Misleading generalized itemset mining in the cloud

Elena Baralis, Luca Cagliero, Tania Cerquitelli, Silvia Chiusano, Paolo Garza, Luigi Grimaudo, Fabio Pulvirenti
Dipartimento di Automatica e Informatica
Politecnico di Torino
Torino, Italy
Email: {name.surname}@polito.it

***Abstract*—In the era of smart cities huge data volumes are continuously generated and collected, thus prompting the need for efficient and distributed data mining approaches. Generalized itemset mining is an established data mining technique, which entails the discovery of multiple-level patterns hidden in the analyzed data by exploiting analyst-provided taxonomies. Among the generalized itemsets, the most peculiar high-level patterns are those with many contrasting correlations among items at different abstraction levels. They represent misleading situations that are worth analyzing separately by experts during manual inspection.**

This paper proposes a novel cloud-based service, named MGI-CLOUD**, to efficiently mine misleading multiple-level patterns, i.e., the Misleading Generalized Itemsets, on a distributed computing environment.** MGI-CLOUD **consists of a set of distributed MapReduce jobs running in the cloud. As a case study, the system has been contextualized in a real-life scenario, i.e., the analysis of traffic law infractions committed in a smart city environment. The experiments, performed on real datasets, demonstrate the efficiency and effectiveness of** MGI-CLOUD**.**

***Keywords*-Generalized itemset mining; distributed computing model; cloud-based service; smart city.**

I. INTRODUCTION

In the last few years the new Information and Communication Technologies (ICT) have supported cities in becoming smart. Hence, the capability of smart cities to generate and collect data of public interest (e.g., information about social events, public service usage, traffic law infractions) has increased at an unprecedented rate, to such an extent that data rapidly scales towards "Big Data".

Big Data analyses are challenging tasks because the computational cost of data mining processes is often very high and, in some cases, prohibitive on a non-distributed system. Hence, relevant research efforts have been devoted to large-scale data mining based on the MapReduce paradigm [1]. Data mining encompasses a large variety of techniques, such as cluster analysis, frequent pattern extraction, and classification [2]. For example, some remarkable attempts to address itemset and association rule mining from Big Data have recently been made (e.g., [3], [4], [5], [6]). This paper focuses on an established pattern mining technique called generalized itemset extraction [7]. This technique has already been applied to data coming from different application domains (e.g., market basket analysis [7], network traffic data analysis [8], genetic data mining [9]). Generalized itemset mining entails discovering correlations among data at different abstraction levels. By exploiting a taxonomy (i.e., a set of is-a hierarchies) built over the analyzed data, frequent generalized itemsets, which represent recurrent co-occurrences among data items at different granularity levels, are extracted. These patterns are worth considering by domain experts to transform huge amounts of raw data into useful and actionable knowledge. However, a subset of peculiar high-level patterns should be analyzed separately during manual result inspection. More specifically, each generalized itemset has a correlation type which indicates the strength of the correlation between the corresponding items. Misleading Generalized Itemsets (MGIs) [10] are generalized itemsets whose correlations type is in contrast to those of most of their low-level descendant itemsets. These high-level patterns are worth considering for in-depth analysis because they are likely to represent misleading and thus potentially interesting situations. In [10] MGI extraction is performed in main memory on top of frequent level-sharing itemsets. Unfortunately, when coping with Big Datasets, a large number of itemsets is often generated at step (i) thus MGI extraction becomes a challenging task. However, to the best of our knowledge, no attempt to mine MGIs on a distributed architecture has been made yet.

This paper presents a cloud-based service, named MGI-CLOUD (Misleading Generalized Itemset miner in the CLOUD), to efficiently mine MGIs on a distributed computing model. To efficiently cope with Big Data collections, the system implementation is distributed and most operations are mapped to the MapReduce programming paradigm [1]. As a reference case study, the proposed approach has been applied to a real application context: the analysis of the traffic law infractions committed by the citizens of Turin, an important business and cultural centre in northern Italy. Real infraction data is provided as open data by the Turin administration. The goal of the analysis is to improve the efficiency of public services, the transparency of public administrations, and the

The research leading to these results has received funding from the European Union under the FP7 Grant Agreement n. 285248 (Integrated Project FI-WARE) and the FP7 Grant Agreement n. 619633 (Integrated Project ONTIC)

awareness of the degree of civilization of urban people. The experimental results show the effectiveness and efficiency of the MGI-CLOUD architecture as well as they demonstrate its applicability to the analyzed use-case.

The paper is organized as follows. Section II overviews most relevant previous works. Section III states the problem addressed in this paper. Section IV presents the MGI-CLOUD architecture while a preliminary experimental evaluation of our approach is reported in Section V. Finally, Section VI draws conclusions and discusses future research directions.

II. RELATED WORK

In the last years, a relevant research effort has been devoted to large-scale itemset mining based on the MapReduce paradigm [1]. The goal is to propose itemset extraction algorithms that distribute data and computation across a distributed architecture to scale the mining process towards Big Data [3], [4], [5]. A parallel version of an established itemset mining algorithm, i.e., FP-Growth, has first been proposed in [3]. The algorithm, named PFP, consists of two separate MapReduce jobs and it achieves an almost linear speedup. It converts transactions of the original database into some new databases of group-dependent transactions so that local FP-trees built from different group-dependent transactions can be separately processed during the recursive conditional FP-tree constructing process. For each group PFP extracts top-k frequent itemsets (i.e., a subset of frequent itemsets). More recently, two new methods, namely Dist-Eclat and BigFIM, for mining all frequent itemsets from large datasets have been presented in [4]. Specifically, Dist-Eclat focuses on improving algorithm speed, while BigFIM is optimized to run on huge datasets. In parallel, a cloud-based service for association rule mining from network traffic data has been presented in [5]. Unlike [3], [4], [5], this paper investigates the applicability of a generalized pattern mining technique on the MapReduce platform.

The frequent generalized frequent itemset and association rule mining problems [7] have largely been studied by the data mining community. The firstly proposed approach [7] generates itemsets by considering for each item all its ancestors in the taxonomy. To avoid generating all the possible itemsets, the authors in [11], [12] proposed to push (analyst-provided) constraints into the mining process. In parallel, many algorithm optimizations and variations have been proposed [13], [14], [15]. For example, the approach presented in [13] proposes an optimization strategy based on a top-down hierarchy traversal, while in [14] the authors propose to mine closed and maximal generalized itemsets. More recently, a new type of generalized pattern, called Misleading Generalized Itemset (MGI), has been proposed [10]. MGIs are high-level (generalized) itemsets for which a relevant subset of frequent descendants have a correlation type in contrast to their common ancestor. MGIs are worth considering separately from traditional itemsets if their low-level contrasting correlations cover almost the same portion of data as the high-level itemset, because the information provided by traditional high-level patterns becomes misleading. Unlike [10], this paper investigates how to perform MGI mining on the MapReduce platform. Furthermore, it presents a cloud-based service for discovering MGIs from Big Data acquired in smart city environments.

III. PRELIMINARY CONCEPTS AND PROBLEM STATEMENT

A relational dataset $\mathcal{D}$ consists of a set of records, where each record is a set of items [2]. Each item is a pair $(attribute,\ value)$. A taxonomy Γ built over the source dataset $\mathcal{D}$ aggregates the data items into higher-level concepts (i.e., the generalized items). Table I and Table II report two representative examples of relational dataset and taxonomy, respectively, which hereafter will be used as running examples.

A k-itemset is a set of k (generalized) items. For example, {(*Time*,*a.m.*), (*Infraction name*,*One-way infraction*)} is a 2-itemset, which indicates that the two items co-occur (possibly at different abstraction levels) in the source data. Items/itemsets are characterized by many notable properties [7], such as support, coverage, descent and level of abstraction according to an input taxonomy Γ. For their definitions please refer to [7], [13], [10]. Similar to [13], [16], we target the correlations among items at same abstraction level, i.e. the itemsets that exclusively contain items with the same level. Such patterns are denoted by *level-sharing itemsets* [13].

The itemset correlation measures the strength of the correlation between its items. Similar to [16], in this paper we evaluate the correlation of a k-itemset I by means of the Kulczynsky (Kulc) correlation measure [17] Kulc values range from 0 to 1. By properly setting maximum negative and minimum positive Kulc thresholds, hereafter denoted by *max_neg_cor* and *min_pos_cor*, the itemsets may be classified as negatively correlated, uncorrelated, or positively correlated itemsets according to their correlation value.

Let $\mathcal{LSI}$ be the set of all frequent level-sharing itemsets in $\mathcal{D}$ according to a minimum support threshold min_sup. Given a frequent level-sharing itemset $X \in \mathcal{LGI}$ of level $l \geq 2$, let Desc$^*[X,\Gamma]$ be the subset of corresponding level-$(l-1)$ X's descendants for which the correlation type is in contrast to those of X. A Misleading Generalized Itemset (MGI) is a pattern in the form $X \triangleright \mathcal{E}$, where $X \in \mathcal{LSGI}$ and $\mathcal{E}$=Desc$^*[X,\Gamma]$ [10].

For example, by enforcing min_sup=10%, max_neg_cor=0.70, and min_pos_cor=0.80, MGI {(Time,a.m.), (Infraction name,Prohibition)} $\triangleright$ {(Time, [8 a.m.,9 a.m.]), (Infraction name,Speeding)} is mined from the dataset in Table I, because {(Time,a.m.), (Infraction name,Prohibition)} has a positive correlation (0.83),

Table I
EXAMPLE DATASET $\mathcal{D}$ AFTER DISCRETIZATION.

Rid	Infraction name	time stamp
1	One-way infraction	[8 a.m.,9 a.m.]
2	One-way infraction	[8 a.m.,9 a.m.]
3	Speeding	[8 a.m.,9 a.m.]
4	Driving without license	[9 a.m.,10 a.m.]
5	Driving without license	[9 a.m.,10 a.m.]
6	Unfastened seat belt	[4 p.m.,5 p.m.]
7	One-way infraction	[8 a.m.,9 a.m.]

Table II
EXAMPLE TAXONOMY BUILT OVER ITEMS IN $\mathcal{D}$

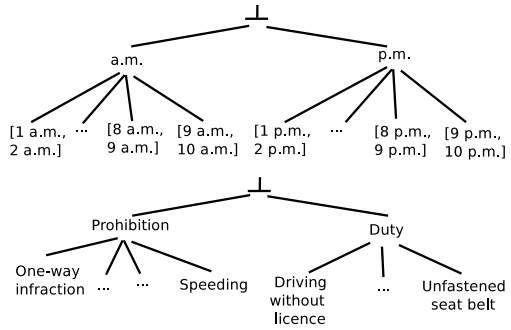

Table III
MISLEADING GENERALIZED ITEMSETS MINED FROM $\mathcal{D}$. MIN_SUP = 10%, MAX_NEG_COR= 0.70, MIN_POS_COR= 0.80, AND MAX_NOD = 80%.

Frequent itemset (level≥2) [correlation type (Kulc value)]	Frequent descendants [correlation type (Kulc value)]	Not overlapping degree (%)
{ (Time, a.m.), (Infraction name, Prohibition)} *[positive (5/6=0.83)]*	{(Time, [8 a.m.,9 a.m.]), (Infraction name, One-way infraction)} *[positive (7/8=0.88)]* {(Time, [8 a.m.,9 a.m.]), (Infraction name, Speeding)} *[[negative (5/8=0.63)]*	75
{(Time, a.m.), (Infraction name, Duty)} *[negative (1/2=0.50)]*	{(Time, [9 a.m., 10 a.m.]), (Infraction name, Driving without license)} *[positive (1)]*	0
{(Time, p.m.), (Infraction name, Duty)} *[negative (2/3=0.66)]*	{(Time, [4 p.m.,5 p.m.]), (Infraction name, Unfastened seat belt)} *[positive (1)]*	0

whereas its descendant itemset {(Time,[8 a.m., 9 a.m.]), (Infraction name,Speeding)} is negatively correlated (0.63).

To measure the degree of interest of a MGI $X \triangleright \mathcal{E}$ with respect to its corresponding traditional itemset version (X), the Not Overlapping Degree (NOD) measure has been defined in [10]. The NOD of an MGI $X \triangleright \mathcal{E}$ is defined as $\frac{\sup(X,\mathcal{D})-\text{cov}(\mathcal{E},\mathcal{D})}{\sup(X,\mathcal{D})}$. It expresses the relative difference between the support of the ancestor itemset X and the coverage of its low-level contrasting correlations in $\mathcal{E}$. The NOD values range from 0 to 1. The lower NOD value we achieve, the more significant the degree of overlapping between the contrasting low-level correlations in $\mathcal{E}$ and their common ancestor X becomes.

The mining task addressed by this paper entails discovering from $\mathcal{D}$ all the MGIs for which the NOD value is less than or equal to a maximum threshold max_NOD. The subset of Misleading Generalized Itemsets mined from Table I by setting the maximum NOD threshold to 80% is reported in Table III.

IV. THE MGI-CLOUD ARCHITECTURE

The MGI-CLOUD architecture provides a cloud-based service for discovering hidden and actionable patterns among potentially Big datasets. We focus our analysis on a specific case study, i.e., the analysis of the traffic law infractions committed in a urban environment. To efficiently cope with Big Data, the system implementation is distributed and most operations are mapped to the MapReduce programming paradigm [1]. The architecture has been designed as a chain of distributed jobs running on an Hadoop cluster, as described below.

A. Data retrieval and preparation

Data about traffic law infractions was collected by traffic engineers. Reports about daily infractions are collected in local data repositories, which are then integrated into open Big Datasets. A traffic law infraction dataset $\mathcal{D}$ consists of a set of records r, each one representing a different infraction. Each record is a set of *items*, where items are pairs (*attribute*,*value*). *Attribute* can be a specific characteristic of the traffic law infraction (e.g., infraction name, law article) or a property of the context in which the infraction was committed (location, date, time), while *value* is the value assumed by the corresponding attribute. Hereafter, for the sake of simplicity, we focus our analysis on the following attribute subset: *Infraction name*, *Vehicle type*, *Location*, *Date*, and *Time*. We discretized time stamps (e.g., 8.10 a.m.) into 1-hour time slots (e.g., from 8 a.m. to 9 a.m.) using ad-hoc mapping functions. The dataset reported in Table I is already the output of the discretization step. Attribute selection and data discretization are performed as distributed MapReduce jobs.

B. Taxonomy generation

To analyze data from a high-level viewpoint, real infraction datasets are equipped with taxonomies. A taxonomy is a

set of is-a hierarchies built over data items in $\mathcal{D}$. An example taxonomy built over the dataset in Table I is depicted in Table II. Items whose value is an high-level aggregation belonging to the taxonomy (e.g., (*Infraction name*,*Duty*)) are called *generalized items*.

Analyst-provided taxonomies could be generated either manually or semi-automatically by domain experts. To perform our analyzes, we built 3-level hierarchies over the contextual attributes (Location, Time, Date). Specifically, geographical addresses are aggregated into the zip code, which in turn are aggregated into the corresponding district; 1-hour time slots are generalized as the corresponding 4- and 12-hour time slots, while dates are generalized as the corresponding month and year. Furthermore, vehicles are generalized as the corresponding category (e.g., *Car*, *Dumper track*, *Pickup track*), and infraction names are aggregated into the corresponding high-level class given by the public administration of Turin. In this work we consider as input balanced taxonomies (i.e., taxonomies whose hierarchies have all the same height). If experts do not provide balanced taxonomies, a re-balancing procedure similar to those adopted in [16] is applied prior to level-sharing itemset mining.

C. Level-sharing itemset mining

Given a preprocessed infraction dataset and a minimum support threshold min_sup, this job accomplishes the first MGI mining step, i.e., the extraction of all frequent level-sharing itemsets [13]. This job performs the following tasks.
Dataset extension. This task entails producing a new dataset version which integrates taxonomy information. To enable frequent level-sharing itemset mining from data containing items at different abstraction levels, it generates multiple copies of each record, one for each taxonomy level. While the original record contains only taxonomy leaves (i.e., the dataset items), each copy contains the corresponding combination of item generalizations at a different abstraction level.To avoid unnecessary I/O operations, the extended dataset version is not materialized on disk, but it is directly generated in the map function of the itemset extraction task and then immediately stored into a compact FP-tree structure [2].
Itemset extraction. To efficiently mine frequent level-sharing itemsets [13] from the extended dataset version, this task exploits a variation of the Hadoop-based itemset mining algorithm proposed in [5].

D. MGI extraction

This job performs MGI mining on top of the frequent level-sharing itemsets. Specifically, it accomplishes the task stated in Section III. This step consists of a MapReduce job, as described in the following. The contribution of this job is new because, to the best of our knowledge, no cloud-based service currently supports MGI mining from Big Data.

To extract MGIs we combine each frequent level-sharing itemset I with its corresponding set of descendant itemsets Desc$[I, \Gamma]$. More specifically, In the map function for each level-sharing itemset I, the following two pairs (key, $value$) are emitted: (i) a pair (key, $value$), where key is the direct ancestor of itemset I and $value$ the itemset I with its main properties (i.e., support and Kulc values) and (ii) a pair (key, $value$), where key is the itemset I is the $value$: itemset I with its main properties (i.e., support and Kulc values). Two pairs are emitted because each itemset can be a descendant of an itemset and a parent of another one at the same time. The first pair allows us to associate I with the ancestor key, whereas the second pair is used to associate I to itself if MGIs in the form $I \triangleright \mathcal{E}$ are extracted. The generated pairs allow us to map each itemset and its corresponding descendants to the same key. Hence, in the reduce function, each key is associated with a specific itemset I and the corresponding set of values contains both the (ancestor) itemset I and its respective descendants. By iterating on the set of values associated with key I, we generate candidate MGIs $I \triangleright \mathcal{E}$, where $\mathcal{E}$ is the set of I's descendants in contrast to I in terms of correlation type, and we compute the corresponding NOD values. Finally, only the MGIs satisfying the max_NOD threshold are stored into the HDFS file system.

V. Experiments

We performed experiments on two real datasets acquired in different domains:
AperTo dataset. This open dataset, available at http://aperto.comune.torino.it, collects information about approximately 2 millions of traffic law infractions committed in the city of Turin over the 3-year period 2011-2013. The dataset is characterized by five attributes (*Infraction name*, *Vehicle type*, *Location*, *Date*, and *Time*). Its size is approximately 198 MB. Hierarchies over the infraction data items were defined according to the guidelines reported in Section IV-B.
BigNetData dataset. This relational network traffic dataset has been obtained by performing different capture stages on a backbone link of a nation-wide ISP in Italy that offers us three different vantage points. The dataset has size 192.56 GB and it consists of 413,012,989 records, i.e., one record for each bi-directional TCP flow). A more detailed dataset description is given in [5].

The MapReduce jobs of the MGI-Cloud workflow (see Section IV) were developed in Java using the new Hadoop Java APIs. The experiments were performed on a cluster of 5 nodes running Cloudera's Distribution of Apache Hadoop (CDH4.5). Each cluster node is a 2.67 GHz six-core Intel(R) Xeon(R) X5650 machine with 32 Gbyte of main memory running Ubuntu 12.04 server with the 3.5.0-23-generic kernel. All the reported execution times are real times obtained from the Cloudera Manager web control panel.

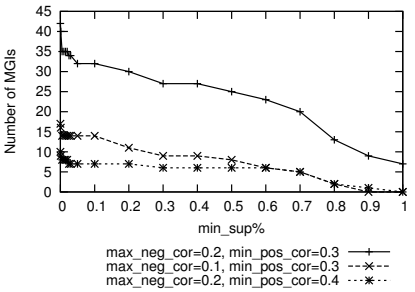

Figure 1. Effect of the minimum support threshold. max_NOD=60%.

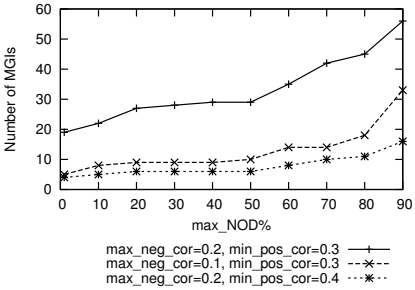

Figure 2. Effect of the maximum NOD threshold. minsup=0.02%.

In the experiments we addressed the following issues: (i) the analysis of the characteristics of the mining results achieved with different parameter settings ((Section V-A)), (ii) the validation of the usefulness of the results achieved for performing in-depth analysis (Section V-B), and (iii) the scalability of the MGI Miner algorithm with the number of nodes (Section V-C). We addressed Tasks (i) and (ii) on AperTo, because data fully complies with the context under analysis (i.e., infraction data analysis), whereas Task (iii) was addressed on BiGNetData, because it is a larger dataset characterized by a fairly complex data distribution.

A. Characteristics of the mining results

We analyzed the impact of the main input parameters of the MGI Miner algorithm on the number of MGIs mined. Figure 1 summarizes the number of mined MGIs by varying the minimum support threshold (min_sup) for different combinations of minimum positive and maximum negative correlation thresholds (max_neg_cor and min_pos_cor, respectively), while Figure 2 shows the number of mined MGIs by varying the max_NOD threshold for the same combinations of correlation threshold values.

The number of mined MGIs non-linearly increases by decreasing the minimum support threshold due to the combinatorial increase in the number of generated frequent itemsets [18]. Since most itemsets have correlation between 0.1 and 0.3, the maximum number of MGIs is extracted if max_neg_cor and min_pos_cor fall in this value range, because the generalization process is most likely to flip the itemset correlation types. As expected, the smaller the gap between max_neg_cor and min_pos_cor, the more MGIs are extracted because correlation type changes occur, on average, more frequently. For all the tested configurations, the set of mined MGIs remains still manageable by domain experts for manual inspection even while setting relatively low support thresholds (e.g., 35 MGIs mined with max_neg_cor=0.2, min_pos_cor=0.3 and min_sup=0.01%).

The number of mined MGIs non-linearly increases while increasing the maximum not overlapping degree threshold max_NOD, because low-level itemsets are more likely to cover a significant portion of data already covered by the corresponding high-level itemsets. However, in all the performed experiments the set of MGIs, which represent anomalies/contrasting situations, remains easily manageable by domain experts for manual exploration.

B. Result validation

We examined the MGIs extracted from the AperTo dataset to validate their interestingness and usefulness in a real-life context, i.e., the analysis of the traffic law infractions committed in a urban environment.

As a first example, let us consider the following MGI extracted by enforcing min_sup=0.02%, max_neg_cor=0.1, min_pos_cor=0.4, and max_NOD=60%: {(Location,Zip code 10125), (Infraction name,Prohibition)} $\triangleright$ {(Location, Sommeiller Avenue), (Infraction name, One-way infraction)}. The high-level itemset {(Location,Zip Code 10125), (Infraction name,Prohibition)} is negatively correlated, whereas its frequent descendant {(Location, Sommeiller Avenue), (Infraction name, One-way infraction)} is positively correlated and it covers a significant portion of data already covered by the high-level itemset ($\sim$59%). Hence, to a certain extent, analyzing only the traditional high-level itemset instead of the complete MGI could be misleading. This pattern indicates that in a certain area of Turin, identified by zip code 10125, a category of infractions (prohibitions) is not very likely to occur, whereas for a specific avenue within the area wrong way driving prohibition is violated more commonly than expected. Hence, road signs in Sommeiller Avenue could be either not well visible or misplaced. The public administration of Turin should deem such information to be worthy for signage maintenance and monitoring.

Let us consider now the following MGI: {(Location,District 1), (Vehicle type,Private car),(Time, p.m.)} $\triangleright$ {(Location, Zip code 10122),(Vehicle type, Private car),(Time, (4 p.m.,8 p.m.]), (Location, Zip code 10121),(Vehicle type, Private car),(Time, (8 p.m.,12 p.m.]), ...}. The high-level itemset is positively correlated, whereas 11 of its descendant itemsets are negatively correlated and the NOD value of the mined MGI is 58%. District 1 of Turin appears to be an area in which many infractions are committed by private cars during the afternoon, evening, or night. Hence, traffic corps should

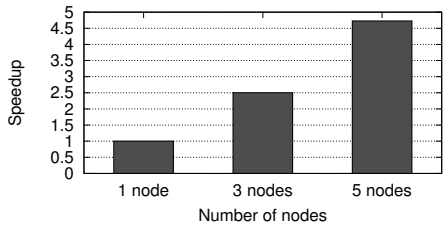

Figure 3. Speedup on the BigNetData dataset.

monitor the area more carefully in these specific daily time periods. However, in 42% of the subareas of district 1 (e.g., the ones identified by zip codes 10121 and 10122, respectively), infractions are less likely to occur than in the others. Therefore, it would be more advisable to monitor the subareas other than district 1.

In summary, MGI extraction from infraction data could help traffic corps optimize road monitoring services and identify anomalous situations be due to either inappropriate citizens' behaviors or to temporary service disruptions.

C. Scalability with the number of cluster nodes

We evaluated the scalability of the proposed architecture by measuring the speedup achieved increasing the number of Hadoop cluster nodes. Specifically, we considered three configurations: 1 node, 3 nodes, and 5 nodes. Figure 3 reports the speedup achieved setting min_sup to 1%, max_neg_cor to 0.1, min_pos_cor to 0.3, and max_nod to 60%. The first box in Figure 3 (i.e., 1 node) corresponds to a run of MGI-CLOUD on a single node. Speedup with increasing nodes is computed against the single-node performance. The achieved results show that our approach scales roughly linearly with the number of nodes and the speedup approximately corresponds to the number of cluster nodes.

VI. CONCLUSIONS AND FUTURE PERSPECTIVES

This paper presents a cloud-based service for discovering Misleading Generalized Itemsets from Big Data equipped with taxonomies. To cope with Big Data the architecture has been designed to run on a distributed Hadoop architecture [1]. A preliminary analysis of the applicability and usefulness of the proposed architecture was conducted on real Big Data acquired in a smart city environment and related to traffic law infractions. However, the offered service could find application in many other application contexts, such as (i) social network analysis, (ii) network data analysis, and (iii) financial data analysis. As future work, we aim at optimizing and extending the current Hadoop architecture as well as testing its applicability in other real-life contexts.

REFERENCES

[1] J. Dean and S. Ghemawat, "Mapreduce: simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

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

[3] H. Li, Y. Wang, D. Zhang, M. Zhang, and E. Y. Chang, "Pfp: parallel fp-growth for query recommendation," in *RecSys'08*, 2008, pp. 107–114.

[4] S. Moens, E. Aksehirli, and B. Goethals, "Frequent itemset mining for big data," in *SML: BigData 2013 Workshop on Scalable Machine Learning*. IEEE, 2013.

[5] D. Apiletti, E. Baralis, T. Cerquitelli, S. Chiusano, and L. Grimaudo, "Searum: A cloud-based service for association rule mining," in *ISPA'13*, 2013, pp. 1283–1290.

[6] E. Baralis, T. Cerquitelli, S. Chiusano, and A. Grand, "P-mine: Parallel itemset mining on large datasets," in *ICDE Workshops*, 2013, pp. 266–271.

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

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

[9] E. Baralis, L. Cagliero, T. Cerquitelli, S. Chiusano, and P. Garza, "Frequent weighted itemset mining from gene expression data," in *BIBE*, 2013, pp. 1–4.

[10] L. Cagliero, T. Cerquitelli, P. Garza, and L. Grimaudo, "Misleading generalized itemset discovery," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1400–1410, 2014.

[11] K. Sriphaew and T. Theeramunkong, "A new method for finding generalized frequent itemsets in association rule mining," in *Symposium on Computers and Communications*, 2002, pp. 1040–1045.

[12] E. Baralis, L. Cagliero, T. Cerquitelli, and P. Garza, "Generalized association rule mining with constraints," *Inf. Sci.*, vol. 194, pp. 68–84, 2012.

[13] J. Han and Y. Fu, "Mining multiple-level association rules in large databases," *IEEE Transactions on knowledge and data engineering*, vol. 11, no. 7, pp. 798–805, 1999.

[14] D. Kunkle, D. Zhang, and G. Cooperman, "Mining frequent generalized itemsets and generalized association rules without redundancy," *J. Comput. Sci. Technol.*, vol. 23, no. 1, pp. 77–102, 2008.

[15] L. Cagliero, "Discovering temporal change patterns in the presence of taxonomies," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 3, pp. 541–555, 2013.

[16] M. Barsky, S. Kim, T. Weninger, and J. Han, "Mining flipping correlations from large datasets with taxonomies," *Proc. VLDB Endow.*, vol. 5, no. 4, pp. 370–381, Dec. 2011.

[17] T. Wu, Y. Chen, and J. Han, "Re-examination of interestingness measures in pattern mining: a unified framework," *Data Min. Knowl. Discov.*, vol. 21, no. 3, pp. 371–397, Nov. 2010.

[18] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *VLDB '94*, 1994, pp. 487–499.