SiAPP: Um Sistema para Análise de Ocorrências de Crimes Baseado em Aprendizado Lógico-Relacional

Alternative Title: SiAPP: An Information System for Crime Analytics Based on Logical Relational Learning

Vítor Lourenço
Paulo Mann
Instituto de Computação (IC/UFF) - Universidade Federal Fluminense
{vitorlourenco,paulomann}@id.uff.br

Aline Paes
Daniel de Oliveira
Instituto de Computação (IC/UFF) - Universidade Federal Fluminense
{alinepaes,danielcmo}@ic.uff.br

RESUMO

O crescente aumento da criminalidade em cidades brasileiras é um tema recorrente tanto nos veículos de comunicação como nas pautas das autoridades governamentais. Para combater efetivamente a criminalidade é necessário que recursos humanos e infraestrutura sejam cuidadosamente aplicados, de forma a não apenas punir quem cometeu o crime, mas preferencialmente prever e evitar que o mesmo aconteça. Dada a dificuldade de coletar um grande volume de informações oficiais relacionadas a crimes em todas as regiões de um município, uma tendência é que os próprios cidadãos atuem como fonte de dados, a partir de sistemas colaborativos baseados na *Web*. Entretanto, tal fonte de dados pode se tornar muito complexa e vasta, dificultando a análise manual de padrões de ocorrências de crimes, de forma a evitar que eles aconteçam. Com essa motivação, desenvolvemos nesse artigo um sistema denominado SiAPP (**Si**stema de **A**poio ao **P**oliciamento **P**reditivo), para apoiar a análise e predição de padrões relacionados a ocorrências de crimes, a partir de um método de aprendizado de máquina. O SIAPP tem como habilidades a coleta automática de informações a partir de dados colaborativos, a criação automática de regras lógicas a partir de tais informações e a visualização geográfica dos padrões descobertos. Resultados experimentais mostram que o SiAPP é uma abordagem promissora para o auxílio no combate ao crime.

Palavras-Chave

Programação Indutiva em Lógica (ILP). Predição de Crimes. Visualização de padrões de crimes.

ABSTRACT

The growing of criminality in Brazilian cities is a common theme addressed by media as well as by the legal authorities. To effectively reduce the criminality, people and infrastructure must be carefully involved to not only punish who had committed crimes, but also predict and prevent it. Since acquiring official data about crimes is far from trivial, citizens have become important data sources through Web-based collaborative systems. These systems provide a huge volume of data that has to be analyzed. How to analyze this volume of data and identify patterns in crimes is an important, yet open, issue. Thus, this work presents a system called SiAPP. Its main objective is to support the analysis and prediction of crime patterns using a machine learning algorithm. SiAPP automatically acquires data from collaborative sources, generate logical rules and visualizes the found patterns. Experimental analysis shows that SiAPP is a promising solution tool to assist crimes prevention.

SBSI 2016, May 17–20, 2016, Florianópolis, Santa Catarina, Brazil.

Categories and Subject Descriptors

I.2.6 [**Computing Methodologies**]: Learning – *induction, knowledge acquisition*. H.2.8 [**Information System**]: Database Applications – *data mining*.

Keywords

Inductive Logic Programming, Crime Analytics, Crime Pattern Visualization.

1. INTRODUÇÃO

Não há dúvidas que a criminalidade vem aumentando em diversas cidades do Brasil nos últimos anos e se tornou pauta prioritária na agenda dos governos, sejam eles estaduais ou municipais [1]. Entretanto, o combate ao crime não é uma tarefa trivial de ser desempenhada, principalmente em metrópoles complexas como Rio de Janeiro, São Paulo e Belo Horizonte. Cada uma dessas cidades contém centenas de bairros, com perfis de crimes diferentes, o que faz com que a organização das forças policiais para combater tais crimes seja complexa [1].

Para melhor exemplificar a situação, tomemos como exemplo o município de Niterói apresentado na Figura 1 (esse exemplo será usado consistentemente por todo o artigo). Niterói é um município do estado do Rio de Janeiro que conta com 52 bairros, uma população de 487.562 habitantes e uma área de 133,916 km^2 segundo os dados do IBGE[1]. Apesar de Niterói possuir o maior Índice de Desenvolvimento Humano (IDH)[2] do Rio de Janeiro a quantidade de crimes de diversos tipos tem aumentado a cada dia. Por exemplo, no período de 2011 a 2015, os assaltos a pedestres na Grande Niterói, que também abrange o município de São Gonçalo,

[1] http://cidades.ibge.gov.br/painel/painel.php?codmun=330330

[2] http://g1.globo.com/economia/idhm-2013/index.html

subiram 88,8% (de 6.573 para 12.420 casos) segundo dados do Instituto de Segurança Pública do Rio de Janeiro (ISP) [2]. Ainda, o número de roubos de aparelhos de celular no acumulado de 2015 aumentou em mais de 60% em relação ao ano de 2014, perfazendo um total de 1.386 roubos desses aparelhos. No mesmo ano, o índice de recuperação de veículos roubados ou furtados caiu em torno de 17,1%, quando comparado ao mesmo período de 2014 [2].

Assim, a área de segurança pública é um desafio atual em Niterói, assim como em outros grandes centros urbanos. O planejamento para combater tais crimes deve ser realizado contando com informações sobre tipo de crimes, ocorrências por região, horário *etc*. A partir da estruturação e da análise de informações contidas nesses dados, a cidade pode gerar conhecimento acerca de suas necessidades, sendo adaptativa e flexível às circunstancias hostis e elaborando planejamentos e ações contingenciais para corrigir e prevenir possíveis problemas urbanos [3].

Iniciativas já vêm sendo tomadas como, por exemplo, aumentar o efetivo de policiais e guardas municipais, e a criação do CISP (Centro Integrado de Segurança Pública)[3]. Entretanto, sem um planejamento que considere quais áreas devem ser cobertas por certo tipo de crime que acontece com mais frequência, que tipo de ações devem ser tomadas por localidade, entre outras, tais iniciativas podem ser difíceis de serem implementadas.

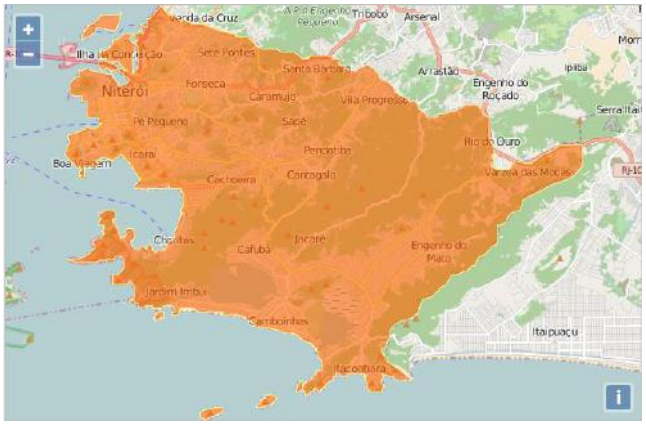

Figura 1 Mapa do município de Niterói segundo o IBGE

A ideia de poder antecipar os crimes para evitá-los (ou pelo menos diminuir seu impacto) se torna interessante e útil para a sociedade. Tal predição se torna possível a medida que sejam desenvolvidas tecnologias capazes de gerar modelos preditivos acurados. Além disso, um grande volume de dados sobre crimes se encontra disponível para análise, seja por vias oficiais como o ISP ou por aplicativos ou sítios *Web* que mapeiam crimes ocorridos como o "Onde fui roubado"[4], o "B.O. Coletivo"[5] e o "WikiCrimes Mobile"[6]. Infelizmente, tais abordagens apenas apresentam os crimes de forma quantitativa em mapas interativos, sem nenhuma análise sobre a inter-relação entre as ocorrências nas diversas áreas, localidades e tipos de crimes.

Assim, de forma a oferecer apoio às autoridades na prevenção de crimes, propomos nesse artigo o sistema de informação SiAPP (**Si**stema de **A**poio ao **P**oliciamento **P**reditivo). O objetivo do SiAPP é gerar modelos de predição que apontem para as autoridades as chances de algum delito ocorrer em determinada região e período do dia. Dessa forma, com o desenvolvimento do SiAPP, pretendemos disponibilizar para as autoridades uma nova possibilidade de análise de ocorrências de crimes.

O SiAPP é um sistema *Web* de código livre que pode ser disponibilizado em uma plataforma de nuvem, garantindo assim sua disponibilidade, e baseado em dados colaborativos, que são coletados por meio de formulários e de fontes externas como as anteriormente apresentadas. Dessa forma, toda informação é fornecida pela população, conseguindo, assim, prover mais uma forma de interação entre a sociedade e a esfera governamental. Todo o processamento do SiAPP é controlado por uma máquina de *workflow* que executa todas as atividades de pré-processamento, geração do modelo preditivo e visualização de padrões. Essa máquina também executa no ambiente de nuvem e é chamada SciCumulus [4]. Para gerar os modelos preditivos, o SiAPP se baseia em algoritmos de aprendizado de máquina, mais especificamente no Aprendizado Lógico-relacional por meio de Programação Indutiva em Lógica (*i.e.* ILP) [5]. Os dados coletados com a população são traduzidos para representação lógica para formar um conjunto de exemplos e por meio do sistema de ILP Aleph [6] os modelos preditivos são gerados baseados em três pontos específicos: tipo de crime, lugar e horário de casos ocorridos anteriormente.

A avaliação experimental do SiAPP foi realizada com o conjunto de dados obtidos do sítio *Web* "Onde fui roubado" relativo a cidade de Niterói. Os resultados obtidos mostram uma acurácia do modelo de mais de 80% enquanto que o percentual de ocorrências falso positivas se manteve menor que 18%. Apesar de inicial, tal resultado mostra que a abordagem é promissora.

Esse artigo se encontra organizado em 5 seções além desta introdução. A Seção 2 apresenta o referencial teórico. Na Seção 3 apresentamos a abordagem proposta. A avaliação experimental é exposta na Seção 4. A Seção 5 aborda os trabalhos relacionados e, por fim, a Seção 6 conclui esse artigo.

2. REFERENCIAL TEÓRICO

Nessa seção, revemos brevemente os fundamentos utilizados para a construção do SiAPP, a saber, o arcabouço para a execução do processo de tratamento dos dados e geração do modelo preditivo por meio de um *Workflow* e conceitos de aprendizado de máquina relacional utilizado para a descoberta de padrões criminais a partir de dados históricos.

2.1 *Workflows* Científicos

Nas últimas décadas, cada vez mais a ciência tem se beneficiado de simulações computacionais. Em grande parte dos casos, essas simulações são representadas por meio do encadeamento de múltiplas combinações de programas que podem consumir grandes quantidades de dados, e são distribuídos em ambientes computacionais heterogêneos. Nessas simulações, cada programa pode ser executado consumindo um grupo específico de parâmetros e dados. A saída de um programa é normalmente utilizada como entrada para outro programa no encadeamento [7].

Os *workflows* científicos [8] são uma alternativa atraente para representar os encadeamentos de programas ao invés de usarmos uma abordagem *ad-hoc* ou baseada em *scripts*. Os *workflows* científicos podem ser definidos como uma abstração para modelar

[3] http://www.niteroi.rj.gov.br/index.php?option=com_content&view=article&id=3355:2015-09-10-00-47-00

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

[5] http://www.bocoletivo.org/

[6] http://wikicrimes.org/

o fluxo de atividades e de dados em um experimento. Em *workflows* científicos, essas atividades são geralmente programas ou serviços que representam algoritmos e métodos computacionais sólidos [9]. Esses *workflows* são controlados e executados por complexos mecanismos chamados de Sistemas de Gerência de *Workflows* Científicos (SGWfC), que visam apoiar a configuração e execução dos *workflows*.

Devido à complexidade inerente aos modelos que fazem parte dos experimentos científicos, estes devem ser executados em ambientes computacionais distribuídos, de grande capacidade de processamento e muitas vezes heterogêneos aliados à aplicação de técnicas de paralelismo. Como exemplos destes ambientes podemos citar os *clusters* [10], as grades computacionais [11], e mais recentemente as nuvens de computadores [12].

Os *workflows* são utilizados no contexto desse artigo para modelar o processo de aquisição e tratamento de dados (pré-processamento) e geração do modelo preditivo de forma automática e em paralelo na nuvem, de forma a diminuir o tempo de execução. Abordagens semelhantes que visavam modelar experimentos de mineração de dados como *workflows* científicos já foram propostas com resultados de sucesso [13].

2.2 Programação Indutiva em Lógica

Os algoritmos de aprendizado de máquina têm como meta induzir de forma automática padrões a partir de um conjunto de dados [14], também chamados de exemplos. Usualmente, tais algoritmos assumem que os exemplos são distribuídos de forma idêntica e independente. Dessa forma, o conjunto de exemplos costuma ser representado por uma matriz, onde as linhas são os exemplos e as colunas são os atributos (*i.e. features*) relevantes para todos os exemplos do domínio. Assim, cada célula da matriz tem um valor para um atributo de um exemplo. Por isso, dizemos que os exemplos estão no formato de atributo-valor.

Entretanto, dados do mundo real, como as ocorrências de crimes, são heterogêneos, no sentido que um exemplo pode possuir um atributo que não faz sentido ser representado em outro exemplo, bem como os exemplos podem diferir na quantidade de atributos. Adicionalmente, dados do mundo real são comumente relacionais, no sentido de que um exemplo pode estar diretamente associado a outro exemplo e assim eles não podem ser tratados de forma independente.

O aprendizado de máquina relacional é um campo de aprendizado de máquina que tem como objetivo descobrir padrões a partir de exemplos relacionais, que não necessariamente precisam estar representados no formato de atributo-valor. Uma abordagem já bastante estabelecida para a execução de aprendizado relacional é a Programação Indutiva em Lógica (ILP, do inglês, *Inductive Logic Programming*) [5]. A entrada para um sistema de ILP é composta por um conjunto de exemplos (E), um conhecimento preliminar (BK) e o bias da linguagem (BL) e do aprendizado (BA). O conjunto de exemplos é usualmente dividido em exemplos positivos e negativos, que são representados por fatos em lógica de primeira-ordem [15], no formato $p(c_1,\ldots,c_n)$, onde p é um predicado e $c_1,\ldots,c_n$ são constantes. O BK é composto por fatos e/ou regras lógicas, que são usados para representar os atributos e propriedades dos objetos do domínio, bem como possíveis relacionamentos entre tais objetos. A saída de um sistema em ILP é um programa lógico, que (idealmente) cubra os exemplos positivos, mas não cubra os exemplos negativos. O *bias* da linguagem define basicamente quais os literais que poderão aparecer no corpo da regra, a tipagem para suas variáveis e/ou constantes. O *bias* do aprendizado define qual a função de avaliação utilizada no processo de otimização de uma regra, se é permitido que exemplos positivos sejam cobertos, entre outros.

O Aleph é um sistema de ILP comumente usado em diversos tipos de domínios, tais como bioinformática e químio-infomática, jogos, e outros. O sistema Aleph tem a capacidade de aprender padrões tendo como base a construção da *cláusula mais específica* a partir de um exemplo positivo [16]. Assim, a cada iteração, a cláusula mais específica é construída a partir de um exemplo positivo ainda não coberto, do BK e do bias da linguagem. A seguir, uma cláusula Horn [17] é formada a partir dos literais presentes na cláusula mais específica de forma a atender a função de avaliação definida no *bias* do aprendizado. Ao usar a função de avaliação *default*, esse passo se resume a selecionar literais que cubram tantos exemplos positivos quanto possível, ao mesmo tempo em que exemplos negativos deixam de ser cobertos. Quando não é mais possível escolher literais bons o suficiente de acordo com a função de avaliação, o sistema parte para a criação de uma nova cláusula, escolhendo outro exemplo positivo ainda não coberto e construindo uma nova cláusula mais específica. Tal aprendizado foi utilizada no SiAPP para produzir os modelos preditivos de ocorrência de crimes. O processo de geração do modelo e a arquitetura do sistema são apresentados a seguir.

3. SiAPP: UM SISTEMA PARA PREVISÃO DE CRIMES

3.1 Arquitetura

O sistema de informação SiAPP foi desenvolvido na linguagem Python e Prolog [18]. Para o desenvolvimento das interfaces, *templates* e telas do SiAPP foi utilizado o arcabouço *Bootstrap*. Como o SiAPP é uma aplicação *Web* que pode ser acessada por um navegador *Web* comum, o mesmo deve ser hospedado em um servidor de aplicação. O servidor escolhido foi o Apache Tomcat e o mesmo foi instalado e configurado em um servidor no ambiente de nuvem da Amazon[7]. A arquitetura do SiAPP é baseada em 4 componentes, conforme apresentado na Figura 2: o SGWfC SciCumulus, o componente de visualização, o sistema de ILP Aleph e a base de conhecimento.

O SGWfC é o componente responsável por executar o *workflow* SiAPP-Wf que representa o processo de tratamento de dados e geração do modelo preditivo. Toda vez que o usuário acessa o SiAPP e requisita que o modelo preditivo seja gerado novamente ou atualizado, o SGWfC SciCumulus é invocado. O SciCumulus então instancia um número de máquinas virtuais na nuvem e executa as etapas de aquisição dos dados, conversão, normalização e geração do modelo preditivo. Cada etapa está associada a invocação de um programa conforme detalhado na Subseção 3.2. A grande vantagem de se utilizar um SGWfC acoplado ao sistema é que se o processo evoluir ou for retificado, apenas se faz necessária a mudança na especificação do *workflow*, sem necessitar modificar os outros componentes do sistema. Uma das atividades do *workflow* é relativa a execução do algoritmo de aprendizado de máquina relacional. Essa atividade invoca o sistema Aleph para que o mesmo gere o modelo preditivo desejado.

O componente de visualização consulta os dados de ocorrências e as regras do modelo preditivo armazenadas na base de conhecimento e expõe os dados em um mapa para visualização por parte do usuário. A base de conhecimento é o componente

[7] O sítio para o sistema será disponibilizado em www.ic.uff.br/~siapp.

responsável por armazenar as ocorrências de crimes e o modelo preditivo gerado, por meio das regras geradas pelo sistema Aleph.

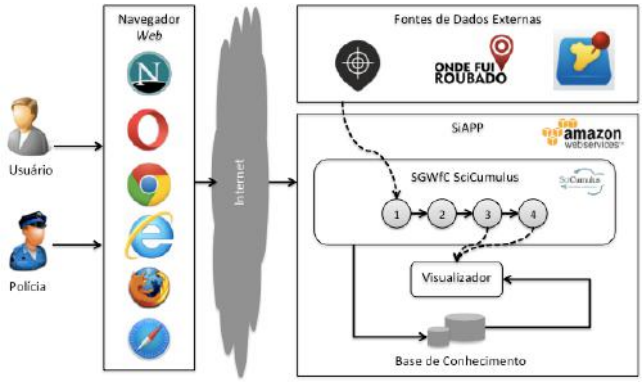

Figura 2. Arquitetura do Sistema.

3.2 *SiAPP-Wf*

O *SiAPP-Wf* (Figura 3) é o *workflow* executado pelo SiAPP e representa o processo de pré-processamento dos dados e de aprendizado e geração do modelo preditivo. O SiAPP-Wf é composto por 4 atividades, a saber: "Aquisição dos Dados", "Conversão dos Dados", "Normalização dos Dados", "Aprendizado do Modelo Preditivo". A seguir detalhamos cada uma dessas atividades.

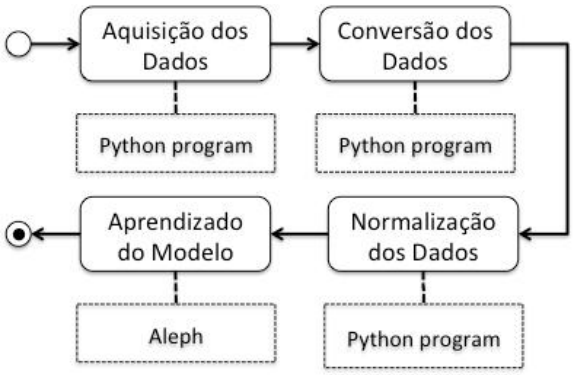

Figura 3 O *Workflow* SiAPP-Wf

3.2.1 *Aquisição dos Dados*

A aquisição dos dados consiste na obtenção de dados brutos oriundos de uma fonte de dados externa ou de serviços *Web*. Sítios e aplicativos que registram ocorrências de crimes em regiões metropolitanas servem como fonte de dados para o SiAPP. Essa atividade é implementada por um *crawler* que deve ser customizado para cada uma das fontes de dados externas.

No experimento a ser discutido na Seção 4 desse artigo, os dados relativos às ocorrências de crimes foram obtidos do sítio *Web* colaborativo "Onde Fui Roubado". Os dados relativos às localidades foram obtidos por meio do sítio, também colaborativo, "*Open Street Maps*" (http://openstreetmaps.org). Inicialmente, foram obtidas 781 ocorrências reportadas no sítio "Onde Fui Roubado" referentes à cidade de Niterói. Em cada ocorrência foram capturados dados relativos à sua geolocalização da ocorrência, ao tipo de crime, à lista de objetos roubados ou furtados, a data e ao horário em que a ocorrência aconteceu. Além disso, ainda é capturada uma descrição associada. A partir das coordenadas identificadas, foram extraídos 281 pontos importantes na cidade de Niterói. Nesses dados, foram obtidos, além da geolocalização, o nome da localidade (*e.g.* Universidade Federal Fluminense ou Colégio Abel) e o tipo da localidade (*e.g.* escola, padaria, universidade, *etc*.).

3.2.2 *Conversão dos Dados*

O processo de conversão se resume a passagem dos dados para tabelas no formato separado por vírgula (*i.e. *.csv*), para uma melhor manipulação dos mesmos pelos programas de conversão. Dessa forma, são criadas duas tabelas correspondentes às ocorrências e às localidades.

Outro ponto importante no processo de conversão é a associação das localidades às ocorrências, gerando o entorno de uma localidade. Esta associação é realizada por meio da fórmula de *haversine* [19], em que é calculada a distância entre a ocorrência e a localidade. Pelas características de proximidade entre as localidades, foi adotado um raio máximo de 500 metros para considerar-se entorno de uma ocorrência.

Após a obtenção dos dados de entorno, é gerada uma única tabela, também no formato *.csv*, contendo as informações existentes na tabela de ocorrências, acrescida de uma coluna correspondente à propriedade "tipo da localidade", presente em entorno.

3.2.3 *Normalização dos Dados*

A etapa de normalização consiste na discretização dos dados, transformando-os em valores não contínuos [20]. Essa etapa é fundamental para o processo de aprendizagem em que padrões serão verificados mais facilmente [20]. Dessa forma, dados contínuos como "latitude", "longitude", "dia" e "hora" foram transformados em intervalos rotulados. A lista de objetos foi transformada em objetos individuais rotulados segundo a sua existência (ou não) nas ocorrências. O entorno e o tipo de ocorrência foram mantidos na representação original, uma vez que possuem valores já discretizados.

A discretização realizada é apresentada na Tabela 1. De acordo com o apresentado na Tabela 1, "Bairro" simboliza os 52 bairros da cidade de Niterói; "Objeto" denota, de forma individual, os 19 tipos de objetos existentes no estudo; "Dia comercial" remete aos dias úteis semanais, ou seja, de segunda-feira a sexta-feira, da mesma forma, sábado e domingo são representados por "Fim de semana"; "Madrugada" refere-se ao período de tempo de 00:00h às 05:59h, "Manhã" corresponde a 06:00h até 11:59h, "Tarde" de 12:00h às 17:59h e, por fim, "Noite" de 18:00h às 23:59h.

Tabela 1 Discretização dos valores no SiAPP

Dados não discretizados	**Dados discretizados**
Latitude/Longitude	Nome do Bairro
Lista de Objetos	Nome do Objeto
Dia	Dia comercial / Fim de semana
Hora	Madrugada / Manhã / Tarde / Noite

Os dados discretizados são a saída da atividade "Normalização dos Dados" do SiAPP-Wf. A partir desse momento, o componente de visualização do SiAPP já pode plotar os pontos onde crimes ocorreram nos mapas. Entretanto, até o presente momento, nenhum modelo preditivo foi gerado pelo *workflow*. Para que os modelos possam ser gerados, os dados normalizados dão origem a uma nova tabela na base de conhecimento, e um fragmento da mesma encontra-se exemplificado na Tabela 2. Uma vez que a

Tabela 2 encontra-se preenchida na base de conhecimento, a atividades 4 do *workflow* pode ser executada.

Tabela 2 Exemplo de tabela de dados de ocorrências no SiAPP

Tipo de Ocorrência	**Bairro**	**Dia**	**Hora**	**Entorno**
furto	ingá	dia_ comercial	tarde	bar
documentos	**celular**	**...**	**tv**	**moveis**
TRUE	TRUE	...	FALSE	FALSE

3.2.4 Geração da Base de Conhecimento e Aprendizado do Modelo Preditivo

Na atividade de aprendizado do modelo preditivo, o sistema Aleph é invocado pela máquina de *workflow* SciCumulus para identificar e construir padrões a respeito dos crimes ocorridos, seguindo os dados descobertos nas atividades anteriores do *workflow*. Para tanto, torna-se necessário transformar os dados normalizados para a representação em lógica de primeira-ordem. Assim, todos os literais (um literal é uma fórmula atômica ou a negação da mesma) presentes no conjunto de exemplos e no conhecimento preliminar são criados automaticamente a partir dos dados discretizados produzidos na atividade anterior.

Os exemplos positivos são construídos a partir do identificador associado à ocorrência e seu tipo. Assim, temos, por exemplo, `occurrence(30, roubo)` como um exemplo positivo, onde *30* é o identificador da ocorrência e *roubo* é o tipo de ocorrência. Cada tipo de ocorrência distinto em uma ocorrência dará origem a um exemplo positivo. Para a criação automática dos exemplos negativos, assumimos a hipótese do mundo fechado da lógica [21], onde para cada ocorrência registrada (exemplo positivo), os tipos de ocorrências que não aconteceram tornam-se exemplos negativos.

Finalmente, todos os demais dados constantes na base de dados são automaticamente convertidos para literais a serem inseridos no conhecimento preliminar. Por exemplo, supondo que a ocorrência *30* apresentada anteriormente tenha ocorrido no bairro *Ingá*, e os objetos roubados na ocorrência sejam *celular* e *relógio*, em uma *segunda-feira*, na parte da *manhã*, próximo a uma *universidade* e a um *restaurante*, teremos os seguintes literais na base de conhecimento:

```
bairro(30,inga)
objeto(30,celular)
objeto(30,relogio)
dia(30,comercial)
periodo(30,manha)
entorno(30,universidade)
entorno(30,restaurante)
```

Após a criação da base de dados, o sistema de ILP Aleph é executado e produz como resposta um programa lógico. Cada regra desse programa lógico é uma cláusula definida, ou seja, uma cláusula com apenas um literal positivo. Tal literal positivo estará na cabeça da cláusula e será uma generalização do exemplo positivo, isto é, a identificação da ocorrência se torna uma variável de ligação e o tipo de ocorrência é uma constante. O tipo de ocorrência também poderia ser outra variável, mas definimos como constante no *bias* da linguagem, de forma a caracterizar os tipos de ocorrências. O corpo da cláusula é uma conjunção de literais também generalizados de acordo com o *bias* da linguagem. Nesse caso, a identificação sempre é substituída por uma variável, porém os demais termos podem ser variáveis ou constantes. Isso é definido pelo processo de aprendizado via Aleph, durante a construção da regra. Assim, a regra a seguir apresenta um exemplo de um programa lógico aprendido pelo sistema Aleph, onde verificamos que se a pessoa se encontra no bairro de Icaraí, portando um cartão de crédito e próxima a uma clínica, existe a chance de a mesma sofrer um roubo.

```
ocorrencia(A,roubo) :- bairro(A,icarai),
objeto_roubado(A,cartao_de_credito),
entorno(A,clinica).
```

3.3 Visualização dos Dados

O componente de visualização é o responsável pela exposição dos dados de forma visual com a finalidade de facilitar a compreensão das

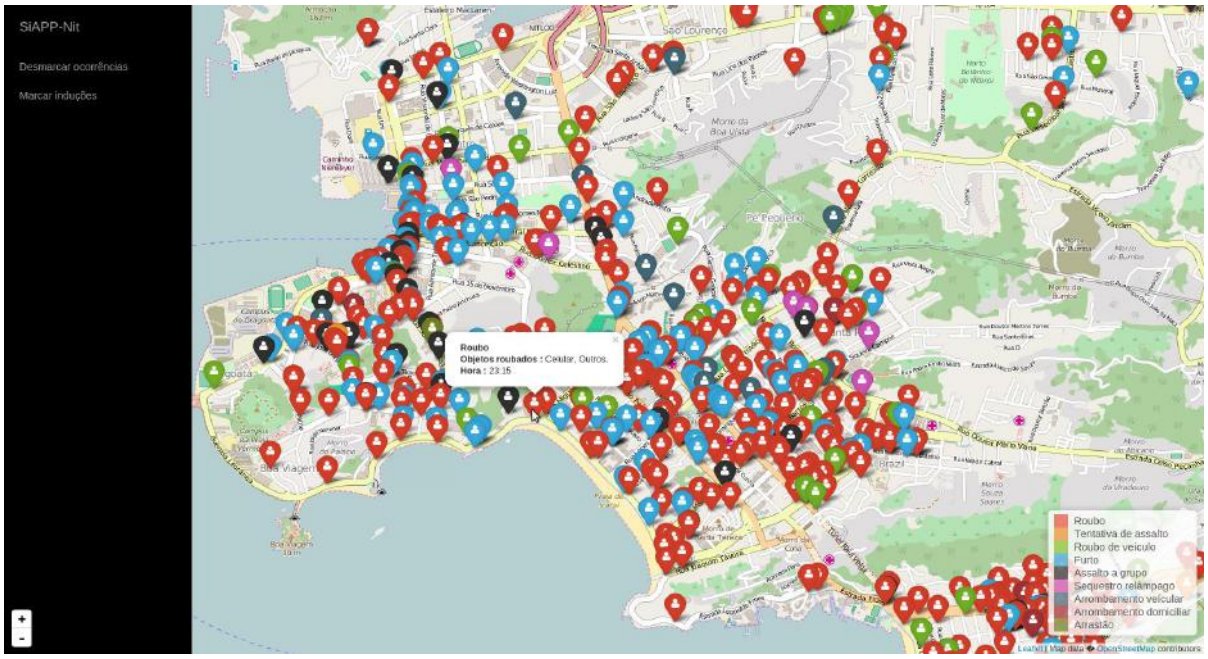

Figura 4 Visualização das ocorrências no SiAPP

ocorrências de crimes e do modelo preditivo gerado. Esse componente é capaz de apresentar dados de duas formas, como as representadas na Figura 4 e na Figura 5.

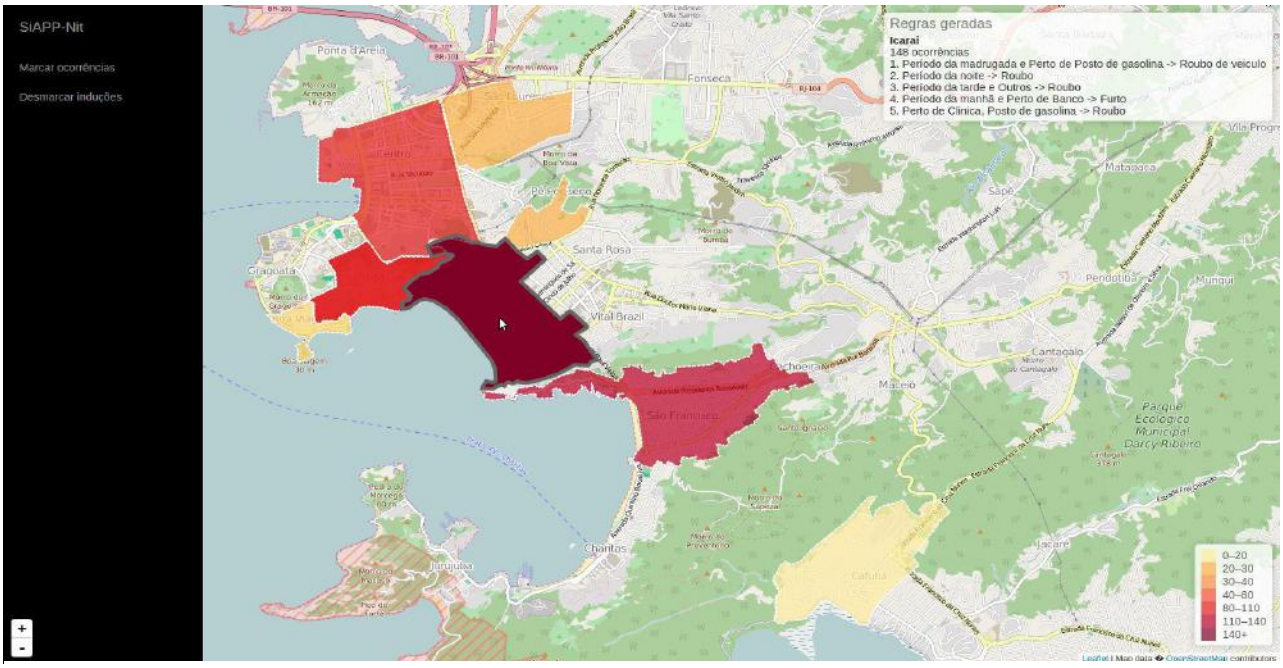

Figura 5 Visualização das predições e regras por bairro no SiAPP

Na Figura 4, apresentamos a visualização dos pontos onde uma ocorrência de crime aconteceu. Nessa visualização, a localidade de onde a ocorrência se deu e detalhes como o tipo de crime, objetos roubados ou furtados e hora são apresentados quando selecionado um dos pontos no mapa. Esse tipo de visualização já é apresentada por abordagens como o "Onde fui roubado" e o "B.O. coletivo", porém tal visualização não permite aos usuários relacionar as ocorrências da cidade ou de uma área de modo a identificar padrões mais precisos e úteis.

Assim, na Figura 5 apresentamos a visualização das regras produzidas pelo modelo preditivo no mesmo mapa. Na Figura 5 os bairros onde regras foram geradas são evidenciados e as regras associadas são apresentadas ao usuário. Por exemplo, na Figura 5 podemos verificar que para o bairro do Ingá em Niterói uma regra foi gerada indicando que se o usuário está a noite e perto de uma escola no bairro, existe a chance de ser roubado. Tal tipo de conhecimento é difícil de ser inferido apenas com visualização apresentada na Figura 4. A seguir apresentamos a validação do SiAPP com o conjunto de dados de ocorrências de crimes na cidade de Niterói.

4. AVALIAÇÃO EXPERIMENTAL – DESCOBERTA DE PADRÕES DE CRIMES NA CIDADE DE NITERÓI/RJ

O SiAPP apresenta-se como uma solução escalável para qualquer cidade que possua um histórico sobre atividades criminais disponível. É importante ressaltar que tal histórico de dados pode ser originário de fontes heterogêneas como órgãos públicos e aplicativos proprietários. Para avaliar o sistema de informação proposto, suas particularidades e vantagens, nesse trabalho, foi escolhida a cidade de Niterói como estudo de caso. Assim, o sistema instanciado foi nomeado de SiAPP-Nit. Conforme mencionado anteriormente, para essa avaliação experimental os dados relativos às ocorrências foram obtidos do sítio colaborativo "Onde Fui Roubado" e os dados relativos às localidades do sítio, também colaborativo, "Open Street Maps. Apesar do sítio "Onde fui roubado" possuir ocorrências de diversos municípios, selecionamos apenas os da cidade de Niterói. No total, 781 ocorrências foram extraídas, considerando 9 tipos de ocorrências, a saber: roubo, furto, roubo de veículo, assalto a grupo, sequestro relâmpago, arrombamento veicular, arrombamento domiciliar, arrastão e tentativa de assalto.

Para avaliar o SiAPP-Nit adotamos a metodologia experimental que se baseia nas métricas de avaliação do modelo produzido. O algoritmo de aprendizado utilizou-se da técnica de validação cruzada com *k-folds* [20]. A utilização da validação com *k-folds* evita que o modelo aprendido seja especializado apenas para um subconjunto das ocorrências de crimes. Essa técnica consiste em dividir os dados em k subconjuntos mutuamente exclusivos, reservando um conjunto para validar o modelo, e todos os outros $(k - 1)$ para o treinamento. Como usual, foi adotado $k = 10$, ou seja, 10 *folds* e foi computada a média dos resultados preditivos das dez execuções, a fim de produzir uma única estimativa. A função de avaliação escolhida foi a m-estimate [22], que é apropriada para lidar com dados com ruídos.

Os resultados experimentais foram analisados a partir de duas métricas, uma qualitativa e outra quantitativa. Para a métrica qualitativa foram analisadas as regras geradas pelo modelo preditivo e comparadas com estatísticas oficiais do ISP e com notícias veiculadas na imprensa relativas a ocorrência de crimes na cidade de Niterói. Desta forma, foram selecionadas como exemplo algumas regras que apontam acontecimentos expostos em [2]. Por exemplo, de acordo com **a regra 1**, a incidência de furtos é alta nas regiões do entorno do *campus* da Universidade Federal Fluminense, situada entre os bairros do Ingá, do Centro e de Boa Viagem. Nessa mesma região há mais dois centros universitários e diversas escolas. A **regra 3** indica que houve um aumento na ocorrência de furtos no entorno de escolas, em dias

comerciais. Tais regras, descobertas automaticamente pelo SiAPP-Nit, são corroboradas por [2], em que roubos em regiões de escola também tiveram seus números aumentados. Outro exemplo é a **regra 5**, que indica incidência de roubos de carteira na região do centro a noite. Essa região é composta por diversos bares e de fato o crime mais comum é o roubo de carteiras e mochilas. As estatísticas presentes em [2] corroboram os resultados do modelo preditivo.

```
1. dia(dia_comercial), objeto(celular),
entorno(universidade) → ocorrencia(furto)

2.bairro(inga), periodo(tarde) →
ocorrencia(roubo)

3. bairro(inga), dia(dia_comercial),
entorno(escola) → ocorrencia(roubo)

4.bairro(boa_viagem) → ocorrencia(roubo)

5. bairro(centro), objeto(carteira),
periodo(noite) → ocorrencia(roubo)
```

Além dos crimes citados nas regras de 1 a 5, outros tipos de crimes comuns também foram observados, como o conhecido como "saidinha de banco" e os roubos em restaurantes. As regras de 6 a 8 são associadas a esses tipos de crimes.

```
6.hora(tarde), objeto(celular),
entorno(banco) →      ocorrencia(roubo)

7.bairro(icarai),periodo(manha),
entorno(banco) →      ocorrencia(furto)

8.objeto(bolsa_ou_mochila), entorno(restaurante),
hora(madrugada) → ocorrencia(assalto_a_grupo)
```

Para a análise quantitativa foram considerados os resultados das matizes de confusão (a matriz de confusão de um modelo oferece uma medida efetiva do modelo de classificação, ao mostrar o número das classificações corretas versus as previsões para cada classe) dos 10 *folds* e delas calculados a acurácia (porcentagem de amostras de ocorrências de crimes positivas e negativas classificadas corretamente sobre a soma de amostras positivas e negativas), a precisão (porcentagem de amostras positivas classificadas corretamente sobre o total de amostras classificadas como positivas); a sensibilidade (porcentagem de ocorrências de crime positivas classificadas corretamente sobre o total de ocorrências positivas), a medida-F (também chamada de *F-score*, é uma média ponderada de precisão e sensibilidade) e a taxa de falsos negativos (*miss rate*). Dessa forma, a média e o desvio padrão dos resultados alcançados são expostos na Figura 6 e na Tabela 3.

O gráfico apresentado na Figura 6 retrata os dados presentes na Tabela 3. Este, aborda uma visualização modificada do tradicional diagrama de caixas (*box-plot*), onde os quartis, inferior (primeiro quartil) e superior (terceiro quartil), representam os desvios padrão para menos e para mais, respectivamente. A dispersão dos dados com seus valores mínimos e máximos representados pelas estimativas do *whisker*. E, finalmente, a representação dos dados compreendida dentro das caixas e a média como o valor médio destas caixas.

De acordo com os resultados da Tabela 3 e da Figura 6 podemos perceber que as regras aprendidas atingiram uma acurácia média preditiva de 83,54%, 72,07% de precisão e 82,65% de sensibilidade. A precisão e sensibilidade têm como foco apontar os exemplos positivos classificados corretamente, o que no caso do SiAPP é essencial, pois estes representam as possíveis ocorrências de crimes. Esse resultado é corroborado pelo valor alcançado pela medida-F (77,68%), que pondera precisão e sensibilidade. Em relação a taxa de falsos negativos (17,35%), o valor alcançado não foi tão reduzido quanto o desejado, o que pode ter sido influenciado pela quantidade de exemplos negativos gerados artificialmente, usando a hipótese do mundo fechado. Apesar de mais testes ainda se fazerem necessários com um conjunto de dados maior, os resultados quantitativos e qualitativos obtidos são promissores e mostram a utilidade e potencial do sistema de informação proposto.

Tabela 3 Resultados da avaliação quantitativa do SiAPP-Nit

Métricas	Média	Desvio Padrão
Acurácia	**83,54%**	**2,30%**
Sensibilidade	**82,65%**	**4,45%**
Precisão	**72,07%**	**3,07%**
Medida-F	**77,68%**	**3,42%**
Taxa de Falsos Negativos	**17,35%**	**4,45%**

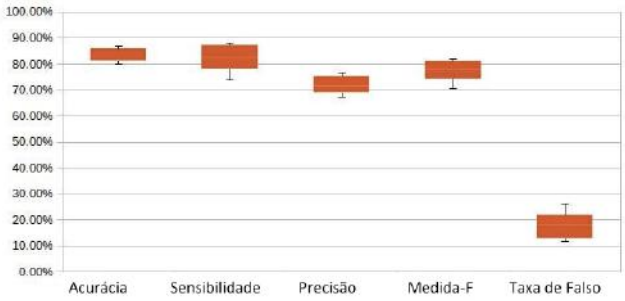

Figura 6 Visualização baseada em *box-plot* para média e desvio padrão dos 10 *folds*

5. TRABALHOS RELACIONADOS

Como o tema "segurança pública" é um desafio global, algumas propostas com objetivos semelhantes podem ser encontradas na literatura e na área comercial. Talvez a abordagem mais proeminente que trata do tema de previsão de crimes seja o sistema PredPol (http://www.predpol.com/). O PredPol é um sistema comercial utilizado pela polícia de Atlanta nos EUA, que utiliza o histórico de crimes de uma região para prever a incidência de novos crimes na cidade. Apesar de bastante popular, tal sistema não divulga como a previsão é realizada, porém, segundo os desenvolvedores, o sistema foi capaz de gerar previsões que reduziram em até 32% o número de assaltos naquela cidade.

Em termos de abordagens comerciais ou governamentais, existem outras iniciativas mais simples e oriundas de departamentos de polícia ao redor do planeta, como os mapas de crimes fornecidos pela polícia do Reino Unido (https://www.police.uk) e pela polícia de Nova Iorque (https://maps.nyc.gov/crime/). Entretanto, essas abordagens apenas exibem graficamente as ocorrências de crimes no mapa e não preveem a ocorrência dos mesmos.

Na área acadêmica, podemos citar o trabalho de Zhang *et al.* [23] que apresenta um sistema para previsão de atividades criminais baseado na teoria dos jogos por meio de um modelo baseado em redes bayesianas dinâmicas. Outra abordagem semelhante é proposta por Nath [24]. Essa abordagem é baseada na detecção de padrões de crimes utilizando um sistema baseado em diferentes técnicas de agrupamento. Apesar de apresentar bons resultados, nas abordagens de Zhang *et al.* e Nath os padrões que dependem do relacionamento de crimes não são detectados, uma vez que técnicas de agrupamento modelo de redes Bayesianas não são algoritmos de mineração de dados relacionais.

6. CONCLUSÕES

Devido ao aumento na ocorrência de crimes nas grandes cidades brasileiras nos últimos anos, a identificação de padrões de ocorrência de crimes em determinadas regiões pode ajudar na previsão dos mesmos e nos planos de segurança dos governos. Apesar de ser uma prioridade para muitos governos, essa tarefa está longe de ser trivial. Analisar a ocorrência de crimes e a inter-relação entre as ocorrências nas diversas áreas, localidades e tipos de crimes é uma tarefa árdua.

Dessa forma, para auxiliar as autoridades na prevenção de crimes, propomos nesse artigo o sistema de informação SiAPP (**Si**stema de **A**poio ao **P**oliciamento **P**reditivo). O objetivo do SiAPP é gerar modelos de predição utilizando algoritmos de aprendizado de máquina lógico-relacional que apontem para as autoridades as chances de algum delito ocorrer em determinada região e período do dia.

Resultados experimentais mostraram que as predições produzidas pelo SiAPP para a região de Niterói tiveram uma acurácia média maior que 83% e são coerentes com estatísticas providas pelo Instituto de Segurança Pública (ISP) e com as notícias e estatísticas veiculadas pela imprensa.

Apesar dos resultados serem promissores, como trabalhos futuros, pretendemos acrescentar na atividade de "Aprendizado do Modelo Preditivo" outros métodos de indução de modelos, incluindo principalmente algoritmos probabilísticos que poderiam lidar melhor com ruídos e observações parciais do que o sistema Aleph.

7. AGRADECIMENTOS

Os autores gostariam de agradecer ao CNPq, CAPES e à FAPERJ pelo financiamento parcial deste trabalho.

8. REFERÊNCIAS

[1] Felipe, "Tendências criminais sul-americanas em perspectiva comparada," *Revista Brasileira de Segurança Pública*.

[2] ISP, *Instituto de Segurança Pública do Rio de Janeiro, http://www.isp.rj.gov.br/*. .

[3] M. Marathe, "Resilient Cities and Urban Analytics: The Role of Big Data and High Performance Pervasive Computing," in *Proceedings of the 2Nd IKDD Conference on Data Sciences*, New York, NY, USA, 2015, pp. 4:1–4:1.

[4] D. 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*, Washington, DC, USA, 2010, pp. 378–385.

[5] S. Muggleton and ILP '96, Eds., *Inductive logic programming: 6th International Workshop, ILP-96, Stockholm, Sweden, August 1996 : selected papers*. New York: Springer, 1997.

[6] E. Alpaydin, *Introduction to machine learning*. Cambridge Mass.: MIT Press, 2004.

[7] M. C. Cavalcanti, R. Targino, F. Baião, S. C. Rössle, P. M. Bisch, P. F. Pires, M. L. M. Campos, and M. Mattoso, "Managing structural genomic workflows using web services," *Data & Knowledge Engineering*, vol. 53, no. 1, pp. 45–74, 2005.

[8] E. Deelman, D. Gannon, M. Shields, and I. Taylor, "Workflows and e-Science: An overview of workflow system features and capabilities," *Future Generation Computer Systems*, vol. 25, no. 5, pp. 528–540, 2009.

[9] A. Barker and J. van Hemert, "Scientific Workflow: A Survey and Research Directions," in *Parallel Processing and Applied Mathematics*, 2008, pp. 746–753.

[10] M. Dantas, "Clusters Computacionais," in *Computação Distribuída de Alto Desempenho: Redes, Clusters e Grids Computacionais*, 1st ed., .Rio de Janeiro: Axcel Books, 2005, pp. 145–180.

[11] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, 2004.

[12] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, "A break in the clouds: towards a cloud definition," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50–55, 2009.

[13] D. Oliveira, F. Baião, and M. Mattoso, "MiningFlow: Adding Semantics to Text Mining Workflows," in *First Poster Session of the Brazilian Symposium on Databases*, João Pessoa, PB - Brazil, 2007, pp. 15–18.

[14] K. Murphy, *Machine Learning: A Probabilistic Perspective*, 1 edition. Cambridge, MA: The MIT Press, 2012.

[15] A. Margaris, *First Order Mathematical Logic*. New York: Dover Publications, 1990.

[16] S. Muggleton, "Inverse entailment and progol," *NGCO*, vol. 13, no. 3–4, pp. 245–286, Dec. 1995.

[17] A. Horn, "On sentences which are true of direct unions of algebras," *Journal of Symbolic Logic*, vol. 16, no. 01, pp. 14–21, Mar. 1951.

[18] I. Bratko, *Prolog programming for artificial intelligence*. Harlow, England; New York: Addison Wesley, 2001.

[19] D. E. Crowley, R. R. Murphy, A. McNamara, T. D. McLaughlin, and B. A. Duncan, "AR Browser for Points of Interest in Disaster Response in UAV Imagery," in *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, New York, NY, USA, 2014, pp. 2173–2178.

[20] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques, Third Edition*, 3rd ed. Morgan Kaufmann, 2011.

[21] J. Minker, "On indefinite databases and the closed world assumption," in *6th Conference on Automated Deduction*, D. W. Loveland, Ed. Springer 1982, pp. 292–308.

[22] S. Džeroski, B. Cestnik, and I. Petrovski, "Using the M-estimate in Rule Induction," *J. Comput. Inf. Technol.*, vol. 1, no. 1, pp. 37–46, Mar. 1993.

[23] C. Zhang, M. Jain, R. Goyal, A. Sinha, and M. Tambe, "Learning, Predicting and Planning Against Crime: Demonstration Based on Real Urban Crime Data (Demonstration)," in *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, Richland, SC, 2015, pp. 1911–1912.

[24] S. V. Nath, "Crime Pattern Detection Using Data Mining," in *2006 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops*, 2006, pp. 41–44.