exercise test, i.e., at 65 W workload, would yield a MAE for VO_{2peak} prediction of about 100 ml/min. In a 75-kg man, this would correspond to 1.3 ml/kg/min, indeed quite an acceptable error for the VO_{2peak} estimate in the clinical setting.

Figures 4.4(a) and 4.4(b) show a decreasing trend on the MAE value on HR_{peak} and VO_{2peak} prediction. This trend is mainly due to the following reasons. (i) The error is higher in the early steps because the reference knowledge base used for prediction contains the majority of the considered dataset. Consequently, tests with different durations (i.e., tests with workload in the range $[85W - 110W]$) contribute to the prediction task. Later, the prediction becomes more accurate because the reference knowledge base tends to progressively include a subset of tests with similar durations. (ii) When postponing



(a) *multiple-test* model for HR_{peak} prediction: MAE by varying the prediction time



(b) *multiple-test* model for VO_{2peak} prediction: MAE by varying the prediction time

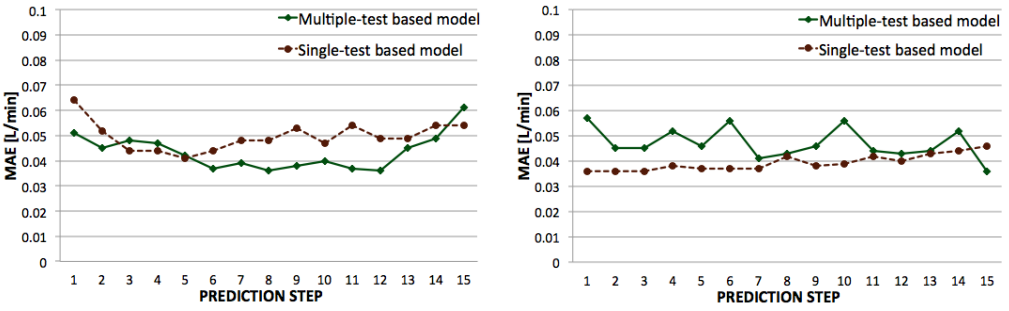
FIGURE 4.4: *multiple-test* model for HR_{peak} and VO_{2peak} prediction

the prediction time, the prediction horizon (i.e., the time interval between the prediction step and the test end) reduces and tends to zero. Thus, the body response at the prediction step tends to get closer to the response at the test end. Because of these two conditions, the prediction of both HR_{peak} and VO_{2peak} is initially affected by a larger, but limited, error. We can also observe that in both Figures 4.4(a) and 4.4(b) MAE curves are more irregulars and not strictly decreasing for the ANN predictor. It follows that, the ANN prediction model is more sensitive than the SVM one in considering a reference knowledge base including tests with different durations.

Prediction of VO_{2next}. As a reference example of the next step prediction task, the VO_{2next} prediction is reported in this study. Both *single-test* and *multiple-test* models are considered. Results, for SVM and ANN prediction algorithms, are reported in Figures 4.5(a) and 4.5(b) respectively. For both models, the MAE value is very low (in the range $0.035 \div 0.065$ l/min), being the horizon prediction always equal to 1 step.

For the SVM predictor (Figure 4.5(a)) the *multiple-test* approach slightly improves the *single-test* except for few prediction times, showing that a larger reference knowledge base can increase the accuracy of the model. Instead, the ANN predictor shows an opposite trend (Figure 4.5(b)), since the *single-test* approach is better than the *multiple-test* one. However, in both cases the prediction error is limited for all prediction times.

Performance evaluation. For the considered dataset, the ANN predictor requires a long training time (i.e., more than 12 hours) to build the prediction model, while SVM requires about 3 hours. Once the model is defined, prediction is very efficient (i.e., few seconds) for both predictors.



(a) VO_{2next} prediction using SVM: MAE by varying the prediction time

(b) VO_{2next} prediction using ANN: MAE by varying the prediction time

FIGURE 4.5: VO_{2next} prediction using SVM and ANN

4.2 Analysis of User-Generated Data in bike-sharing systems

Bicycle sharing systems are eco-friendly transportation systems that have found wide application in Smart urban environments. In bicycle sharing systems, a key performance indicator to monitor is the number of occupied slots per station, which corresponds to the number of parked bicycles. Monitoring the level of occupancy of a station could highlight critical situations, which may lead to service disruption. In this section we focus on predicting and characterizing critical situations in which station occupancy levels are relatively low, i.e., situations in which there is a lack of parked bicycles. These conditions are unpleasant, because users are likely to have no bicycles available to rent from the station and, thus, they have to move to another one. Consequently, this service disruption can discourage citizens from using bike sharing systems.

This section presents STation Occupancy Predictor (STOP), a new data mining-oriented system aimed at analyzing the occupancy levels of the stations of bike sharing systems and predicting critical occupancy levels. For each station, STOP acquires the number of occupied slots at a given sampling rate. Historical data are then analyzed by means of classification techniques to build a model aimed at predicting at a given time instant t_p, the upcoming occupancy level (i.e., critical or non-critical) of the station at a time instant in the near future (i.e., at time $t_p + \gamma$, where γ is a limited time horizon). The predicted occupancy level will be exploited to support domain experts in making appropriate decisions and planning maintenance activities in the short and medium term. This prediction allows the system managers to discover such critical situation in advance and, thus, to prevent service disruption. For example, system managers could plan the re-balancing of the bicycles in the stations or the resizing of the total number of

bicycles available in the system. As a case study, STOP has been thoroughly evaluated on real data acquired from the bicycle sharing system of New York City.[1]

Different research directions have been pursued to analyze bicycle-based transport data. For each direction, different works have been proposed and different facets of the problem have been addressed. Previous research activities addressed the following issues: (i) predicting the occupancy levels of the stations [111, 112], (ii) repositioning bicycles in the stations [113, 114, 115], (iii) advising services to support user navigation of city through the bike-sharing system [116], (iv) analyzing imbalances in bicycle distribution among stations [114, 117], (v) characterizing urban human mobility and activity patterns through the analysis of social network data [118, 119], (vi) analyzing the trip data of the bike sharing system [117, 120], and (vii) evaluating the performance of bicycle sharing systems [121, 122]. In the research work we focus on issue (i).

The authors in [111, 112] analyzed the occupancy values of the stations of the bicycle sharing system in Barcelona to discover temporal and geographic mobility patterns as well as to predict the number of available bicycles for any station in the near future. The work presented in [112] proposed to use regression-based models, i.e., a simple baseline model, a gradient-based prediction model, auto-regressive moving average (ARMA) model, to predict the number of available bicycles. For each station a regression model is built to predict its bicycle availability based on the characteristics of the station itself and its nearby stations. Instead, in [111], clustering and regression techniques are applied to identify shared behaviors across stations and to characterize these behaviors with location, neighbourhood, and temporal information. Unlike [111, 112], in our study we address the problem of predicting critical occupancy levels of the stations of the bike sharing system through interpretable yet accurate classification models (rather than regression techniques). These models can help domain experts understand the underlying events and thus avoid service disruption. Classification algorithms have successfully been exploited in different application contexts (e.g., mobile wireless network [123], activity recognition [124]). Among the available classification methods those generating interpretable models [125] are preferred in real application scenarios, because they allow experts to gain insights into the classifier predictions. For this reason, the framework proposed in this study combines the accuracy of prediction models with the interpretability property of associative classifiers.

The main architecture blocks of the STOP framework are depicted in Figure 4.6, and a more detailed description of each block is given below.

[1]The research leading to these results was partially funded by the Italian Ministry of Research (MIUR) under the Smart Cities and Communities Grant Agreement n. SCN_00325 (Project s[m2]art).

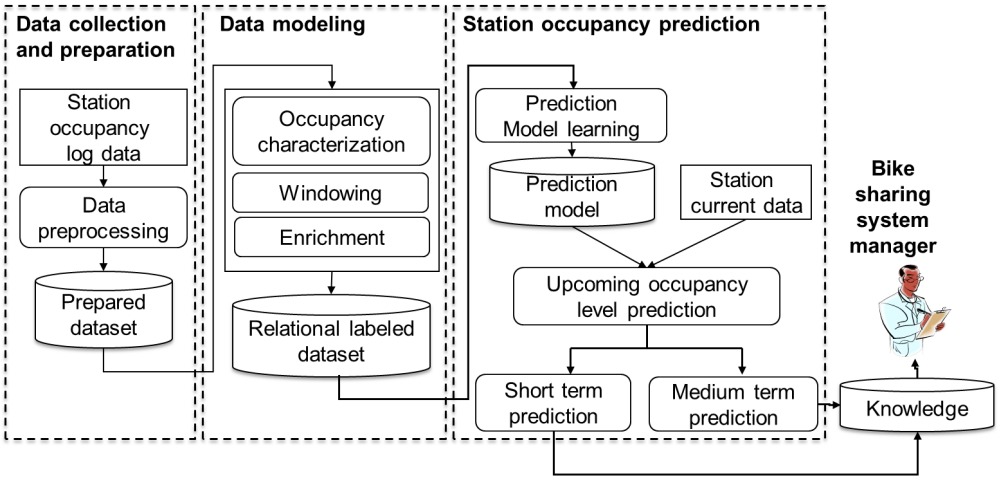


FIGURE 4.6: The STation Occupancy Predictor architecture.

4.2.1 Data collection and preparation

Station occupancy measurements are acquired at a given sampling rate for all the stations in the bicycle sharing system and then collected into a unique data repository.

Let S be the set of stations in the bicycle sharing system and $c(s_j)$ be the capacity of station $s_j \in S$ (i.e., the total number of available slots). For each station $s_j \in S$, the STOP framework considers the following two complementary data features. (i) The *number of occupied slots* $o(s_j, t_i) \leq c(s_j)$ at timestamp t_i, which corresponds to the number of parked bicycles at a given instant of time and indicates the level of usage of the station. (ii) The *timestamp* t_i (*date* and *time*) at which the occupancy measurement $o(s_j, t_i)$ was acquired.

The *number of occupied slots* $o(s_j, t_i)$ is an interesting feature because low values imply a lack of parked bicycles at the station and, thus, a potential service disruption. For example, bikers who plan to leave from an almost empty station may not find any free bicycle to rent from there at their arrival. These issues can be tackled by scheduling targeted maintenance actions. For instance, bicycles can be re-balanced among stations.

Timestamp t_i is worth considering to predict potentially critical situations, because the numbers of occupied slots per station are likely to be temporarily correlated with each other.

While monitoring the number of occupied slots, a few readings could be missed. To tackle this issue, various approaches have been proposed to replace missing data values with reliable estimates (e.g. the average, the mode) [126]. The STOP system replaces the missing values occurring in the acquired data with the average measurement reading over all the sampling time instants.

4.2.2 Data modeling

In this phase, the STation Occupancy Predictor framework prepares data to the next classification process. Each sampled measurement is labeled with a class label (critical or non-critical). Next, a *windowing technique* is applied to sample the acquired measurements according to the corresponding timestamps. Then, station occupancy measurements are transformed into a relational data representation to support the subsequent classification phase.

4.2.2.1 Critical occupancy categorization

The measures on station occupancy acquired at a given time instant are classified as the corresponding occupancy level (critical or non-critical), according to an (analyst-provided) occupancy threshold *thr*. More specifically, the number of occupied slots per station can be labeled as follows: (i) *Critical*, if the percentage of occupied slots in a station is below a given (analyst-provided) occupancy level threshold *thr*, or (ii) *Non-critical*, if the percentage of occupied slots is equal to or above *thr*. In bicycle sharing systems, a critical occupancy level (i.e., the lack of bicycles) per station may indicate a suboptimal system usage, which can be addressed either by planning targeted maintenance actions or by increasing the number of bicycles in the system.

4.2.2.2 Data windowing

The STOP framework addresses the prediction of the future occupancy level (critical or non-critical) for each station in the bike sharing system. Let us consider a station $s_j \in S$ in the system. We define as *prediction time* t_p the time instant at which the STOP system predicts, for station s_j, the occupancy level (critical or non-critical) at a subsequent instant t_f ($t_f > t_p$). The time gap $\gamma = |t_f - t_p|$ defines the *prediction horizon*.

The numbers of occupied slots per station are likely to be temporarily correlated with each other. Consequently, for each station s_j and prediction time t_p we consider not only the current number of occupied slots but also a snapshot of occupancy data acquired at the previous instants. This procedure allows considering the temporal evolution of the number of occupied slots while predicting the future occupancy level of a station. More formally, given a station s_j and a prediction time t_p, we define a time window $W(s_j, t_p)$ including an ordered sequence of WL timestamps. This sequence includes prediction time t_p and $WL - 1$ timestamps preceeding t_p, i.e., t_{p-1},t_{p-2},...,t_{p-WL-1}. For each timestamp t_k in window $W(s_j, t_p)$, the measurement o(s_j, t_k) on the station occupancy

is considered. For the sake of simplicity, hereafter we will assume to sample the time window uniformly, i.e., $\forall\ t_i, t_{i-1} \in W$ the sampling interval $\Delta{=}t_i - t_{i-1}$ is fixed.

The adopted windowing approach requires the definition of the two parameters listed below. (i) The *time window size* (WL), which determines the width of the learning time period. Typically, medium window sizes are preferable. On the one hand, a too small window may bias the quality of the prediction, because only the most recent occupancy measurements are considered. On the other hand, a too large window may include also past sampling time instant which are almost uncorrelated with the prediction time t_p. (ii) The *prediction horizon* (γ), which defines the temporal distance between the sampling time window and the time instant at which the station occupancy level must be predicted. Small/medium γ values are preferable. In fact, considering large γ values make the prediction task more complex, because future occupancy values are less likely to be correlated with past ones.

4.2.2.3 Data enrichment

The station occupancy level (critical and non-critical) may be related to various temporal factors, such as the day category (holiday or working day), the day of the week, and the daily timeslots. To consider the temporal information at different granularity levels during occupancy level prediction and characterization, in the STOP system the time instant at which the prediction is performed is enriched with additional information characterizing its temporal context.

A temporal hierarchy is created to aggregate time and date information into higher-level concepts. Times are generalized as hourly timeslot, 2-hour timeslot, and 4-hour timeslot. 4-hour timeslots partition daily periods into *morning* ([4am-8am),[8am-12am)), *afternoon* ([12am-4pm),[4pm-8pm)), and *evening* ([8pm-12pm),[12pm-4am)). 2-hour and 1-hour timeslots further partition each 4-hour timeslot into shorter periods. Dates are generalized as the corresponding day of the week and as working day or high day. High days comprise both weekends and the most important feast days (e.g. Christmas, Easter, New Year's day).

4.2.2.4 Data representation

This phase transforms station occupancy data into a format suitable for the subsequent data analyses. To this aim, for each station of the system the station occupancy measurements and temporal information presented in Sections 4.2.2.1, 4.2.2.2, and 4.2.2.3 are modeled as a relational dataset [2].

A *relational dataset* is a set of records. A *record* is a set of items, where an *item* is a pair *(attribute_name, value)*. While *attribute_name* is the description of a specific data feature, *value* is the collected information. In our context, each record corresponds to a different prediction timestamp t_p for a station s_j in the system. The record is structured as follows.

Definition 4.1. Relational station record. Let $s_j \in S$ be an arbitrary station and t_p be a given prediction timestamp. Let $W(s_j, t_p)$ be the time window of size WL for the pair of station s_j and timestamp t_p. Let t_p, t_{p-1}, ..., t_{WL-p-1} be the ordered sequence of timestamps in W. Record $r(t_p, s_j)$, corresponding to prediction timestamp t_p, is characterized by the following attributes:

- prediction timestamp t_p
- attributes describing the station occupancy at the different sampling instant t_k in time window $W(s_j, t_p)$, i.e., number of occupied slots at timestamps t_p, t_{p-1}, ..., t_{p-WL-1}
- class, which indicates the occupancy level (critical/non-critical) of station s_j in the near future, i.e., at timestamp $t_p + \gamma$.
- attributes describing the temporal context at timestamp $t_p+\gamma$, i.e., hourly, 2-hour, and 4-hour timeslots, day of the week, day category (high day, working day)

Let us consider an example record corresponding to the prediction timestamp t_p for station s_j. By considering a window size with length WL=2, the record stores the measurements acquired at the prediction time and one preceding timestamp. These values are stored in attributes *num. of occupied slots* t_p and *num. of occupied slots* t_{p-1}. The class attribute indicates the future occupancy level (critical/non-critical) at timestamp $t_p + \gamma$. The remaining record attributes describe the temporal context at $t_p + \gamma$. For example, the corresponding pairs (attribute,value) are (*hourly timeslot*, *[4pm,5pm)*), (*2-hour timeslot*, *[4pm,6pm)*), (*4-hour timeslot*, *[4pm,8pm)*), (*Day of the week*, *Monday*) , (*Day category*, *Working day*).

For classification purposes, a station record is a *training record* when the value of the class attribute is known. Instead, any new station record for which the station occupancy level at an upcoming time instant is unknown is called *test record*. The subset of training records representing the historical measurements of station occupancy for a station s_j is called training dataset.

Definition 4.2. Relational training dataset. A relational training dataset $D(s_j)$ corresponding to station s_j is the set of all training records $r(t_p, s_j)$ corresponding to station s_j for all considered prediction times t_p.

4.2.3 Station occupancy prediction

The aim of this step is twofold. (1) *Training phase.* for each station a classification model is generated by applying a classification algorithm on a collection of historical measurements of station occupancy. (2) *prediction phase.* The upcoming occupancy level (i.e., the critical or non-critical labels of the class attribute) of each station is predicted by applying the classification model. The training phase is applied to all the training records collected in training dataset (see Definition 4.2). Conversely, the prediction phase is applied to all new records, called *test record*, for which the station occupancy level is unknown.

To generate the classification model, many different techniques are available in literature (e.g. Bayesian classifiers [127], associative classifiers [128], Support Vector Machines [129], Decision trees [130]). To perform occupancy level prediction, the STOP system can straightforwardly integrate many classification algorithms. To select the most appropriate classification algorithms to use, the following two complementary aspects have been considered: (i) the ability of the classifier to *accurately predict* the station occupancy level, and (ii) the *interpretability* of the generated model. The former aspect (i) is considered because occupancy level predictions must be as much accurate as possible to minimize the bias due to classification errors. Therefore, we selected the most performant classifiers among a large set of established algorithms. The latter property (ii) implies the readability of the generated prediction models. If the generated models are humanly readable, domain experts can not only exploit the result of the prediction phase but also manually explore the output of the training phase to pinpoint potentially useful information. This aspect is further discussed in Section 4.2.4. Associative classifiers and rule induction algorithms are typical examples of classifiers that produce humanly readable models.

An extensive experimental evaluation, reported in Section 4.2.5, was performed on data acquired from a real bicyle system. The results demonstrated that (i) the Averaged One Dependence Estimators with Subsumption resolution (AODEsr) [131] classifier performed best, according to established performance indices (e.g. average accuracy and F1-measure [2]), in predicting the critical class compared to several other classification algorithms, Bayesian and not. (ii) the Live and Let live (L^3) [128] associative classifier is the most competitive algorithm among those generating interpretable classification models. Therefore, based on the achieved results, in this study the STOP system integrates the Bayesian AODEsr classifier as the most accurate classification strategy on the analyzed data, and the associative L^3 classifier as the approach that achieved the best trade-off between model accuracy and interpretability. To the best of our knowledge, AODEsr and L^3 are the most recent and accurate classifiers on structured data among

those belonging to the respective categories. A brief description of the two selected classification approaches is given below.

AODEsr. Bayesian classifiers are statistics-based classifiers, which predict class membership probabilities, such as the probability that a given sample belongs to a particular class by exploiting the Bayes theorem [127]. The pioneer Bayesian classifier, namely Naive Bayes [127], assumes the conditional indipendence of all data attributes given the class. Bayesian Networks are more recent and accurate Bayesian approaches that consider also the class-conditional dependences between pairs of attributes [2]. Among them, the Averaged One-Dependence Estimators with Subsumption Resoluton (AODE) classifier is, to the best of our knowledge, the most accurate Bayesian Network on structured data. More details on the classification model can be found in [132].

L^3. Associative classifiers rely on association rules, which represent implications between multiple data items that frequently hold in the analyzed data [81]. The subset of associations rules that are worth considering for classification purposes, called *classification rules*, are in the form R: $A \rightarrow C$. A is the rule antecedent and it consists of a set of items belonging to different attributes, while C is the rule consequent and it consists of a single item belonging to the class attribute, i.e., (Class, Critical) or (Class, Non-critical) in this study.

For example, {(*Day of the week*, *High Day*)} $\rightarrow$ (*Class*, *Critical*) is a classification rule stating that if attribute *Day of the week* assumes value *High Day* then the test record should be labeled as *Critical* because this implication frequently co-occur in the analyzed data. More complex rules can be extracted as well. For example, rule {(*Day of the week*, *High Day*), (*Hourly timeslot*, *10pm*)} $\rightarrow$ (*Class*, *Critical*) is a specialization of the former one stating that if the day of the prediction is an high day and the timeslot is between 10pm and 11pm, then the class label is *Critical*.

In the L^3 classifier [128] a subset of most interesting classification rules is generated. Each rule is characterized by two main quality indices: support and confidence. The support of a classification rule R: $A \Rightarrow C$ in a relational dataset D, denoted as sup(R,D), is defined as the observed frequency of the set of items $A \cup C$ in D. The confidence of R in D, denoted as conf(R,D), is the conditional probability of occurrence in D of $A \cup C$ given A, i.e., $\frac{sup(R,D)}{sup(A,D)}$.

Let us consider an example rule {(*Day of the week*, *Working Day*)} $\rightarrow$ (*Class*, *Critical*). Rule has support equal to 10% because it occurs in a tenth of records in the training dataset. Its confidence is 66%, meaning that 66% of the training records containing item {(*Day of the week*, *Working Day*)} are labeled with class *Critical*.

To classify a new test record, rules in the model are sorted in order of decreasing confidence and support. Then, the top ranked rule matching the test record is used to assign the class label. For example, rule $\{(Day\ of\ the\ week, High\ Day)\} \rightarrow (Class, Critical)$ can be applied if attribute *Day of the week* of the test record assumes value *High Day*.

4.2.4 System exploitation

In this section we envision how managers of bike sharing systems can exploit the STOP framework to improve service provision. Two main features of the classification models will be exploited: (i) the *accuracy* in the prediction of the upcoming occupancy level of a station, and (ii) the *interpretability*, which allows domain experts to explore the rules generated by the associative classification algorithm to perform advanced analyses.

Short term occupancy level prediction. Accurate predictions of station occupancy levels can be exploited in the short term to react to potentially critical situations. Let us suppose that the STOP system predicts an upcoming critical occupancy level for station s_j in the next 30 minutes This implies a lack of available bikes in a station of s_j, which may cause a service disruption, because users reaching station s_j could not rent a bicycle from that station and need to move to another one. Based on the predictions made by the STOP system, system maintainers can promptly react in the short term to this type of service malfunctioning. For example, they can plan re-balancing actions of the number of bicycles per station by sending an employee of the assistance service to move bicycle from non-critical stations to critical ones. However, maintenance actions can be costly and time consuming.

To effectively plan re-balance actions, the schedule of maintenance actions must be optimized. For example, at a given time instant, the STOP system may predict multiple critical situations related to different stations with a prediction horizon equal to 30 minutes. To schedule maintenance actions, a priority level should be assigned to each critical condition based on the output of the prediction algorithm. According to the type of classification algorithm used, different strategies can be adopted.

Bayesian classifiers compute a conditional probability $P(c_i|T)$ of occurrence for each class label (Critical/Non-critical) given the test record generated at the prediction time instant. The class label with maximal probability is assigned to the given test record. To identify most critical situations and design a smart action plan, a AODEsr classification model can be first generated for each station in the system. Then, stations for which a critical occupancy level is predicted can be ranked by decreasing probability $P(c_i|T)$.

Associative classifiers assign the class label according to the classification rule that best fits the given test record. Specifically, in the L^3 classifier rules with maximal confidence are preferred. The rule confidence, which indicates the conditional probability of occurrence of the class label given the occurrence of some peculiar record items, is selected as the most discriminative ones for classification purposes. Therefore, to identify most critical situations and to react with targeted actions, a L^3 classification model can be first generated for each station in the system. Then, stations for which a critical occupancy level is predicted can be ranked by decreasing confidence of the classification rule used to make the prediction.

Medium term characterization of critical conditions. To support domain experts in decision-making, the property of readability of the rule-based models generated by the associative classifiers can be profitably exploited. Classification rules not only predict an upcoming critical occupancy level of a station, but also describe the context in which the identified criticality frequently occurs. These rules can be exploited, for instance, to plan maintenance actions in the medium term.

For example, classification rule {(*Day of the week*, *Working Day*), (*2-hourly timeslot,[3pm,4pm)*)} $\rightarrow$ (*Class*, *Critical*), Confidence=100%, indicates that, for a given station, a critical occupancy level occurs in 100% of the working days in the hourly timeslot [3pm,4pm). The rule indicates an interesting recurrence in the occurrence of critical levels, which depends only on day category and timeslot. This rule can be considered for planning periodic re-balance actions. For example, system managers can plan a re-balance action for every working day before 3pm. Based on the classification rules selected by the L^3 classifier, domain experts can characterize the temporal distribution of the critical conditions occurring in each station and refine maintenance actions accordingly. Note that the maintenance actions triggered by the exploration of classification rules can be planned many hours/days before the critical events occur, whereas, based on short time predictions, critical conditions can be addressed only immediately before they occur.

Different types of rules can be considered, eventually organized in a hierarchical fashion, to plan maintenance actions. Specifically, we identified the rule types reported in Table 4.3.

The rules involving the *Day category* attribute are the more general ones since they represent critical conditions that hold for many days. They can be used to plan long term actions that can be applied in all days of the specific category reported in the rule immediately before the begin of the timeslot involved in the rule. Among the rules involving *Day category*, the system administrator will first analyze the ones associated

Rule type
{(*Day category*, ...), (*4-hour timeslot*, ...)} → (*Class*, *Critical*)
{(*Day category*, ...), (*2-hour timeslot*, ...)} → (*Class*, *Critical*)
{(*Day category*, ...), (*Hourly timeslot*, ...)} → (*Class*, *Critical*)
{(*Day of the week*, ...), (*4-hour timeslot*, ...)} → (*Class*, *Critical*)
{(*Day of the week*, ...), (*2-hour timeslot*, ...)} → (*Class*, *Critical*)
{(*Day of the week*, ...), (*Hourly timeslot*, ...)} → (*Class*, *Critical*)

TABLE 4.3: Rule types.

with a 4-hour timeslot because they provide more general knowledge than those involving attributes *2-hour timeslot* or *Hourly timeslot*.

Also the rules involving the *Day of the week* attribute, combined with a timeslot, are useful for long term action planning. However, since they are associated to a specific day, they are less general than the ones based on *Day category*. Hence, they should be considered after the former ones.

In conclusion, while occupancy level predictions provide useful hints for system managers in the short term, rule-based model inspection can be performed to characterize the distribution of the station occupancy levels in the medium/long term. As discussed in Section 4.2.5, the L^3 associative classifier generated many classification rules potentially useful for system usage characterization on the analyzed datasets.

4.2.5 Experimental results

The usability and effectiveness of the proposed STOP framework have been validated on a dataset storing the occupancy values of the stations of a real bike sharing system. We analyzed the following aspects: (i) the accuracy of the occupancy level predictions and the interestingness of the rule-based classification models, (ii) the impact of the system parameters, (iii) the choice of the classification algorithms integrated in the system, (iv) the use of regression-based algorithms, rather than classification approaches, to perform occupancy level prediction, and (v) the execution time taken by the STOP system.

Dataset. As reference case study we considered a real station occupancy dataset acquired from the *Citi Bike* system of New York. *Citi Bike* features thousands of bikes about 500 stations across New York and Jersey City. Bicycles are available 24/7, 365 days a year. Per-station occupancy values were acquired every 5 minutes over a time period of approximately 13 months (i.e., between October 23rd 2014 and November 17th, 2015). More information about the system and a map of the stations available at https://member.citibikenyc.com/.

We focused our analysis on the 50 largest stations, which correspond to the mostly used ones and, thus, are most likely to be in a critical situation. We performed experiments on three different time periods of year 2015 (April-May, June-July, September-October). To consider different data distributions, each period falls in different season. For each time period, data corresponding to the first month was used to train the per-station classification models, while data corresponding to the second month was used to test the classifiers.

Evaluation setup. In the current STOP system implementation, we used the AODEsr algorithm version available in the RapidMiner toolkit (https://rapidminer.com) and the L^3 classifier version provided by the respective authors [128]. Experiments were performed on a 2.66-GHz Intel(R) Core(TM)2 Quad PC with 8 GBytes of main memory, running linux (kernel 3.2.0). Whenever not otherwise specified, hereafter we consider the following parameter configuration for the STOP system: window size WL=7 previous occupancy samples, prediction horizon γ=60min, and occupancy level thresholds thr=10%.

Quality measures. To validate the results of the classification process we considered established quality measures [2]. Specifically, per-class classifier predictions were evaluated according to Recall, Precision, and F1-measure. For the sake of completeness, in our experiments we computed the statistics for both the critical and non-critical classes. However, in our context, the focus of the analysis is mainly on the critical class, because to plan maintenance actions and improve the quality of the offered service accurately predicting critical situations is definitely more important than recognizing normal system conditions. For this reason, Recall(critical), Precision(critical), and F1(critical) are the reference quality measures that we consider in our discussion.

Recall(critical) indicates how many critical situations (i.e., stations with a lack of parked bicycles) the system is able to predict with respect to the total number of critical situations that actually occurred. Conversely, Precision(critical) indicates the number of critical situations that actually revealed themselves as critical. To avoid service disruption, the bicycle sharing system administrator should be warned of all the critical situations and plan balancing actions that increase the number of parked bicycles in the stations in a critical situation. Hence, achieving a Recall(critical) as much high as possible is desirable. On the other hand, Precision(critical) must be taken into consideration as well. In fact, a low Precision(critical) value indicates that many non-critical situations were mistakenly classified as critical. This yields a waste of time and money. In summary, a good trade-off between recall and precision is needed to properly handle critical situations in real bicycle sharing systems. The F1-measure [2], which is computed

as the harmonic average of precision and recall, quantitatively estimates the balancing between recall and precision.

4.2.5.1 Use-case analysis

We analyzed the occupancy level predictions of the STOP system on the real dataset with two complementary goals: (i) short term prediction of the occupancy levels, and (ii) medium term characterization of the critical conditions occurring in the stations.

Analysis of the quality of short term predictions on use case data. Table 4.4 reports the results achieved by the AODEsr and L^3 classifiers integrated in the STOP system. We reported Recall, Precision, and F1-measure for both classes (i.e., critical and non critical). However, as discussed in the evaluation setup, our main focus is on class critical.

April-May time period						
Alg.	**R(Cr)%**	**P(Cr)%**	**F1(Cr)%**	**R(NCr)%**	**P(NCr)%**	**F1(NCr)%**
AODEsr	70.7	74.1	72.3	85.5	84.9	85.2
L^3	62.9	74.0	68.0	87.2	81.6	84.3
June-July time period						
Alg.	**R(Cr)%**	**P(Cr)%**	**F1(Cr)%**	**R(NCr)%**	**P(NCr)%**	**F1(NCr)%**
AODEsr	76.1	74.2	75.1	88.1	90.2	89.1
L^3	72.7	75.8	74.2	88.9	88.7	88.8
September-October time period						
Alg.	**R(Cr)%**	**P(Cr)%**	**F1(Cr)%**	**R(NCr)%**	**P(NCr)%**	**F1(NCr)%**
AODEsr	81.7	72.1	76.6	87.9	90.9	89.4
L^3	75.9	73.9	74.9	87.3	90.9	89.0

TABLE 4.4: Prediction quality of STOP in different time periods by using AODEsr and L^3.

Our approach achieved good results for every considered time period and classification algorithm. For example, in September-October, AODEsr achieved a Recall(critical) equal to 81.7%, meaning that 81.7% of the critical situations (i.e., almost empty stations) that occurred in the considered time periods were predicted by the STOP system and thus could be avoided through appropriate maintenance actions. Furthermore, the Precision(critical) value is 72.1%, meaning that in approximately 7 out of 10 cases the predictions made turned out to be correct. Thus, the cost of handling false positive alarms is limited. Overall, the balancing between precision and recall for class critical is satisfactory.

Based on the results reported in Table 4.4, both classifiers performed high-quality predictions. AODEsr performed slightly better than the L^3 associative classifier. Hence, to produce short-term occupancy level prediction using the Bayesian classifier is the best choice on the analyzed data. However, as discussed below, the classification model

generated by L^3 is easily interpretable by domain experts. Therefore, it can be used to characterize the critical situations and to schedule medium term maintenance actions.

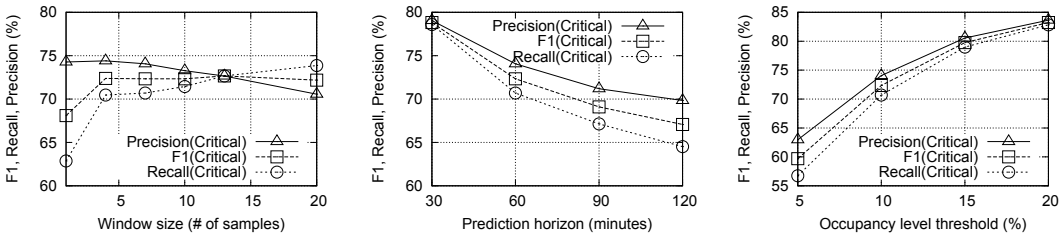
Medium term characterization of critical conditions. The predictions made by the STOP system in the short term (temporal horizon in the range [30m, 120m]) allow the system manager to discover upcoming critical conditions in station occupancy levels. However, as discussed in Section 4.2.4, system managers are also interested in planning a medium term schedule of maintenance actions. To address this issue, the rule-based model generated by the L^3 classifier can be exploited. For example, the classification rules can be manually explored at the beginning of the week to plan the daily schedule of the major maintenance actions.

Id	Rule	Conf.%	Sup.%
1	$\{(day\ category, high\ day), (hour, [1pm - 2pm))\} \rightarrow critical$	100	0.5
2	$\{(day\ category, working\ day), (2 - hour\ timeslot, [16 - 18))\} \rightarrow critical$	87.2	2.4
3	$\{(dayOfWeek, Wednesday), (4 - h\ timeslot, [4am, 8am)])\} \rightarrow critical$	100	1.2
4	$\{(dayOfWeek, Monday), (4 - h\ timeslot, [4am, 8am)])\} \rightarrow critical$	98.6	0.9
6	$\{(dayOfWeek, Saturday), (4 - h\ timeslot, [8am, 12am))\} \rightarrow critical$	94.4	0.9
5	$\{(dayOfWeek, Monday), (4 - h\ timeslot, [12am, 4pm)])\} \rightarrow critical$	92.8	1.1
7	$\{(dayOfWeek, Friday), (2 - hour\ timeslot, [10am - 12am)])\} \rightarrow critical$	100	0.5
8	$\{(dayOfWeek, Tuesday), (2 - hour\ timeslot, [2pm - 4pm))\} \rightarrow critical$	97.9	0.6
9	$\{(dayOfWeek, Tuesday), (hour, [3pm - 4pm))\} \rightarrow critical$	100	0.3
10	$\{(dayOfWeek, Friday), (hour, [1pm - 2pm))\} \rightarrow critical$	94.4	0.2

TABLE 4.5: Representative classification rules for Station 423 (Apr-May).

Table 4.5 reports some representative rules, which were selected from the classification model corresponding to Station with id 423. Similar rules appear in the classification models of the other analyzed stations. For instance, rule (1) in Table 4.5 involves the day category and the hourly time slot attributes. The rule states that during high days the station is always in a critical condition (confidence=100%) in the time slot [1pm-2pm). Based on this rule, specific monitoring/maintenance activities targeted to Station 423 can be scheduled every high day before 1pm. Rules with maximal confidence value (100%) are implications that always hold in the training dataset. Conversely, for lower-confidence rules (e.g., rules (2) and (10) in Table 4.5) some exceptions occur in the training data. To plan an effective medium term schedule, high-confidence rules should be preferred, because they appear to be more reliable according to historic data.

Execution time. The most computationally intensive and time consuming task accomplished by the STOP system is classification model learning. Hence, we measured the time spent by the AODEsr and L^3 classifier in model learning. The per-model training time ranges between 1s (AODEsr) and 2.3s (L^3). Hence, it is acceptable in our context of analysis. The prediction time per data is on average few milliseconds for both algorithms.



(a) Window size (b) Prediction horizon (c) Occupancy threshold

FIGURE 4.7: Effect of the STOP system parameters (Apr-May).

4.2.5.2 Analysis of impact of the system parameters

We analyzed the effect of the following parameters on the quality of the occupancy level prediction: (i) *window size* (WL), which indicates the number of sampled past measurements, (ii) the *prediction horizon* (γ), which indicates the gap between the current and predicted time instant, and (iii) the occupancy level threshold (thr), which categorizes the station occupancy values as critical or non-critical. For each parameter we tested the models generated by both classifiers by varying the values of one parameter at a time from the standard configuration. Classifier performance was evaluated in terms of Precision, Recall and F1-measure of the critical class. Here we discuss the results achieved by the AODEsr classifier on the April-May dataset as a representative scenario. Similar trends were achieved by the other classifier and on the other datasets.

Window size. Figure 4.7(a) shows the impact of the window size on the classifier performance. When WL=1 only the station occupancy value at the prediction time is considered, while increasing the value of WL the past occupancy values are taken into account during prediction model learning. Since samples were acquired every 5 minutes, the time gap corresponding to the analyzed window sizes ranges between 0 and 90min. The recall of class critical is relatively low when the past values are ignored (WL=1) and it becomes fairly high for window sizes in the range [5,13]. Conversely, the precision is relatively stable up to WL=7, then it linearly decreases while increasing the window size. Therefore, when the window size is not so large as to include out-of-date measurements, the more historic data we consider the more effective the system becomes in predicting critical situations. To achieve the best balancing between precision and recall in our context of analysis, we recommend domain experts to set the window size to 7.

Prediction horizon. Figure 4.7(b) shows the impact of the time horizon parameter γ on the classifier performance. Increasing the value of γ implies increasing the temporal distance between the time window analyzed during the training phase and the point of time at which we would like to predict the station occupancy level. The quality of the

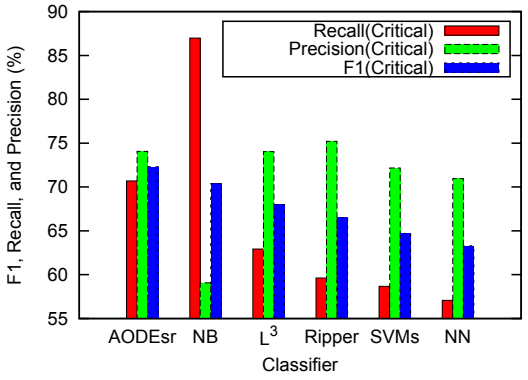


FIGURE 4.8: Impact of the classification algorithm on the prediction quality.

prediction decreases while increasing the value of the time horizon parameter, because since the data distribution changes over time, the larger the time gap the less likely the model fits the test data. We recommend domain experts to set γ to 30 min, because this parameter setting allows achieving fairly good results (recall=78.6%, precision=79.1%).

Critical occupancy threshold. Figure 4.7(c) shows the impact of the occupancy threshold *thr* on the classifier performance. The higher the occupancy threshold the more likely stations may incur in critical situations and, thus, the problem of predicting critical occupancy levels becomes simpler. For this reason, the prediction quality increases while increasing the value of *thr*.

4.2.5.3 Impact of the classification algorithm on the prediction quality

We compared the performance of different well-known classifiers of different categories on the occupancy data acquired by the STOP system. Specifically, we selected two Bayesian classifiers (AODEsr and Naive Bayes (NB) [127]), one associative classifier (L^3), one rule induction approach (Ripper [133]), a Support Vector Machine classifier (SVMs [129]), and one neural network technique (NN [134]). The experiments were performed on the April-May dataset using the standard parameter setting. The results, in terms of F1-measure, Recall, and Precision, are summarized in Figure 4.8.

Most of the tested approaches achieved similar F1-measure values (the values range from 72.3% of AODEsr to 63.3% of NN). However, the variance of Recall(critical) and Precision(critical) values is significantly higher. Bayesian approaches performed best among all the tested categories. Specifically, the Naive Bayes classifier achieved the highest recall (87%), but a fairly precision (59%). Conversely, the AODEsr classifier achieves very good results in terms of both recall and precision of class critical.

4.2.5.4 Comparison with regression-based approaches

Some previous works (e.g. [112]) focused on predicting the exact number of bicycles per station at a given time instant. This specific problem cannot be tackled by classification algorithms, but rather using regression techniques. On the other hand, the problem addressed by this paper, i.e., predicting the station occupancy level (critical or non-critical), can be tackled by post-processing the results of regression-based approaches. Specifically, regression algorithms can predict the number of occupied slots at the time of interest. On top of the regression process, the predicted value can be categorized as critical or non-critical based on the occupancy threshold (thr).

To highlight the advantages of using classification instead of regression algorithms to tackle the problem under analysis, we compared the performance of three well-know regression algorithms (Linear regression, Neural network regression, and SVMs-based regression) with that of the AODEsr classifier on the April-May dataset. The achieved results demonstrated that the regression algorithms performed significantly worse than AODEsr in terms of Recall(critical) (regression-based approaches range from 43% to 56%, while AODEsr 71%). Hence, the use of classification algorithms appears to be more appropriate than regression-based ones to address the problem under analysis.

Chapter 5

Conclusions

In this thesis work, we focus on the design and development of innovative solutions to support data mining activities over User-Generated Data (UGD) characterised by different critical issues, via the integration of different data mining techniques in a unified framework. Real datasets coming from three example domains characterized by the above critical issues are considered as reference cases, i.e., health care, social network, and urban environment domains.

Chapter 2 addressed the design and development of data mining algorithms for the analysis of sparse data collections with large cardinality. We presented a unified framework, i.e., Multiplel-Level Data Analysis(MLDA), by jointly exploiting multiple-level clustering, association, and classification analysis. The framework achieves excellent results when considering sparse, high-dimensional real UGD from two reference domains, i.e., patient treatments in health-care domain and content from Twitter in social network domain. Experimental results showed that (i) the proposed multiple-level clustering strategy can perform a valuable preprocessing step for partitioning the data collection into cohesive groups, that are then locally analyzed, (ii) the discovered clusters can be concisely characterized through association analysis capturing correlations among data features, (iii) the mobile application allows a real-time classification of new data objects into one of the above groups based on the classification model, collecting domain-expert feedbacks on the proposed classifications, as well as updating the data collections with new UGD through the mobile device.

Some limitations of the MLDA framework as proposed in this study can represent interesting future research directions to enhance MLDA. (i) In the multiple-level clustering strategy, the number of iteration levels should be currently experimentally tuned by trading-off the computed and the expected quality of the cluster set. Preliminary

studies on the data distribution to identify the diverse degree of data sparseness can support in selecting this parameter. (ii) Nevertheless the multiple-level clsutering is able to discover cohesive clusters, high data sparseness can enforce, at some iteration levels, clusters with limited size. To cope with this issue, as future work, *data taxonomies* can be locally exploited for reducing data sparseness by climbing the abstraction level used for data representation. (iii) Moreover, alternative patterns as for example *weighted sequential patterns* [135] can be adopted to characterize the cluster content. (iv) From the technological perspective, to deal with huge data collections, a future activity can address the deployment of the proposed framework in a *cloud-based platform*, as Apache Mahout or Spark. Some of the above research issues (e.g., issues *(ii)* and *(v)*) have been addressed in the works presented in Chapter 3.

Chapter 3 presents data mining techniques designed and developed in this thesis work to analyze heterogeneous data collections with large cardinality. Heterogeneous data is a common characteristic of datasets in various domains by modeling data in different facets. Innovative data analytics solutions able to acquire, integrate and analyze data containing very large amount of heterogeneous dimensions are needed. For example, when analyzing medical examinations in health care domain, despite patient treatments, patient profile information such as age and gender can be also taken into account. In social network domain, to gain interesting knowledge from tweets, the rather heterogeneous dimensions characterizing Twitter data, such as spatial-temporal information (describing where and when tweets are posted) and the tweet textual content, should be considered in data analysis process. Considering real datasets from these two domains as reference cases, we proposed novel combined distance measures taking into account all considered facets, and integrated the distance measures into the multiple-level clustering analysis in the MLDA framework. The distance measures have been properly tailored to the problem under analysis in each specific application domain. Experimental results demonstrate the ability of the proposed approach in efficiently discovering well-separated cohesive groups through clustering analysis, and in characterizing data collections in terms of different facets through classification or association analysis. In current approach, during the cluster analysis, the number of clusters as well as the parameters weighting the distance of different facets should be experimentally tuned by trading-off the computed and the expected quality of the cluster set. In future work, the selection of these parameters can be supported via preliminary *studies on the data distribution* to identify the diverse degree of data sparseness.

Afterwards, to deal with the limitation of the MLDA framework when aiming at finding correlations among heterogeneous data which may significantly increase the data dimensionality and sparseness, data taxonomy integrated with association analysis (i.e.

generalized association rules) has been presented, for reducing data sparseness by climbing the abstraction level. The approach has been validated over real air pollution data in urban environment domain, where the relationship between pollutant concentrations with traffic conditions representing people's mobility has been monitored. Experimental results demonstrate the potential of the proposed methodology in modeling interesting correlations at different abstraction levels. There is still room for improvements for analysing air pollution data. For example, the system may be enriched with (i) *other kinds of interesting data* affecting air quality such as people's mobility and private/public transport data, and (ii) data mining algorithms to discover *correlations* among weighted air pollution-related data.

In Chapter 4, to cope with the historical UGD, which is often represented as time series for prediction purpose, we presented a windowing approach integrated with various classification and/or regression analysis to perform the prediction. Real datasets coming from two application domains have been considered. In health care domain, patient's cardiopulmonary response has been analysed through the analysis of physiological signals monitored during the cardiopulmonary test. Experimental results, obtained on a real dataset, showed the proposed approach is able to predict both different signal values at the next and final steps of the test with a limited and acceptable error. As future developments of this work, the following issues can be addressed. (i) *Exploitation of different data mining algorithms* to perform the prediction, (ii) *development of visualization tools* to graphically show the prediction with the corresponding error during the test.

In urban environment domain, the upcoming presence of critical conditions in the occupancy levels of bike sharing stations has been predicted. The achieved results showed the effectiveness of the approach and its usefulness to support maintenance actions based on short term predictions and readable models. Future developments of this work will address (i) the integration of the proposed approach with *optimization-based strategies* (e.g. [125]) aimed to optimize the work schedule of people in charge of system maintenance, (ii) the enrichment of station occupancy data with spatial knowledge.

Bibliography

[1] Salton G., *The SMART retrieval system: experiments in automatic document processing.* Prentice-Hall, 1971.

[2] P.-N. Tan, M. Steinbach, V. Kumar *et al.*, *Introduction to data mining.* Pearson Addison Wesley Boston, 2006, vol. 1.

[3] T. Cerquitelli, S. Chiusano, and X. Xiao, "Exploiting clustering algorithms in a multiple-level fashion: A comparative study in the medical care scenario," *Expert Systems with Applications*, vol. 55, pp. 297–312, 2016.

[4] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *KDD Workshop on Text Mining*, 2000.

[5] R. Kashef and M. S. Kamel, "Efficient bisecting k-medoids and its application in gene expression analysis," in *Image Analysis and Recognition.* Springer, 2008, pp. 423–434.

[6] D. Antonelli, E. Baralis, G. Bruno, T. Cerquitelli, S. Chiusano, and N. Mahoto, "Analysis of diabetic patients through their examination history," *Expert Systems with Applications*, vol. 40, no. 11, pp. 4672–4678, 2013.

[7] M. J. Zaki, "Spade: An efficient algorithm for mining frequent sequences," *Mach. Learn.*, vol. 42, no. 1-2, pp. 31–60, jan 2001.

[8] A. Sengur and I. Turkoglu, "A hybrid method based on artificial immune system and fuzzy k-nn algorithm for diagnosis of heart valve diseases," *Expert Systems with Applications*, vol. 35, no. 3, pp. 1011–1020, 2008.

[9] B. Zheng, S. W. Yoon, and S. S. Lam, "Breast cancer diagnosis based on feature extraction using a hybrid of k-means and support vector machine algorithms," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1476–1482, 2014.

[10] H. W. Khaing, "Data mining based fragmentation and prediction of medical data," in *Int. Conf. Computer Research and Development (ICCRD)*, March 2011, pp. 480–485.

[11] N. Esfandiari, M. R. Babavalian, A.-M. E. Moghadam, and V. K. Tabar, "Knowledge discovery in medicine: Current issue and future trend," *Expert Systems with Applications*, vol. 35, no. 41, pp. 4434–4463, 2014.

[12] M. Phanich, P. Pholkul, and S. Phimoltares, "Food recommendation system using clustering analysis for diabetic patients," in *IEEE International Conference on Information Science and Applications (ICISA)*, 2010, pp. 1–8.

[13] Z. Sawacha, G. Guarneri, A. Avogaro, and C. Cobelli, "A new classification of diabetic gait pattern based on cluster analysis of biomechanical data," *Journal of Diabetes Science and Technology*, vol. 4, pp. 1127–38, 2010.

[14] K. Chaturvedi, "Geographic concentrations of diabetes prevalence clusters in texas and their relationship to age and obesity," *http://www.ucgis.org/summer03/studentpapers/kshitijchaturvedi.pdf. Retrieved*, vol. 9, no. 7, p. 2010, 2003.

[15] A. Purwar and S. K. Singh, "Hybrid prediction model with missing value imputation for medical data," *Expert Systems with Applications*, vol. 42, no. 13, pp. 5621–5631, 2015.

[16] O. Karan, C. Bayraktar, H. Gümüşkaya, and B. Karlık, "Diagnosing diabetes using neural networks on small mobile devices," *Expert Systems with Applications*, vol. 39, no. 1, pp. 54 – 60, 2012.

[17] M. E. Menshawy, A. Benharref, and M. Serhani, "An automatic mobile-health based approach for {EEG} epileptic seizures detection," *Expert Systems with Applications*, vol. 42, no. 20, pp. 7157 – 7174, 2015.

[18] K. Polat, S. Güneş, and A. Arslan, "A cascade learning system for classification of diabetes disease: Generalized discriminant analysis and least square support vector machine," *Expert Systems with Applications*, vol. 34, no. 1, pp. 482 – 487, 2008.

[19] B.-H. Juang and L. Rabiner, "The segmental k-means algorithm for estimating parameters of hidden markov models," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 38, no. 9, pp. 1639–1641, Sep 1990.

[20] Kaufman, L. and Rousseeuw, P. J., *Finding groups in data: An introduction to cluster analysis.* Wiley, 1990.

[21] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Knowledge Discovery and Data Mining (KDD)*, 1996, pp. 226–231.

[22] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Computational and Applied Mathematics*, pp. 53–65, 1987.

[23] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.

[24] M. A. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," *Remote sensing of environment*, 1997.

[25] I. ICD-9-CM, "International Classification of Diseases, 9th revision, Clinical Modification. Available: http://icd9cm.chrisendres.com. Last access on March 2011," 2011.

[26] R. M. Rapid Miner Project, "The Rapid Miner Project for Machine Learning. Available: http://rapid-i.com/ Last access on May 2016," 2013.

[27] P. Fournier-Viger, C.-W. Wu, A. Gomariz, and V. S. Tseng, "Vmsp: Efficient vertical mining of maximal sequential patterns," in *Advances in Artificial Intelligence*. Springer, 2014, pp. 83–94.

[28] P. Fournier-Viger, A. Gomariz, T. Gueniche, A. Soltani, C.-W. Wu, and V. S. Tseng, "Spmf: a java open-source pattern mining library," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3389–3393, 2014.

[29] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *ACM SIGMOD Record*, vol. 29, no. 2. ACM, 2000, pp. 1–12.

[30] E. Baralis, T. Cerquitelli, S. Chiusano, L. Grimaudo, and X. Xiao, "Analysis of twitter data using a multiple-level clustering strategy," in *Model and Data Engineering*. Springer, 2013, pp. 13–24.

[31] Z. Yin, R. Li, Q. Mei, and J. Han, "Exploring social tagging graph for web object classification," in *15th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining*, 2009, pp. 957–966.

[32] X. Li, L. Guo, and Y. E. Zhao, "Tag-based social interest discovery," in *17th Int. Conf. on World Wide Web*, 2008, pp. 675–684.

[33] M. Bender, T. Crecelius, M. Kacimi, S. Michel, T. Neumann, J. Parreira, R. Schenkel, and G. Weikum, "Exploiting social relations for query expansion and result ranking," in *IEEE 24th Int. Conf. on Data Engineering Workshop*, 2008, pp. 501 –506.

[34] P. Heymann, D. Ramage, and H. Garcia-Molina, "Social tag prediction," in *31st Int. ACM SIGIR Conf. on Research and development in information retrieval*, 2008, pp. 531–538.

[35] F. Alvanaki, S. Michel, K. Ramamritham, and G. Weikum, "See what's enblogue - real-time emergent topic identification in social media." in *15th Int. Conf. on Extending Database Technology*, 2012, pp. 336–347.

[36] M. Mathioudakis and N. Koudas, "Twittermonitor: trend detection over the twitter stream," in *ACM Int. Conf. on Management of data*, 2010, pp. 1155–1158.

[37] L. Cagliero and A. Fiori, "Generalized association rule mining from Twitter," *Intelligent Data Analysis*, vol. 17, no. 4, 2013.

[38] M. Cheong and V. Lee, "Integrating web-based intelligence retrieval and decision-making from the twitter trends knowledge base," in *2nd ACM Workshop on Social web search and mining*, 2009, pp. 1–8.

[39] A. A. Lopes, R. Pinho, F. V. Paulovich, and R. Minghim, "Visual text mining using association rules," *Comput. Graph.*, vol. 31, no. 3, pp. 316–326, Jun. 2007.

[40] K. L. Qing Chen, Shipper T., "Tweets mining using wikipedia and impurity cluster measurement," in *Int. Conf. Intelligence and Security Informatics*, 2010, pp. 141–143.

[41] J. K. Y.-H. P. Sungchul Kim, Sungho Jeon, "Finding core topics: Topic extraction with clustering on tweet," in *IEEE Int. Conf. on Cloud and Green Computing*, 2012, pp. 777–782.

[42] K. Subramani, A. Velkov, I. Ntoutsi, and P. Kroger, "Density-based community detection in social networks," in *IEEE Int. Conf. on Internet Multimedia Systems Architecture and Application*, 2011, pp. 1–8.

[43] DBDMG, "Available at http://dbdmg.polito.it/wordpress/research/analysis-of-twitter-data-using-a-multiple-level-clustering-strategy/," 2013.

[44] D. I. Ben-Tovim, J. E. Bassham, D. Bolch, M. A. Martin, M. Dougherty, and M. Szwarcbord, "Lean thinking across a hospital: redesigning care at the flinders medical centre," *Australian Health Review*, vol. 31, no. 1, pp. 10–15, 2007.

[45] J. H. Waldhausen, J. R. Avansino, A. Libby, and R. S. Sawin, "Application of lean methods improves surgical clinic experience," *Journal of pediatric surgery*, vol. 45, no. 7, pp. 1420–1425, 2010.

[46] C. McDermott and F. Venditti, "Implementing lean in knowledge work: Implications from a study of the hospital discharge planning process," *Oper Manag Res*, pp. 1–13, 2015.

[47] R. Cima, M. Brown, J. Hebl, R. Moore, J. Rogers, A. Kollengode, G. Amstutz, C. Weisbrod, B. Narr, and C. Deschamps, "Use of lean and six sigma methodology to improve operating room efficiency in a high-volume tertiary-care academic medical center," *J Am Coll Surg.*, vol. 213, no. 1, pp. 83–92, 2011.

[48] L. Rutman, K. Stone, J. Reid, G. A. T. Woodward, and R. Migita, "Improving patient flow using lean methodology: an emergency medicine experience," *Current Treatment Options in Pediatrics*, vol. 1, no. 4, pp. 359–371, 2015.

[49] M. J. Vermeulen, T. A. Stukel, A. Guttmann, B. H. Rowe, M. Zwarenstein, B. Golden, A. Nigam, G. Anderson, R. S. Bell, M. J. Schull, M. Afilalo, G. Anderson, R. S. Bell, D. Carew, M. Carter, M. Cooke, B. Golden, A. Guttmann, A. Nigam, B. Rowe, T. Rutledge, M. Schull, T. Stukel, M. Vermeulen, and M. Zwarenstein, "Evaluation of an emergency department lean process improvement program to reduce length of stay," *Annals of Emergency Medicine*, vol. 64, no. 5, pp. 427 – 438, 2014.

[50] T. Exarchos, C. Papaloukas, D. Fotiadis, and L. Michalis, "An association rule mining-based methodology for automated detection of ischemic ecg beats," in *IEEE Transactions on Biomedical Engineering*, 2006, pp. 1531–1540.

[51] R. Malpani, M. Lu, D. Zhang, and W. Sung, "Mining transcriptional association rules from breast cancer profile data," in *IEEE Int. Conf. on Information Reuse and Integration (IRI)*, 2011, pp. 154–159.

[52] N. Jay, F. Kohler, and A. Napoli, "Using formal concept analysis for mining and interpreting patient flows within a healthcare network," in *4th Int. Conf. on Concept Lattices and Their Applications*. Springer Berlin Heidelberg, 2008, pp. 263–268.

[53] T. Chen, L. Chou, and S. Hwang, "Application of a data-mining technique to analyze coprescription patterns for antacids in taiwan." Clin Ther, 2003, pp. 2453–2463.

[54] Y. Kung, Y. Chen, S. Hwang, T. Chen, and F. Chen, "The prescriptions frequencies and patterns of chinese herbal medicine for allergic rhinitis in taiwan." Allergy 61, 2006, pp. 1316–1318.

[55] F. Chen, Y. Kung, Y. Chen, M. Jong, T. Chen, and et al., "Frequency and pattern of chinese herbal medicine prescriptions for chronic hepatitis in taiwan." J Ethnopharmacol 117, 2008, pp. 84–91.

[56] Y. Chun, T. Chen, and L. Chou, "Application of frequent itemsets mining to analyze patterns of one-stop visits in taiwan." PLoS One, 2011.

[57] H. Elghazel, V. Deslandres, K. Kallel, and A. Dussauchoy, "Clinical pathway analysis using graph-based approach and markov models," in *2nd Int. Conf. on Digital Information Management, ICDIM '07.*, 2007, pp. 279–284.

[58] S. Khanna, J. Boyle, N. Good, and J. Lind, "Time based clustering for analyzing acute hospital patient flow," in *Int. IEEE Conf. on Engineering in Medicine and Biology Society (EMBC)*, 2012, pp. 5903–5906.

[59] D. Sundaramoorthi, V. C. Chen, S. B. Kim, J. M. Rosenberger, and D. F. Buckley-Behan, "A data-integrated nurse activity simulation model," in *Proceedings of the 38th conference on Winter simulation.* Winter Simulation Conference, 2006, pp. 960–966.

[60] E. Cecchetti, "Piano sanitario regionale 2002-2004: approvazione del programma pluriennale di interventi sanitari strategici." *Consiglio regionale - deliberazioni n 000202 del 23/12/2002 (boll. n 4 del 22/01/2003, parte seconda, sezione i).*, 2002.

[61] R. Srikant and R. Agrawal, *Mining sequential patterns: Generalizations and performance improvements.* Springer, 1996.

[62] J. W. Vaupel and A. I. Yashin, "Heterogeneity's ruses: some surprising effects of selection on population dynamics," *The American Statistician*, vol. 39, no. 3, pp. 176–185, 1985.

[63] G. Bruno, T. Cerquitelli, S. Chiusano, and X. Xiao, "A clustering-based approach to analyse examinations for diabetic patients," in *2014 IEEE International Conference on Healthcare Informatics, ICHI 2014, Verona, Italy, September 15-17, 2014*, 2014, pp. 45–50.

[64] E. Georga, V. Protopappas, A. Guillen, G. Fico, D. Ardigo, M. T. Arredondo, T. P. Exarchos, D. Polyzos, and D. I. Fotiadis, "Data mining for blood glucose prediction and knowledge discovery in diabetic patients: The metabo diabetes modeling and management system," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE.* IEEE, 2009, pp. 5633–5636.

[65] V. S. Tseng, C.-H. Lee, and J. C.-Y. Chen, "An integrated data mining system for patient monitoring with applications on asthma care," in *Computer-Based Medical Systems, 2008. CBMS'08. 21st IEEE International Symposium on.* IEEE, 2008, pp. 290–292.

[66] K. Srinivas, B. K. Rani, and A. Govrdhan, "Applications of data mining techniques in healthcare and prediction of heart attacks," *International Journal on Computer Science and Engineering (IJCSE)*, vol. 2, no. 02, pp. 250–255, 2010.

[67] B. Honigman, J. Lee, J. Rothschild, P. Light, R. M. Pulling, T. Yu, and D. W. Bates, "Using computerized data to identify adverse drug events in outpatients," *Journal of the American Medical Informatics Association*, vol. 8, no. 3, pp. 254–266, 2001.

[68] P. S. Python Software Foundation, 2014.

[69] R. Lee, S. Wakamiya, and K. Sumiya, "Discovery of unusual regional social activities using geo-tagged microblogs," *World Wide Web*, vol. 14, no. 4, pp. 321–349, 2011.

[70] T. Kim, G. Huerta-Canepa, J. Park, S. J. Hyun, and D. Lee, "What's happening: Finding spontaneous user clusters nearby using twitter." in *Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 806–809.

[71] S. Wakamiya, R. Lee, and K. Sumiya, "Measuring crowd-sourced cognitive distance between urban clusters with twitter for socio-cognitive map generation," in *Proceedings of the Fourth International Conference on Emerging Databases-Technologies, Applications, and Theory, EDB*, vol. 12, 2012, pp. 181–192.

[72] B. De Longueville, R. S. Smith, and G. Luraschi, "\"omg, from here, i can see the flames!\": a use case of mining location based social networks to acquire spatio-temporal data on forest fires." in *Proceedings of the 2009 international workshop on location based social networks*. ACM, 2009, pp. 73–80.

[73] C. H. Lee, "Mining spatio-temporal information on microblogging streams using a density-based online clustering method," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9623–9641, 2012.

[74] T. Sakaki, M. Okazaki, and Y. Matsuo, "Earthquake shakes twitter users: real-time event detection by social sensors." in *Proceedings of the 19th international conference on World wide web*. ACM, 2010, pp. 851–860.

[75] M. Nagarajan, K. Gomadam, A. P. Sheth, A. Ranabahu, R. Mutharaju, and A. Jadhav, *Spatio-temporal-thematic analysis of citizen sensor data: Challenges and experiences*. Springer, 2009.

[76] L. Cagliero, T. Cerquitelli, P. Garza, and L. Grimaudo, "Twitter data analysis by means of strong flipping generalized itemsets," *Journal of Systems and Software*, vol. 94, pp. 16–29, 2014.

[77] K. Saito, K. Ohara, M. Kimura, and H. Motoda, "Change point detection for burst analysis from an observed information diffusion sequence of tweets," *Journal of Intelligent Information Systems*, vol. 44, no. 2, pp. 243–269, 2015.

[78] E. Lloret, A. Balahur, J. M. Gómez, A. Montoyo, and M. Palomar, "Towards a unified framework for opinion retrieval, mining and summarization," *Journal of Intelligent Information Systems*, vol. 39, no. 3, pp. 711–747, 2012.

[79] J. Makkonen, R. Kerminen, I. D. Curcio, S. Mate, and A. Visa, "Detecting events by clustering videos from large media databases," in *Proceedings of the 2nd ACM international workshop on Events in multimedia.* ACM, 2010, pp. 9–14.

[80] C.-H. Lee, H.-C. Yang, T.-F. Chien, and W.-S. Wen, "A novel approach for event detection by mining spatio-temporal information on microblogs," in *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on.* IEEE, 2011, pp. 254–259.

[81] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *SIGMOD Conference*, 1993, pp. 207–216.

[82] M. Lodovici, M. Venturini, E. Marini, D. Grechi, and P. Dolara, "Polycyclic aromatic hydrocarbons air levels in florence, italy, and their correlation with other air pollutants," *Chemosphere*, vol. 50, no. 3, pp. 377–382, January 2003.

[83] M. Statheropoulos, N. Vassiliadis, and A. Pappa, "Principal component and canonical correlation analysis for examining air pollution and meteorological data," *Atmospheric Environment*, vol. 32, no. 6, pp. 1087 – 1095, 1998.

[84] H. K. Elminir, "Dependence of urban air pollutants on meteorology," *Science of The Total Environment*, vol. 350, no. 1-3, pp. 225 – 237, 2005.

[85] Y. Zheng, F. Liu, and H. Hsieh, "U-air: when urban air quality inference meets big data," in *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 1436–1444.

[86] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. 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*, 2015, pp. 2267–2276.

[87] E. Baralis, L. Cagliero, T. Cerquitelli, V. D'Elia, and P. Garza, "Support driven opportunistic aggregation for generalized itemset extraction," in *5th IEEE International Conference on Intelligent Systems*, 2010, pp. 102–107.

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

[89] R. Agrawal and R. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases," *Proceedings of the 20th IEEE Int. Conf. on Very Large Data Bases*, pp. 487–499, 1994.

[90] R. Srikant and R. Agrawal, "Mining generalized association rules," in *VLDB 1995*, 1995, pp. 407–419.

[91] Wikipedia Meteo information about metereological data, *Available at https://en.wikipedia.org/wiki/Rain, https://en.wikipedia.org/wiki/Wind, https://en.wikipedia.org/wiki/Ultravioletindex, https://en.wikipedia.org/wiki/Atmosphericpressur Last access: February 2016.*

[92] E. Baralis, T. Cerquitelli, S. Chiusano, A. Giordano, A. Mezzani, D. Susta, and X. Xiao, "Predicting cardiopulmonary response to incremental exercise test," in *Computer-Based Medical Systems (CBMS), 2015 IEEE 28th International Symposium on.* IEEE, 2015, pp. 135–140.

[93] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 38:1–38:55, Sep. 2014.

[94] D. M. Lambrick, J. A. Faulkner, A. V. Rowlands, and R. G. Eston, "Prediction of maximal oxygen uptake from submaximal ratings of perceived exertion and heart rate during a continuous exercise test: the efficacy of rpe 13," *European journal of applied physiology*, pp. 1–9, 2009.

[95] T. S. Bowen, D. T. Cannon, G. Begg, V. Baliga, K. K. Witte, and H. B. Rossiter, "A novel cardiopulmonary exercise test protocol and criterion to determine maximal oxygen uptake in chronic heart failure," *Journal of Applied Physiology*, pp. 451–458, 2012.

[96] F. Sartor, G. Vernillo, H. M. de Morree, A. G. Bonomi, A. L. Torre, H.-P. Kubis, and A. Veicsteinas, "Estimation of maximal oxygen uptake via submaximal exercise testing in sports, clinical, and home settings," *Sports Medicine*, vol. 43, no. 9, pp. 865–873, 2013.

[97] M. F. Akay, E. I. M. Zayid, E. Aktürk, and J. D. George, "Artificial neural network-based model for predicting vo2max from a submaximal exercise test," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 2007–2010, 2011.

[98] N. Chbat, M. Giannessi, A. Albanese, and M. Ursino, "A comprehensive cardiopulmonary simulation model for the analysis of hypercapnic respiratory failure," in *Engineering in Medicine and Biology Society, EMBC 2009*, 2009, pp. 5474–5477.

[99] S. Urooj and M. Khan, "A computer based prediction for diagnosis of pulmonary edema," in *Medical Measurements and Applications Proceedings (MeMeA), 2010*, 2010, pp. 28–31.

[100] J. Batzel, L. Ellwein, and M. Olufsen, "Modeling cardio-respiratory system response to inhaled co2 in patients with congestive heart failure," in *Engineering in Medicine and Biology Society, EMBC, 2011*, 2011, pp. 2418–2421.

[101] M. Pollock, G. Gaesser, and J. Butcher, "The recommended quantity and quality of exercise for developing and maintaining cardiorespiratory and muscular fitness, and flexibility in health adults." *Medicine & Science in Sports & Exercise*, vol. 30, pp. 975–991, 1998.

[102] R. Bunescu, N. Struble, C. Marling, J. Shubrook, and F. Schwartz, "Blood glucose level prediction using physiological models and support vector regression," in *Machine Learning and Applications (ICMLA), 2013*, vol. 1, 2013, pp. 135–140.

[103] C.-Y. Chang, J.-Y. Zheng, and C.-J. Wang, "Based on support vector regression for emotion recognition using physiological signals," in *Neural Networks (IJCNN)*, 2010, pp. 1–7.

[104] J. Bock and D. Gough, "Toward prediction of physiological state signals in sleep apnea," *Biomedical Engineering, IEEE Transactions on*, pp. 1332–1341, 1998.

[105] E. Baralis, T. Cerquitelli, S. Chiusano, V. D'Elia, R. Molinari, and D. Susta, "Predicting the highest workload in cardiopulmonary test," in *IEEE CBMS*, 2010, pp. 32–37.

[106] ——, "Early prediction of the highest workload in incremental cardiopulmonary tests," *ACM TIST*, vol. 4, no. 4, p. 70, 2013.

[107] S. Kasetty, C. Stafford, G. P. Walker, X. Wang, and E. Keogh, "Real-time classification of streaming sensor data," in *Proceedings of the 20th IEEE ICTA*, ser. Volume 01. IEEE Computer Society, 2008, pp. 149–156.

[108] W. D. McArdle, F. Katch, and V. L. Katch, *Essentials of exercise physiology.* Lippincott Williams & Wilkins, 2006.

[109] Maugeri, "Fondazione Salvatore Maugeri - Clinica del Lavoro e della Riabilitazione I.R.C.C.S. Available at http://www.fsm.it/."

[110] S. Ferrara, "Master thesis. Analysis of the cardiopulmonary response in incremental exercise tests using data mining techniques." 2015, [Online]. Available: http://dbdmg.polito.it/wordpress/research/physiological-data-analysis/.

[111] J. Froehlich, J. Neumann, and N. 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)*, Nov. 2008.

[112] A. Kaltenbrunner, R. Meza, J. Grivolla, J. Codina, and R. Banchs, "Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system," *Pervasive Mob. Comput.*, vol. 6, no. 4, pp. 455–466, Aug. 2010.

[113] M. Rainer-Harbach, P. Papazek, G. R. Raidl, B. Hu, and C. Kloimüllner, "Pilot, grasp, and VNS approaches for the static balancing of bicycle sharing systems," *J. Global Optimization*, vol. 63, no. 3, pp. 597–629, 2015.

[114] P. Vogel, T. Greiser, and D. C. Mattfeld, "Understanding bike-sharing systems using data mining: Exploring activity patterns," *Procedia - Social and Behavioral Sciences*, vol. 20, pp. 514 – 523, 2011.

[115] I.-L. Wang and C.-W. Wang, "Analyzing bike repositioning strategies based on simulations for public bike sharing systems: Simulating bike repositioning strategies for bike sharing systems," in *Advanced Applied Informatics (IIAIAAI), 2013 IIAI International Conference on*, Aug 2013, pp. 306–311.

[116] J. W. Yoon, F. Pinelli, and F. Calabrese, "Cityride: A predictive bike sharing journey advisor," in *13th IEEE International Conference on Mobile Data Management, MDM 2012, Bengaluru, India, July 23-26, 2012*, 2012, pp. 306–311.

[117] A. Sarkar, N. Lathia, and C. Mascolo, "Comparing cities' cycling patterns using online shared bicycle maps," *Transportation*, vol. 42, no. 4, pp. 541–559, 2015.

[118] S. Hasan, X. Zhan, and S. 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 International Workshop on Urban Computing*, 2013, pp. 6:1–6:8.

[119] F. Girardin, F. Calabrese, F. D. Fiore, C. Ratti, and J. Blat, "Digital footprinting: Uncovering tourists with user-generated content," *IEEE Pervasive Computing*, vol. 7, no. 4, pp. 36–43, Oct. 2008.

[120] C. Etienne and O. Latifa, "Model-based count series clustering for bike sharing system usage mining: A case study with the Velib' system of paris," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, pp. 39:1–39:21, Jul. 2014.

[121] V. Ciancia, D. Latella, M. Massink, and R. Pakauskas, "Exploring spatio-temporal properties of bike-sharing systems," in *Self-Adaptive and Self-Organizing Systems Workshops (SASOW), 2015 IEEE International Conference on*, Sept 2015, pp. 74–79.

[122] Y. Zhang and Z. Huang, "Performance evaluation of bike sharing system in wuchang area of wuhan, china," in *China Planning Conference (IACP), 2012 6th International Association for*, June 2012, pp. 1–10.

[123] W. Zhang, Y. Zhang, and T. Kim, "Detecting bad information in mobile wireless networks based on the wireless application protocol," *Computing*, vol. 96, no. 9, pp. 855–874, 2014.

[124] Q. Kong and T. Maekawa, "Reusing training data with generative/discriminative hybrid model for practical acceleration-based activity recognition," *Computing*, vol. 96, no. 9, pp. 875–895, 2014.

[125] S. Srinivasan and S. Ramakrishnan, "A social intelligent system for multi-objective optimization of classification rules using cultural algorithms," *Computing*, vol. 95, no. 4, pp. 327–350, 2013.

[126] L. Gruenwald, H. Yang, M. S. Sadik, and R. Shukla, "Using data mining to handle missing data in multi-hop sensor network applications," ser. MobiDE '10, 2010, pp. 9–16.

[127] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *ECML*, 1998, pp. 4–15.

[128] E. Baralis, S. Chiusano, and P. Garza, "A lazy approach to associative classification," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 20, no. 2, pp. 156–171, 2008.

[129] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[130] J. Quinlan, *C4.5: Programs for Machine Learning*. The Morgan Kaufmann, 1993.

[131] F. Zheng, G. I. Webb, P. Suraweera, and L. Zhu, "Subsumption resolution: an efficient and effective technique for semi-naive bayesian learning," *Machine Learning*, vol. 87, no. 1, pp. 93–125, 2012.

[132] F. Zheng and G. I. Webb, "Efficient lazy elimination for averaged one-dependence estimators," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: ACM, 2006, pp. 1113–1120.

[133] W. W. Cohen, "Fast effective rule induction," in *Proceedings of the Twelfth International Conference on Machine Learning.* Morgan Kaufmann, 1995, pp. 115–123.

[134] M. I. Jordan and C. M. Bishop, "Neural networks," *ACM Comput. Surv.*, vol. 28, no. 1, pp. 73–75, Mar. 1996.

[135] U. Yun, "A new framework for detecting weighted sequential patterns in large sequence databases," *Know.-Based Syst.*, vol. 21, no. 2, pp. 110–122, Mar. 2008.