Towards Safer (Smart) Cities: Discovering Urban Crime Patterns Using Logic-based Relational Machine Learning*

Vítor Lourenço
Department of Computer Science
Universidade Federal Fluminense
Niterói, RJ, Brazil
vitorlourenco@id.uff.br

Paulo Mann
Department of Computer Science
Universidade Federal Fluminense
Niterói, RJ, Brazil
paulomann@id.uff.br

Artur Guimarães
Department of Computer Science
Universidade Federal Fluminense
Niterói, RJ, Brazil
arthur.sanches27@gmail.com

Aline Paes♣
Department of Computer Science
Universidade Federal Fluminense
Niterói, RJ, Brazil
alinepaes@ic.uff.br

Daniel de Oliveira♣
Department of Computer Science
Universidade Federal Fluminense
Niterói, RJ, Brazil
danielcmo@ic.uff.br

***Abstract*—Smart cities initiatives have the potential to improve the life of citizens in a huge number of dimensions. One of them is the development of techniques and services capable of contributing to the enhancement of security public policies. Finding criminal patterns from historical data would arguably help in predicting and even preventing thefts and burglaries that continuously increase in urban centers worldwide. However, accessing such history *and* finding patterns across the interrelated crime occurrences data are challenging tasks, particularly to underdevelopment countries. In this paper, we address these problems by combining three techniques: we collect crime data from existing crowd-sourcing systems, we automatically induce patterns with relational machine learning, and we manage the entire process using scientific workflows. The framework developed under these lines is named CRiMINaL (Crime patteRn MachINe Learning). Experimental results conducted from a popular Brazilian source of data and a traditional relational learning system shows that CRiMINaL is a promising tool to induce interpretable models that can assist police departments on crime prevention.**

***Index Terms*—Smart Cities, Crime Patterns, Relational Machine Learning, Scientific Workflows.**

I. INTRODUCTION

The definition of the Smart City concept has led to several new initiatives worldwide in the last years, such as the well-known MIT City Science program[1] and IBM Smarter Cities[2]. Smart Cities can be defined as complex systems involving people with a variety of expertises, interacting and using a set of services to improve the development of the city, and, consequently, the quality of life for their citizens [1], [2]. According to Fundação Getúlio Vargas (FGV)[3], there are 10 dimensions that indicate the level of intelligence of a specific city: governance, public administration, urban planning, technology, environment, international connections, social cohesion, human capital and economy. In this paper, we focus on a specific dimension: the public administration, or more specifically, in security issues in public administration.

Smart cities are, as a concept, safe cities. As such, providing solutions for reducing urban criminality is a top priority. The urban criminality is an old, yet open, issue in many countries, specially underdevelopment ones [3]–[5]. Brazil is an example of underdevelopment country with high urban criminality rates. In several Brazilian cities (such as Rio de Janeiro, Recife and São Paulo) citizens daily face robberies and thefts, among other urban crimes[4] [5] [6] [7]. Due to this fact, urban crimes have become a priority in government's agendas[8].

Let us consider as an example the city of Niterói (this city will be consistently used as example throughout this paper), in the state of Rio de Janeiro, Brazil. Niterói has 52 neighborhoods, population of 487,562 citizens and an area of 133.919 square kilometers[9]. Although Niterói presents the best Human Developing Index (HDI) amongst all cities in the state of Rio de Janeiro, the number of crimes is increasing in a fast pace in recent years. According to reports of the Rio de Janeiro Public Security Institute (ISP)[10] homicides, car theft, and robbery have increased in the last years. For example, from the year of 2011 to the year of 2015, the pedestrian robberies increased at the rate of 88.8% (from 6,573 cases to 12,420). Moreover, cell phones thefts increased at a rate of 60% in the year of 2015 in comparison with the year of 2014.

*Authors would like to thank CNPq, CAPES and FAPERJ for partially sponsoring this research

[1] https://www.media.mit.edu/groups/city-science/overview/

[2] https://www.ibm.com/smarterplanet/us/en/smarter_cities/overview/

[3] http://fgvprojetos.fgv.br/noticias/o-que-e-uma-cidade-inteligente (in Portuguese)

[4] https://g1.globo.com/rio-de-janeiro/noticia/numero-de-assaltos-no-grande-meier-no-rio-cresceu-84-este-ano.ghtml

[5] https://g1.globo.com/rj/rio-de-janeiro/noticia/motorista-reage-a-assalto-e-joga-carro-em-ladroes-que-atiram-veja-video.ghtml

[6] https://g1.globo.com/mg/zona-da-mata/noticia/onibus-com-destino-a-juiz-de-fora-e-assaltado-no-rio-de-janeiro.ghtml

[7] http://videos.band.uol.com.br/14241736/assaltos-terminam-em-morte-no-rio-de-janeiro.html

[8] http://arquivos.proderj.rj.gov.br/isp_imagens/Uploads/LegislacaoISP001.pdf

[9] https://cidades.ibge.gov.br/painel/painel.php?codmun=330330

[10] www.isp.rj.gov.br/

978-1-5090-6014-6/18/$31.00 ©2018 IEEE

One possible reason for such high crime rates in Niterói is that criminal groups have been pushed out Rio de Janeiro city due to pacification programs[11].

Clearly, it is a complex task to reduce the number of crimes in a big city such as Niterói. Such cities have many neighborhoods and districts that present quite different crime patterns. One possible (and fruitful) action for reducing crimes is ostensible policing. The goals of ostensible policing in a modern democracy are to uphold the rule of law and safeguard human rights according to Legran and Bronitt [6]. Thus, police departments have to identify such patterns and plan the actions (where to place police officers, for instance) according to identified patterns. However, it is far from trivial to identify such patterns in different neighborhoods or districts of the same city. Identifying such patterns requires a variety of data (crime type, locality, date/time, stolen objects, *etc.*) that may be not easily gathered and related.

The idea of identifying crime patterns in several neighborhoods and/or districts of a specific city in order to plan actions is extremely interesting and useful to citizens and governments. Such pattern identification allows for police departments to place police officers in a better coordinate way. In addition, there is a large amount of urban crime data available for analysis, either by government institutes such ISP or by websites that map crimes using crowd-sourcing such as *Onde fui Roubado*[12] (Where I was Robbed, in free translation), and *WikiCrimes Mobile*[13]. Unfortunately, such approaches only visualize crime occurrences, without a deep analysis of an interrelationship between occurrences in the various regions of a city, locations, and types of crimes.

Thus, aiming at helping police departments to identify crime patterns, in this paper we propose an approach named CRiMINaL (Crime patteRn MachINe Learning). The purpose of CRiMINaL is to generate predictive and interpretable models to inform police authorities of common crime patterns in city areas, in specific date/time, and near some specific locations. Thus, with the development of CRiMINaL, we intend to provide to the police authorities a new crime analytical capability. CRiMINaL provides a simple Web interface that can be adapted for each city and can be deployed in the cloud, thus ensuring its availability. This interface is based on crowd-sourcing data, which can be collected through web forms and enriched with external sources such as ISP database. This way, all information is provided by citizens, thus improving the interaction between citizens and government. The data processing in CRiMINaL is managed by the SciCumulus workflow management system [7] that executes all preprocessing and predictive model generation activities, and also runs in the cloud. In order to generate the predictive models, CRiMINaL relies on machine learning algorithms, more specifically on Logical-Relational Learning through Inductive Logic Programming (*i.e.* ILP) [8]. ILP is used since traditional Machine Learning algorithms such as C4.5 are not well-suited to process heterogeneous, related data. This way, all data collected is converted into logical representation to form a set of examples and use the ILP Aleph system [9]. The predictive models are generated based on three specific points: type of crime, where (place) and when (time) crimes occurred.

The experimental evaluation of CRiMINaL was carried out using data obtained from the *Onde fui Roubado* web site for the city of Niterói and with data collected using CRiMINaL crowd-sourcing web interface. The obtained results show an accuracy of the generated predictive model over 80% while the percentage of false positive occurrences remained lower than 18%. These results shows that the approach is promising.

This paper is organized in 5 sections besides this introduction. Section 2 presents background knowledge on ILP and workflows. In Section 3 we present the proposed approach named CRiMINaL. The experimental evaluation is discussed in Section 4. Section 5 discusses related work, and finally Section 6 concludes this paper and points future work.

II. BACKGROUND KNOWLEDGE

In this section we overview the techniques used to develop CRiMINaL, namely, logic-based machine learning concepts, used for discovering criminal patterns from the historical data, and the workflow system used for managing the whole process.

A. *Logic-based Relational Learning*

Machine Learning algorithms aim at automatically inducing patterns from data [10]. Classical machine learning algorithms such as Decision Trees, Logistic Regression, and Support Vector Machines, to name a few, assume that the examples extracted from the data are independent and homogeneously distributed. In this way, they are represented as a matrix, where each line i is an example, each row j is an attribute of the example, and each cell i, j has a value for the attribute j and example i. However, real world data such as the crime occurrences addressed in this paper, are heterogeneous, in the sense that an example may have an attribute that is not shared for all the other examples. Moreover, real world data are commonly relational, in the sense that the entities and their attributes are related to each other in different examples. Such relationships would not have been properly handled if one assumes that the examples are independent from each other.

Relational Machine Learning [8], [11] is a subfield of Machine Learning that has as goal to induce patterns from relational examples not represented as attribute-value pairs. When it is necessary to have clear explanations and expressive hypothesis, one may rely on logic-based representations [12] and take advantage of algorithms that induce logical programs, developed in the area of Inductive Logic Programming area [8], [13], [14]. In addition to the set of examples, it is also possible to use a preliminary knowledge about the domain, named as Background Knowledge, both expressed as definite clauses [15]. The set of examples is usually divided into positive and negative facts, i.e., definite clauses with no

[11] https://oglobo.globo.com/rio/traficantes-da-mangueira-se-mudam-para-niteroi-3598400

[12] http://www.ondefuiroubado.com.br/

[13] http://www.wikicrimes.org/

negative literals, into the format $p(c_1, \ldots, c_n)$, where p is a predicate and $c_1, \ldots, c_n$ are constants of the domain. The output of an ILP system is a logical program, ideally covering all the positive examples while not covering the negative examples.

In this paper, we have used the ILP system Aleph [9], as it implements a number of standard ILP algorithms. Aleph relies on the concept of a most specific clause, known as the Bottom Clause [16], to build the clauses that are going to constitute the induced logical program. The Bottom Clause $\perp(e)$ with regard to a positive example e and background theory BK is the most specific clause within the hypothesis space that covers the example e, i.e., $BK \cup \perp(e) \vDash e$.

Any single clause covering the example e with regard to BK must be more general than $\perp(e)$. Any clause that is not more general than $\perp(e)$ cannot cover e and can be safely disregarded. Thus, the bottom clause bounds the search for a clause covering the example e, as it captures all relevant information to e and BK.

As a typical ILP system, Aleph has as goal to induce a logical program that covers as much as possible positive examples of the domain, while making as few as possible of the negative examples unprovable. To generate each candidate clause to be part of the logical program, Aleph starts from an uncovered positive example e, builds $\perp(e)$, and specializes a most general clause $\mathcal{C}$, by including literals from the Bottom Clause in the body of $\mathcal{C}$. The literals are selected from $\perp(e)$ accordingly the score computed from some evaluation function, such as $accuracy$. Aleph inserts literals in $\mathcal{C}$ as long as there is some improvement in the score, but constraining the coverage of negative examples to a maximum number, defined as the $noise$ parameter. Thus, if the $noise$ parameter is 0, then not a single negative example is allowed to be covered by the clause.

After reaching the end of this construction process, the yielded clause is included at the hypothesis $\mathcal{H}$ and the process restarts. The default setting of Aleph removes from the set of examples all the positive ones covered by the current hypothesis, but keeps considering all the negative examples during the whole process, since it is not a problem to produce clauses covering positive examples already covered before, but it is a problem to produce a single proof for a negative example.

B. Modeling the Knowledge Discovery Process Using Workflows

The process of knowledge discovery from data requires a chaining of multiple combinations of programs, usually consuming large amounts of data. Each program has as input a specific group of parameters and data, and its outputs may be used as input to another program in the flow. In order to manage the execution of such programs chaining, together with its inputs and outputs, one can model a scientific workflow [17], instead of relying on ad-hoc or scripts-based approaches. In general, a workflow is defined as an abstraction for modeling the flow of activities and data in an experiment [17]. When referring to a computational experiment, the activities are programs or services representing algorithms and solid computational methods. Due to the inherent complexity of the execution of multiple programs, it is recommendable to run them in parallel, in distributed computational environments, such as clusters, grids, and clouds [18], [19]. To configure, manage, and control the execution of entire workflows, it is necessary to rely on Workflows Manager Systems [17]. In this paper, we take advantage of a well-known Scientific Workflow Manager System named SciCumulus [20], since it has been largely used to manage several types of Scientific Workflows [20], [21].

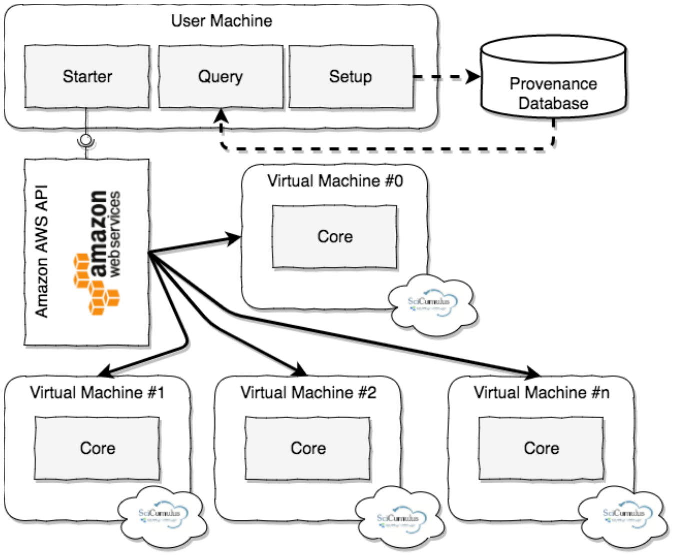


Fig. 1. Conceptual Architecture of SciCumulus

SciCumulus is composed by four main components (Fig. 1): Core, Setup, Starter and Query Processor. The *Setup* is responsible for storing and retrieving prospective provenance (the structure of the workflow) to/from the provenance database. Using this component, scientists inserts/updates the structure of the workflows in the database. When the structure of the workflow is already loaded in the provenance database, *Starter* can be invoked. Starter component is responsible for configuring the environment for executing the workflow. In case of executions in cloud environments, Starter is responsible for deploying virtual machines and configuring storage services before the workflow execution. Starter has to be compatible to the cloud API. In the current version, Starter works with Amazon AWS API.

When all virtual machines and storage services are running, Starter invokes SciCumulus Core (or simply Core) in each virtual machine. Core is a MPJ (MPI-like)[14] application, running in all virtual machines at the same time with message passing interface. The Core component is responsible for scheduling the workflow activities in the several virtual machines. In addition, the Core is responsible for collecting retrospective

[14] http://mpj-express.org/

provenance data (historical data about the execution of the workflow) and store it in the provenance database. The Query processor component is responsible for querying the provenance database during or after the workflow execution. It can be used by scientists to steer the workflow or to perform a *post-mortem* analysis of the results. A high-level conceptual architecture is summarized in Figure 1. For more information about SciCumulus please refer to Oliveira *et al.* [20].

III. CRiMINaL: An Approach for Identifying Urban Crime Patterns

This section introduces the workflow for identifying crime patterns and the architecture of CRiMINaL framework that encompasses the modeled workflow.

A. A Workflow for Crime Pattern Identification

The CRiMINaL-Wf (Fig. 4) is the workflow executed by CRiMINaL approach. CRiMINaL-Wf is composed of 4 activities, namely: *Data Acquisition*, *Data Conversion*, *Data Normalization*, and *Predictive Model Learning*.

The *Data Acquisition* activity consists of obtaining raw data from an external data source, or Web services/sites and applications that provide crime occurrences data. This activity executes a crawler that should be customized for each external data source. In the experiment presented in Section IV, data was obtained from the website *Onde fui Roubado* and from the Data Ingestion Module of CRiMINaL (presented in the next subsection). Locality data was obtained through the Web service provided by *Open Street Maps*[15]. For each crime occurrence, the *Data Acquisition* activity defines its geolocation and associates the occurrence with the type of crime, the list of stolen objects, date, and time at which the event occurred.

The *Data Conversion* activity is responsible for converting data to a comma-separated format (*i.e.* csv) to be further processed in the next activities. In this way, the *Data Conversion* activity produces 2 tables corresponding to the crime occurrences and locations. Besides generating these 2 tables, the *Data Conversion* activity also collects information about the surroundings of a location where the crime occurred. This surrounding information is collected by using Haversine formula [22]. Due to proximity characteristics between 2 locations, a maximum radius of 500 meters was adopted for considered to be the surroundings of an occurrence.

The *Data Normalization* activity consists of the discretization of data, transforming them into non-continuous values [23]. This activity is fundamental for the *Predictive Model Learning* activity. In this way, data such as *latitude*, *longitude*, *day* and *hour* is transformed into labeled intervals. The surroundings information and the type of occurrence were kept in the original form, since they are already discrete values. The discretization performed by *Data Normalization* activity is presented in Table I where *Neighborhood* represents the neighborhood names of a city; *Day* represents the type of day where the crime occurred (Business day, Weekend or Holidays) and *Hour* is the period of the day when the crime occurred (*Dawn* refers to the time period from 00:00 to 05:59 a.m., *Morning* corresponds to the period from 06:00 to 11:59 a.m., *Afternoon* corresponds to the period from 12:00 a.m. to 5:59 p.m., and finally *Evening* corresponds to the period from 6:00 p.m. to 11:59 p.m.).

TABLE I
Data Discretization

Non-discretized data	Discretized data
Latitude/Longitude	Name of the Neighborhood (*e.g.* Copacabana)
Day	Business day / Weekend / Holiday
Hour	Dawn / Morning / Afternoon / Evening

Once all data is discretized it can be visualized in the Web portal of CRiMINaL (the portal will be presented following). However, until this point, no predictive model has been generated by the workflow. In order to generate such models, the discretized data is used to produce a new table that is the input of the *Predictive Model Learning* activity. Table II presents the attributes used by the *Predictive Model Learning* activity to generate the predictive model. Note that is this format each object is the head of a column, producing a possibly very sparse table. We avoid this problem by relying on a relational learning system, that allows heterogeneous examples.

In the *Predictive Model Learning* activity the Aleph system is invoked by the SciCumulus workflow engine to identify crime patterns in different locations of a city. In order to execute Aleph, it is necessary to transform the normalized data to a representation in first-order logic (FOL). All crime occurrences are considered as positive examples for Aleph. The examples are constructed with the identifier associated with the crime occurrence and its associated type. Let us consider the $occurrence(30, theft)$ as a positive example, where 30 is the identifier of the occurrence and *theft* is the type of the crime. To automatically generating negative examples, we assume a closed-world assumption [24], where for each occurrence in our knowledge base (*i.e.* positive example), the types of occurrences that did not happen become negative examples. Let us also consider that for the $occurrence(30, theft)$, the objects stolen are a cell phone and a wallet, the crime occurred on a Monday, in the morning, next to a university and a restaurant. The FOL representation is in Figure 2.

```
neighborhood(30,inga).
object_stolen(30,cell_phone).
object_stolen(30,wallet).
day(30,business_day).
hour(30,morning).
surroundings(30,university).
surroundings(30,restaurant).
```

Fig. 2. Part of the Contextual Background Knowledge Related to the Example $occurrence(30, theft)$, Given as Input to the Aleph System

When Aleph system is executed, it produces as output a logic program. Each rule of this logic program is a defi-

[15] http://openstreetmaps.org

nite clause, that is, a clause with only one positive literal. Such positive literal is in the head of the clause and is a generalization of the positive examples, *i.e.*, the identification of the occurrence is a connection variable and the type of occurrence is a constant. The occurrence type can also be another variable, but it was defined as constant here, in order to better characterize the types of occurrences. The body of the clause is a conjunction of literals also generalized according to the bias of language. In this case, the ID is always replaced by a variable, but the other terms can be variables or constants. That is defined by the learning process via Aleph, during the construction of the rule. Thus, the rule presented in Figure 3 is an example of a logic program learned by the Aleph system, indicating that if a person is in the neighborhood of Icaraí, with a credit card and next to a hospital, there is a chance of a theft occurs.

```
occurrence(A,theft) :- neighborhood(A,icarai),
object_stolen(A,credit_card),
surroundings(A,hospital).
```

Fig. 3. An example of a Rule Induced by the Aleph System

B. Architecture of CRiMINaL

The architecture of CRiMINaL is composed of 2 layers (Figure 5): Client Layer and Server Layer. Modules in the client layer execute in the local web browser while modules in the server layer can be deployed in any server or in the cloud.

The first module to be invoked is the *Data Ingestion* module that is implemented as a crowd-sourcing web interface. The *Data Ingestion* module should be customized for each different city (to comply with local agenda, etc.). In its current version, we implemented a web interface for the city of Niterói (which was chosen as case study of the paper). The web interface for Niterói was named SiAPP [25] (public available at http://siapp.ic.uff.br:8082/) and allows for citizens to provide data about crime occurrences in Niterói (Step 1 in Figure 6). Figure 6 presents an excerpt of the web interface (in Portuguese) where citizens can choose an address where crime occurred and then can provide information about the crime (type, objects stolen, etc.). SiAPP web interface was developed using Bootstrap and it stores collected data in a MongoDB database.

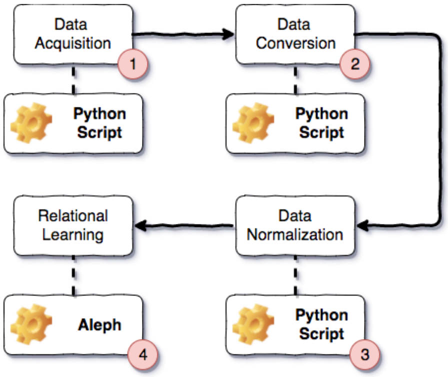


Fig. 4. The CRiMINaL-Wf Worflow

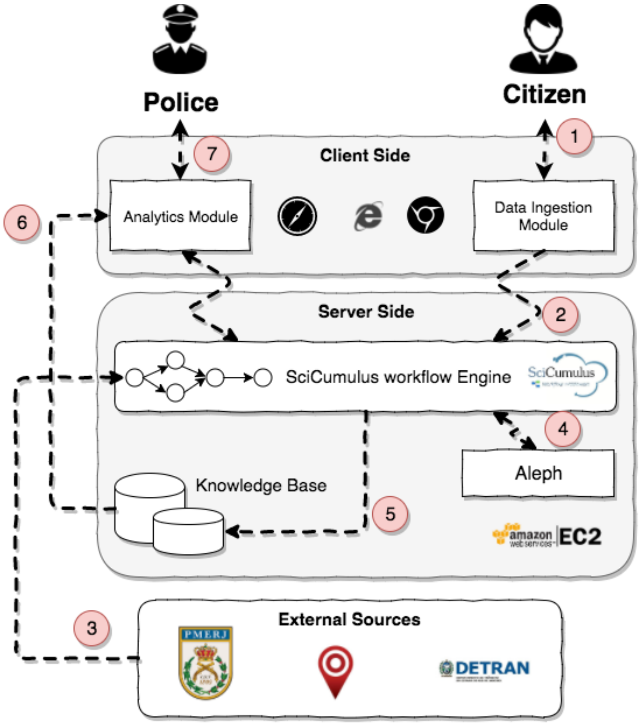
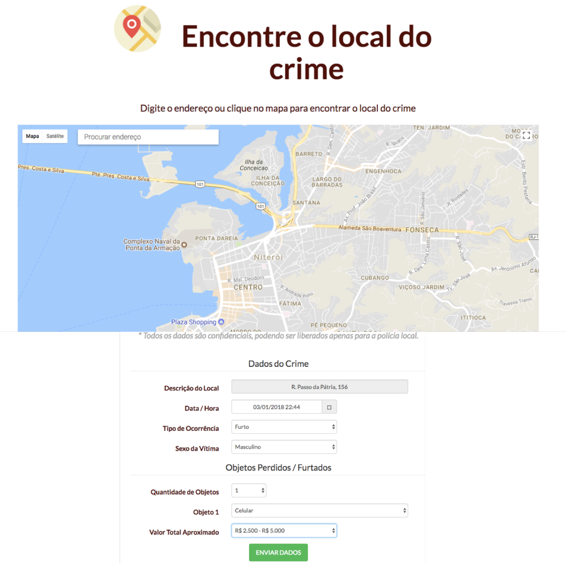


Fig. 5. CRiMINaL Conceptual Architecture


Encontre o local do crime
Digite o endereço ou clique no mapa para encontrar o local do crime
Mapa
Satélite
Procurar endereço
Niterói
CENTRO
Plaza Shopping

* Todos os dados são confidenciais, podendo ser liberados apenas para a polícia local.

Dados do Crime

Descrição do Local: R. Passo da Pátria, 156

Data / Hora: 03/01/2018 22:44

Tipo de Ocorrência: Furto

Sexo da Vítima: Masculino

Objetos Perdidos / Furtados

Quantidade de Objetos: 1

Objeto 1: Celular

Valor Total Aproximado: R$ 2.500 - R$ 5.000

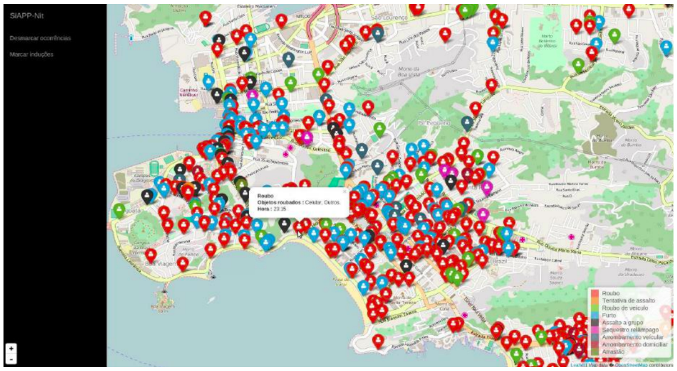
ENVIAR DADOS

Fig. 6. The SiaPP Crowd-sourcing Web Interface

Once data is stored it can be further processed by the modules in the server layer. The server layer is composed of the SciCumulus Workflow System, the Aleph System and the Knowledge Base. SciCumulus is responsible for executing

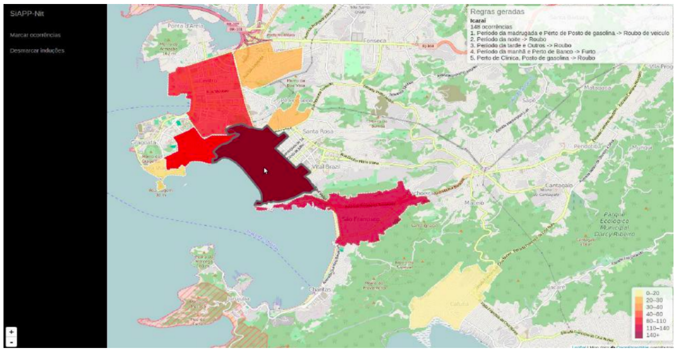
TABLE II
STRUCTURED CONTENT OF OCCURRENCES

Type of Occurrence	Neighborhood	Day	Hour	Surroundings	Document	Cell Phone	Car	Money	Wallet
Theft	Ingá	Business Day	Dawn	Market	TRUE	TRUE	FALSE	TRUE	TRUE
Theft	Icaraí	Weekend	Evening	Pub	FALSE	FALSE	TRUE	FALSE	FALSE
Burglary	Centro	Business Day	Evening	Shopping	TRUE	TRUE	FALSE	TRUE	TRUE
Kidnapping	Icaraí	Business Day	Evening	Market	FALSE	FALSE	FALSE	FALSE	FALSE



Roubo
Objetos roubados : Celular, Outros.
Hora : 23:15

Fig. 7. Visualization of Crime Occurrences in the Web Interface



Regras geradas
Icarai
148 ocorrências
1. Período da madrugada e Perto de Posto de gasolina -> Roubo de veiculo
2. Período da noite -> Roubo
3. Período da tarde e Outros -> Roubo
4. Período da manhã e Perto de Banco -> Furto
5. Perto de Clinica, Posto de gasolina -> Roubo

Fig. 8. Visualization of Crime Patterns in the Web Interface

the workflow CRiMINaL-Wf which aims at generating the predictive model. Every time the user accesses CRiMINaL and requests the predictive model to be generated or updated, SciCumulus is invoked (step 2). SciCumulus then instantiates a number of virtual machines in the cloud and executes *Data Acquisition*, *Data Conversion*, *Data Normalization*, and *Predictive Model Learning* activities, as previously discussed. Each activity is associated with the invocation of Python Scripts to import data from external sources and databases (step 3) and Aleph (step 4) as detailed in the previous subsection. One of the advantages of using a workflow system in CRiMINaL architecture is that if the process evolves or is modified, only a change in the workflow specification is needed, without requiring to modify other modules of CRiMINaL. SciCumulus is implemented in Java but can be invoked in a command line. All data handled by Aleph is represented using Prolog (step 5).

When the predictive model is generated by Aleph, the *Analytics* module can be invoked (step 6). Similarly to the *Data Ingestion* module, the *Analytics* module is implemented in a web interface. Figure 7 presents the visualization of crime occurrences in the map of Niterói. This functionality only plots a point for each occurrence in the map. On the other hand, Figure 8 presents the generated rules (identified crime patterns) identified for each neighborhood of Niterói. Figure 8 presents some patterns identified for the neighborhood of Icaraí such as *morning AND near bank* → *theft* and *dawn AND near gas station* → *car theft*. CRiMINaL is open sourced at https://github.com/UFFeScience/CRiMINaL.

IV. EVALUATION OF CRIMINAL

In order to evaluate CRiMINaL, in this paper, the city of Niterói was chosen as a case study. As previously mentioned, for this experimental evaluation the data was obtained from the web site *Onde fui Roubado* and from the SiAPP web interface presented in the previous section. Although the site *Onde fui Roubado* has crime occurrences of several cities, we selected only those from the city of Niterói. 781 crime occurrences were extracted and 50 crime occurrences were obtained from SiAPP crowd-sourcing interface.

To evaluate CRiMINaL, we adopted the experimental methodology which is based on the evaluation metrics of the generated predictive model. We used the validation technique k-fold cross validation [10]. Using k-folds cross validation avoids that the model learned is specialized only for a subset of crime occurrences. This technique consists of dividing the data into k mutually exclusive subsets, reserving a set to validate the predictive model, and all others *(k-1)* for training. In this paper, we adopted $k=10$ (10 folds) and the mean of the predictive results was computed from 10 executions in order to produce a single estimation. The chosen evaluation function was m-estimate [9], which is appropriate to handle noisy data.

The experimental results were analyzed from 2 perspectives: qualitative and quantitative. In the qualitative analysis (Figure 9) we compared the identified crime patterns (rules generated) with official ISP statistics and press reports of crime occurrences in the city of Niterói. In this way, we selected as examples patterns that are supported by data presented by ISP. For example, according to **Rule 1**, the number of thefts is high in the regions surrounding the University campus located between the Ingá, Centro and Boa Viagem neighborhood. In the same neighborhood, there are 2 more universities and several high schools. **Rule 3** indicates that there was an increase in the occurrence of shoplifting in schools in business days. Such rules, automatically discovered by CRiMINaL, are corroborated by ISP, which informs that the number of

TABLE III
CONFUSION MATRIX

Metric	Mean	Standard Deviation
Accuracy	83.54%	2.30%
Sensitivity	82.65%	4.45%
Precision	72.07%	3.07%
F-Measure	77.68%	3.42%
Miss Rate	17.35%	4.45%

robberies in such neighborhoods have also increased. Other example is **Rule 5**, which indicates the incidence of wallet theft in the Centro area at night. This region is composed of several bars and pubs, and in fact the most common crime is the theft of backpacks and wallets. The statistics presented by ISP corroborate the results of the predictive model.

```
(1) day(business_day), object_stolen(cell_phone),
    surroundings(university) -> occurrence(theft)

(2) neighborhood(inga), hour(afternoon) ->
    occurerence(theft)

(3) neighborhood(inga), day(business_day),
surroundings(school) -> occurrence(theft)

(4) neighborhood(boa_viagem) -> occurrence(theft)

(5) neighborhood(centro), objet_stolen(wallet),
hour(evening) -> occurrence(robbery)
```

Fig. 9. Examples of Crime Patterns Identified by CRiMINaL for the City of Niterói

For the quantitative analysis, the results of the confusion matrix (the confusion matrix of a model offers an effective measure of the predictive model, by showing the number of correct classifications versus predictions for each class) of the 10 folds and their accuracy, precision, sensitivity, F-measure (also called F-score, is a weighted average of precision and sensitivity) and the false negatives (miss rate). The mean and standard deviation results are also presented in Figure 10 and Table III.

According to the results presented by Table III and Figure 10 we can view that the identified crime patterns reached an average accuracy of 83.54%, 72.07% of precision and 82.65% of sensitivity. Accuracy and sensitivity are focused on positive examples that were correctly classified, which in the case of the CRiMINaL is essential, as they represent the possible crime occurrences. This result is corroborated by the F-measure (77.68%), which also considers sensitivity. In relation to the miss rate (17.35%), it was not as low as desired, which may have been influenced by the number of negative examples artificially generated since we are using the closed-world assumption. Although more tests are still needed with a larger dataset, both quantitative and qualitative results obtained are promising and show the utility and potential of the CRiMINaL approach.

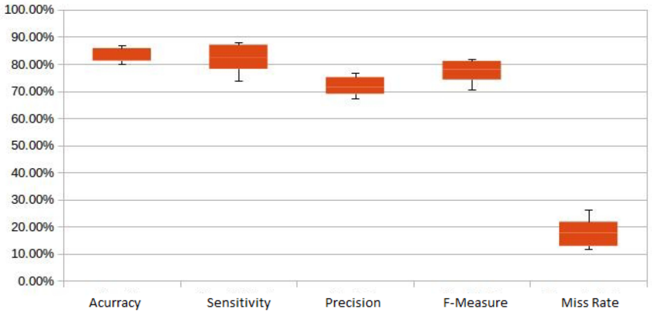


Fig. 10. Mean and Standard Deviation of the Quantitative Results

V. RELATED WORK

Public security is a top priority both for industry and scientific community, mainly because the potential benefits that developments in this area can provide to the society and their citizens.

Arguably, the most prominent approach is the *PredPol* system[16], a commercial software used by the Atlanta Police Department. *PredPol* considers the history of crimes to predict the incidence of new crimes in the city. Although its apparent success, as the system has been capable of helping on reducing 32% of the burglaries in that city, it is not clear to the general public how the system operates to make the predictions.

Other simpler initiatives used by police departments worldwide include the crime maps provided by the United Kingdom Police[17] and the New York Police department[18]. However, such approaches only present crime occurrences in visual maps, not providing ways to identify patterns or predicting crimes.

Zhang *et al.* [26] addresses the crime prevention problem by introducing a system for predicting criminal activities based on the theory of games and dynamic Bayesian networks. Another similar approach is proposed by Nath [27]. This approach is based on detecting patterns of crimes using a system based on traditional clustering techniques such as K-means. Parvez [28] *et al.* proposes a system to predict crimes based both on space and time proximity of past crimes for predicting a future crime. Turedi [29] proposes an approach to identify relationship between socio-economic factors and crime, and to understand the underlying criteria of new police station locations decisions. Buczak and Gifford [30] proposes the usage of association rule mining for community crime pattern discovery.

Although the existing approaches represent a step forward, they all use traditional data mining algorithms to identify crime patterns. Traditional data mining algorithms look for patterns in a single table (propositional patterns). However, the identification of crime patterns may involve data that are not best represented in a single table. For example, we have to consider types of stolen objects in several crime

[16] http://www.predpol.com/
[17] https://www.police.uk
[18] (https://maps.nyc.gov/crime/)

occurrences. Using traditional data mining algorithms each object type would be an attribute and the value (TRUE or FALSE) indicates if the object was stolen or not. However, when the types of stolen objects increase in a fast pace, it may be impractical to create such table. Furthermore, the data in such a table would be very sparse, which is a problem for most of the traditional approaches. This way, several tables would have to be used in the crime pattern identification. One of the advantages of CRiMINaL is that it is based on relational data mining algorithms that look for patterns among multiple tables (relational patterns).

VI. CONCLUSIONS

Unfortunately, in the last years citizens living at the great urban centers have suffered from significantly increased violence waves. Techniques and services developed by Smart Cities initiatives have the potential to provide solutions to alleviate this problem. In this work we made one step in that direction by devising the CRiMINaL (Crime patteRn MachINe Learning) framework focusing on automatically finding criminal patterns. Our solution benefited from a relational machine learning method so that the crime occurrences are represented as heterogeneous and related examples. Moreover, as the method induces interpretable logical rules, both the common citizen and the government can take the model into account, the former to become aware of the most dangerous places and times, and the latter to accurately plan public policies. Furthermore, all the data collection and manipulating, together with the learning process, and visualization are defined as a workflow and managed by the SciCumulus system. Experimental results gathered from crime occurrences informed by the citizens in a large urban city in Brazil have demonstrated a high accuracy and are consistent with the statistics provided by the Public Security Institute and the news published in the press.

As future work, we would like to adopt statistical relational learners [31], [32], since they handle partial observations and noisy better than purely relational logical approaches, but still inducing interpretable models. In addition, we intend to use official crime occurrence data to train the models.

REFERENCES

[1] J. M. Shapiro, “Smart cities: quality of life, productivity, and the growth effects of human capital,” *The review of economics and statistics*, vol. 88, no. 2, pp. 324–335, 2006.

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

[3] J. Baldwin, “Urban criminality and the problemestate,” *Local Government Studies*, vol. 1, no. 4, pp. 12–20, 1975.

[4] P. Sociales, “Crime as a social cost of poverty and inequality: a review focusing on developing countries,” *Facets of Globalization*, p. 171, 2001.

[5] G. Gribanova, R. Vulfovich *et al.*, “Modern city safety as a complex problem,” *Public administration issues*, no. 5, pp. 83–100, 2017.

[6] T. Legrand and S. Bronitt, “Policing to a different beat: measuring police performance,” in *Policing and Security in Practice*. Springer, 2012, pp. 1–19.

[7] D. de Oliveira, E. Ogasawara, F. Baião, and M. Mattoso, “Scicumulus: A Lightweight Cloud Middleware to Explore Many Task Computing Paradigm in Scientific Workflows,” in *3rd International Conference on Cloud Computing*, 2010, pp. 378–385.

[8] L. De Raedt, *Logical and relational learning*. Springer Science & Business Media, 2008.

[9] A. Srinivasan, “The aleph manual: version 4 and above,” *URL: http://www. comlab. o x. ac. uk/activities/machinelearning/Aleph/(accessed 10.09. 10)*, 2007.

[10] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, *Machine learning: An artificial intelligence approach*. Springer Science & Business Media, 2013.

[11] J. Struyf and H. Blockeel, *Relational Learning*. Springer US, 2010.

[12] R. J. Brachman, H. J. Levesque, and R. Reiter, *Knowledge representation*. MIT press, 1992.

[13] S. Muggleton, “Inductive logic programming,” *New generation computing*, vol. 8, no. 4, pp. 295–318, 1991.

[14] J. Picado, A. Termehchy, A. Fern, and P. Ataei, “Schema independent relational learning,” in *Proceedings of the 2017 ACM International Conference on Management of Data, SIGMOD Conference 2017*. ACM, 2017, pp. 929–944.

[15] U. Nilsson, *Logic, programming and Prolog*, 1990.

[16] S. Muggleton, “Inverse entailment and progol,” *New Generation Comput.*, vol. 13, no. 3&4, pp. 245–286, 1995.

[17] E. Deelman, G. Singh, M. Livny, B. Berriman, and J. Good, “The cost of doing science on the cloud: The montage example,” in *Proc. of the SC’08*, 2008, pp. 50:1–50:12.

[18] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, “A break in the clouds: Towards a cloud definition,” *SIGCOMM Rev.*, vol. 39, no. 1, pp. 50–55, Dec. 2008.

[19] D. de Oliveira, F. A. Baião, and M. Mattoso, *Towards a Taxonomy for Cloud Computing from an e-Science Perspective*. London: Springer London, 2010, pp. 47–62.

[20] D. de Oliveira, K. A. C. S. Ocaña, F. A. Baião, and M. Mattoso, “A provenance-based adaptive scheduling heuristic for parallel scientific workflows in clouds,” *J. Grid Comput.*, vol. 10, no. 3, pp. 521–552, 2012.

[21] K. A. Ocaña, D. de Oliveira, E. Ogasawara, A. M. Dávila, A. A. Lima, and M. Mattoso, “Sciphy: a cloud-based workflow for phylogenetic analysis of drug targets in protozoan genomes,” in *2011 BSB*. Springer, 2011, pp. 66–70.

[22] C. C. Robusto, “The cosine-haversine formula,” *The American Mathematical Monthly*, vol. 64, no. 1, pp. 38–40, 1957.

[23] S. Zhang, C. Zhang, and Q. Yang, “Data preparation for data mining,” *Applied artificial intelligence*, vol. 17, no. 5-6, pp. 375–381, 2003.

[24] J. Minker, “On indefinite databases and the closed world assumption,” in *International Conference on Automated Deduction*. Springer, 1982, pp. 292–308.

[25] V. Lourenço, P. Mann, A. Paes, and D. de Oliveira, “Siapp: Um sistema para análise de ocorrências de crimes baseado em aprendizado lógico-relacional,” 2016.

[26] C. Zhang, A. Sinha, and M. Tambe, “Keeping pace with criminals: Designing patrol allocation against adaptive opportunistic criminals,” in *Proceedings of the 2015 international conference on Autonomous agents and multiagent systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2015, pp. 1351–1359.

[27] S. V. Nath, “Crime pattern detection using data mining,” in *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, ser. WI-IATW ’06. IEEE Computer Society, 2006, pp. 41–44.

[28] M. R. Parvez, T. Mosharraf, and M. E. Ali, “A novel approach to identify spatio-temporal crime pattern in dhaka city,” in *Proc. of the 8th International Conference on Information and Communication Technologies and Development*, ser. ICTD ’16. ACM, 2016, pp. 41:1–41:4.

[29] S. Turedi, “Spatial analysis of ohio police station locations using geographical information systems,” in *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications*, ser. COM.Geo ’12. ACM, 2012, pp. 23:1–23:6.

[30] A. L. Buczak and C. M. Gifford, “Fuzzy association rule mining for community crime pattern discovery,” in *ACM SIGKDD Workshop on Intelligence and Security Informatics*, ser. ISI-KDD ’10. ACM, 2010, pp. 2:1–2:10.

[31] L. Getoor and B. Taskar, *Introduction to statistical relational learning*. MIT press, 2007.

[32] G. Farnadi, S. H. Bach, M. Moens, L. Getoor, and M. D. Cock, “Soft quantification in statistical relational learning,” *Machine Learning*, vol. 106, no. 12, pp. 1971–1991, 2017.