NATAL, 2018

Realização: SBC

Organização: UFRN

anais 2018

XXXVIII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO

1º WBCI – WORKSHOP BRASILEIRO DE CIDADES INTELIGENTES

CENTRO DE CONVENÇÕES | NATAL•RN | 22 A 26 DE JULHO DE 2018

#COMPUTAÇÃOESUSTENTABILIDADE

CSBC 2018

anais 2018

XXXVIII CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO
CENTRO DE CONVENÇÕES | NATAL•RN | 22 A 26 DE JULHO DE 2018
#COMPUTAÇÃOESUSTENTABILIDADE

Coordenador Geral

Francisco Dantas de Medeiros Neto (UERN)

Comissão Organizadora

Bartira Paraguaçu Falcão Dantas Rocha (UERN)
Camila Araújo Sena (UERN)
Everton Ranielly de Sousa Cavalcante (UFRN)
Felipe Torres Leite (UFERSA)
Ilana Albuquerque (UERN)
Isaac de Lima Oliveira Filho (UERN)
Priscila Nogueira Krüger (UERN)

Realização

Sociedade Brasileira de Computação

Organização

Universidade do Estado do Rio Grande do Norte

CSBC 2018
XXXVIII Congresso da
Sociedade Brasileira de Computação

Apresentação

Estes anais registram os trabalhos apresentados durante o XXXVIII Congresso da Sociedade Brasileira de Computação (CSBC 2018), realizado em Natal-RN, de 22 a 26 de julho 2018. O evento teve como tema central a Computação e Sustentabilidade, pois se compreende que o avanço da computação e as questões ambientais devem caminhar lado-a-lado, tendo em vista que as técnicas computacionais necessitam ser usadas para possibilitar o desenvolvimento sustentável, e, desse modo, equilibrar as necessidades ambientais, econômicas e sociais.

Organizar o maior evento acadêmico de Computação da América Latina foi um privilégio e um desafio. Foi enriquecedor promover e incentivar a troca de experiências entre estudantes, professores, profissionais, pesquisadores e entusiastas da área de Computação e Informática de todo o Brasil. Ao mesmo foi desafiador termos que lidar, principalmente, com às dificuldades impostas pelo momento de crise que o nosso Brasil vem enfrentando. Uma crise que afeta diretamente nossas pesquisas e, consequentemente, o desenvolvimento e inovação do nosso amado Brasil.

Por meio de seus 25 eventos, o CSBC 2018 apresentou mais de 300 trabalhos, várias palestras e mesas-redondas. O Congresso ainda abrigou diversas reuniões, que incluem a reunião do Fórum de Pós-Graduação, a reunião do CNPq/CAPES, a reunião dos Secretários Regionais SBC, a reunião das Comissões Especiais e a reunião do Fórum IFIP/SBC.

O sucesso do CSBC 2018 só foi possível devido à dedicação e entusiasmo de muitas pessoas. Gostaríamos de agradecer aos coordenadores dos 25 eventos e aos autores pelo envio de seus trabalhos. Além disso, gostaríamos de expressar nossa gratidão ao Comitê Organizador, por sua grande ajuda em dar forma ao evento; e, em especial, à equipe da Sociedade Brasileira de Computação (SBC), por todo apoio.

Por fim, reconhecemos a importância do apoio financeiro da CAPES, do CNPq, do CGI.br, do Governo do Estado do Rio Grande do Norte, da Prefeitura Municipal do Natal, da Prefeitura Municipal de Parnamirim, da CABO Telecom, da ESIG Software e Consultoria, da DynaVideo e do SENAI.

Natal (RN), 26 de julho de 2018.

Chico Dantas (UERN)
Coordenador Geral do CSBC 2018

Anais do CSBC 2018

1º WBCI – WORKSHOP BRASILEIRO DE CIDADES INTELIGENTES

Coordenação Geral

- Frederico Lopes (UFRN)
- Fabio Kon (USP)
- Flávia C. Delicato(UFRJ)
- Paulo F. Pires (UFRJ)

Comitê de Programa

- Adrião Duarte - DCA/UFRN
- Alfredo Goldman - IME/USP
- Allan Martins - DEE/UFRN
- Alvaro de Oliveira - IMD/UFRN
- Antonio Augusto Frohlich - UFSC
- Augusto José Venancio Neto - DIMAp/UFRN
- Augusto Sampaio - CIn/UFPE
- Bernadete Loscio - CIn/UFPE
- Carlos Eduardo da Silva - IMD/UFRN
- Daniel Batista - IME-USP
- Daniel Sabino - IMD/UFRN
- Daniel Sadoc Menasche - UFRJ
- Danielo Gomes - UFC
- Edmundo Madeira - IC-Unicamp
- Everton Cavalcante - DIMAp/UFRN
- Fabio Kon - IME/USP
- Fabio Moreira Costa (UFG)
- Flavia C. Delicato - UFRJ
- Flávio de Oliveira Silva - UFU
- Frederico Lopes - IMD/UFRN
- Guilherme H. Travassos - UFRJ
- Jair Leite - DIMAp/UFRN
- Jose Marcos Nogueira - DCC/UFMG
- Kelly Rosa Braghetto – IME/USP
- Kiev Gama - CIn/UFPE
- Leandro Aparecido Villas -IC-Unicamp
- Luis Henrique M. K. Costa - UFRJ
- Luiz Fernando Bittencourt -IC-Unicamp
- Markus Endler - PUC-Rio
- Nelio Cacho - DIMAp/UFRN
- Noemi Rodriguez - PUC-Rio
- Paulo Pires - UFRJ
- Rossana Andrade - UFC
- Sergio Soares - UFPE
- Thais Batista - DIMAp/UFRN
- Wagner Meira Junior - DCC/UFMG

SUMÁRIO

Arquitetura para Construção de Índices Ambientais Apoiada por Ontologias: um Estudo Exploratório sobre Qualidade do Ar

Mateus G. Belizario[1], Ricardo M. Taques[2], Elizziane M. B. Farias[2], Cesar A. Tacla[1,2], Rita C. Berardi[1]

[1]Departamento de Informática – Universidade Tecnológica Federal do Paraná (UTFPR), CEP 80230-901, Curitiba, PR, Brazil

[2]Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (CPGEI)
Universidade Tecnológica Federal do Paraná (UTFPR)
CEP 80230-901, Curitiba, PR, Brazil
{mateusbelizario, rtaques, elizzianefarias}@alunos.utfpr.edu.br,
{tacla, ritaberardi}@utfpr.edu.br

Abstract. *Public policy-making often comes from a participatory process. Thus, this project proposes the development of an architecture that allows users, both citizens and managers, to build their own environmental indexes, monitor them in real time and publish them in an open format. It is intended to allow citizens to be able to monitor the level of pollution in the region in which they live without requiring access to specific data provided by institutions holding the information. To achieve these objectives, an architecture is proposed that uses an ontology as a way to define a common vocabulary and to allow the semantic relationship of the data.*

Resumo. *A elaboração de políticas públicas, muitas vezes, provém de um processo participativo. Assim, este projeto propõe o desenvolvimento de uma arquitetura que permita a usuários, tanto cidadãos quanto gestores, construir seus próprios índices ambientais, monitorá-los em tempo real e publicá-los em formato aberto. Pretende-se conceder aos cidadãos a possibilidade de acompanhar o nível de poluição da região em que têm interesse sem necessitar requisitar acesso a dados específicos fornecidos por instituições detentoras da informação. Para alcançar esse objetivo, propõe-se uma arquitetura que utiliza ontologias como forma de definir vocabulários e assim permitir o relacionamento semântico dos dados.*

1. Introdução

Segundo Lemos (2013), o conceito de cidades inteligentes surge com o intuito de incentivar o ambiente público na tomada de decisões, ampliar os laços comunitários e a participação política. O cenário potencial onde se insere o problema abordado neste artigo é um no qual a tecnologia pode contribuir neste aumento de participação permitindo uma melhora no monitoramento e controle da poluição ambiental o que pode atenuar os danos ao meio ambiente. Apesar de todos os possíveis benefícios que a aplicação do conceito de cidades inteligentes possui no que tange ao monitoramento de agentes poluentes em centros urbanos, existem obstáculos que dificultam a coleta, manipulação e inferência de

informação sobre este grande número de dados. Quando o foco é o sensoriamento de indicadores ambientais, uma das problemáticas estudadas é a dificuldade de se encontrar dados abertos para livre acesso e processamento. E quando eles existem, são disponibilizados em formatos muito distintos e sem padronização para coleta e análise, assim exige um grande esforço para conseguir relacionar e extrair informações deste material.

Tendo em vista o cenário de indicadores ambientais e a problemática apontada, propomos uma arquitetura que visa realizar o tratamento de dados oriundos de sensores de forma semântica, com o objetivo de relacioná-los e conectá-los, gerando informações úteis a partir desses dados e disponibilizando-as para o uso por diferentes tipos de usuários. Além disso, com essa arquitetura, os usuários finais poderão compor novos índices de controle e acompanhamento a partir do tratamento semântico aplicado sobre os dados de medições previamente incorporados em seu modelo ontológico.

A proposta descrita passou por uma primeira fase de desenvolvimento, através da qual foi possível realizar um estudo exploratório com dados de sensores de qualidade de ar inspirados nos cenários de uso do IAP (Instituto Ambiental do Paraná). Esse estudo exploratório ajudou a visualizar os desafios relacionados ao monitoramento de dados em cidades inteligentes, e compreender o contexto no qual ele é aplicado.

Este artigo está organizado como segue. Na seção 2, outras propostas de sistemas de monitoramento são apresentadas juntamente com suas arquiteturas. A seção 3 apresenta a arquitetura proposta por esta pesquisa e como ela foi utilizada em um estudo exploratório. Na seção 4 são compartilhados os resultados alcançados com a arquitetura proposta por meio de um estudo exploratório. Na seção 5 são expostas as discussões advindas da execução deste projeto, as conclusões sobre ele e por fim apresentamos possibilidades para trabalhos futuros.

2. Trabalhos Correlatos

Diversas cidades têm utilizado sistemas de monitoramento a partir de sensores ou dispositivos de coleta de dados para obtenção de informações relevantes sobre o meio ambiente, para controlar ou mesmo antecipar o surgimento de problemas. Dentre eles, encontramos projetos que procuram soluções para um tráfego urbano mais eficiente, para controle da qualidade da água ou da qualidade do ar nas regiões com muito tráfego de veículos ou próximas de indústrias e fábricas.

Assim, foram estudadas arquiteturas em trabalhos correlatos que fazem uso ou não de ontologias com o intuito de compreender como ocorre o tratamento de dados gerados por sensores para a geração da informação e, como são compartilhados estes dados com os usuários.

2.1. Arquitetura de sistemas de monitoramento para *Smart Cities*

No projeto descrito por Montori et al. (2017), a arquitetura SenSquare tem a finalidade de tratar medições do ambiente originadas de dispositivos diversos, de fontes confiáveis ou não que são armazenadas em um formato padronizado, acrescidas de metadados e referências espaço-temporais. Em seu módulo principal de processamento são incorporados outros dados referentes aos dispositivos e usuários que participaram das coletas em sua base de dados. Para exteriorizar resultados, esta arquitetura utiliza-se de

APIs para controlar o acesso e a disponibilização de informações na nuvem em modo *open data* para uso por serviços *web* ou *mobile*.

Shahanas e Bagavathi (2016) apresentam um sistema de gerenciamento da qualidade da água. Sua arquitetura faz uso de sensores pelos quais os dados são coletados e transmitidos para um servidor Arduino e Raspberry Pi. Na sua camada intermediária são executados scripts para analisar e gerar resultados que são compartilhados via SMS ou correio eletrônico. Os dados advêm de sensores em PoU (*Points of Use*) dentro de reservatórios de água, são depois carregados em uma base MySQL e scripts os processam gerando alertas e relatórios. Ademais uma interface web foi criada para apresentar informações sobre as condições da água.

Uma solução semelhante é vista em Yuntao et al. (2016). Sua arquitetura é hierarquicamente dividida em quatro camadas, cada qual responsável por funções específicas. Em sua base estão dispostos os sensores sobrepostos por uma rede física e uma rede sem fio que faz uso de um framework complexo para capturar dados somente de sensores, estáticos ou móveis. Além disso, outros módulos realizam simulações, diagnósticos, emitem alertas prévios, enviam dados de ajustes para os reservatórios, elaboram planos de emergência para incidentes que são direcionados para componentes do sistema. Seu objetivo principal é permitir um controle eficiente das condições da água em todo o sistema hidráulico envolvido.

2.2. Arquiteturas que utilizam ontologias no monitoramento ambiental

Com outro tipo de abordagem, Claudine et al. (2012) propõem em seu projeto, para uso em cidades inteligentes, uma ontologia chamada de OUPP (*Ontology of Urban Planning Process*) utilizada em conjunto com a ontologia CityGML (OGC, 2006). Este projeto serve para analisar as condições do ar em um ambiente urbano a partir das relações semânticas entre estes dois modelos. Em sua modelagem, OUPP contempla conceitos e axiomas ligados aos fatores do clima como temperatura e vento. Na modelagem presente em CityGML[1] estão conceituados os elementos geométricos de uma cidade, como ruas e edificações. Com o alinhamento semântico entre as ontologias tornou-se possível auxiliar na melhoria do controle de poluição do ar em espaços urbanos. Entretanto, não são descritos aspectos da arquitetura utilizada no projeto que nos impede saber como os dados recebidos são tratados e como são disponibilizados.

Outra proposta que faz uso de ontologias para monitoramento ambiental é descrita em Oprea (2005) em seu sistema especialista chamado de SBC_MEDIU. Sua arquitetura é composta de uma base de conhecimento, uma máquina de inferência e de uma base de dados com padrões e medições armazenadas em séries temporais. Em seu módulo sobre poluição do ar denominado por SBC_AIR ocorre o uso da base de conhecimento implantada pelo sistema DIAGNOZA_MEDIU. Este *framework* atua sobre os dados, regras e parâmetros realizando avaliações para produzir diagnósticos como o estado e o risco de poluição do ar, além de sugerir soluções preventivas. Quatro anos depois, Oprea (2009) incorporou o uso de uma ontologia de domínio para fazer as análises e o controle de poluição do ar em regiões urbanas com atividades industriais denominada de

[1] Disponível em https://www.citygml.org/

AIR_POLLUTION_Onto. Para o desenvolvimento desta ontologia foram identificados termos específicos como *Pollutant, Pollution_Source, Emission, CO, SO_2, NO_2, O_3, PM_{10}, $PM_{2.5}$* entre outros, que compõem a sua taxonomia.

As principais características que permitem classificar e estudar os sistemas de monitoramento presentes na literatura se referem a conectividade do sistema em relação a diferentes fontes de dados, a existência de tratamento semântico dos dados em contrapartida ao uso de bancos de dados com esquemas de acesso via SQL e a possibilidade de exteriorização das informações em formato aberto.

3. Arquitetura Proposta: Smart Architecture for Quality Environment Indicators (S.A.Q.E.I.)

Dentre as funcionalidades da arquitetura se encontram a homogeneização de dados coletados de fontes e formatos distintos, a criação de uma rede de dados conectados através de um vocabulário próprio e a construção e manipulação de índices e indicadores ambientais por parte do usuário final. Nas próximas seções são detalhadas as camadas da arquitetura e o fluxo de informação interno, desde a coleta de dados de sensores até a disponibilização em formato aberto dos indicadores criados por usuários. A Figura 1 apresenta as camadas da arquitetura e suas principais responsabilidades e papéis de forma simplificada na arquitetura.

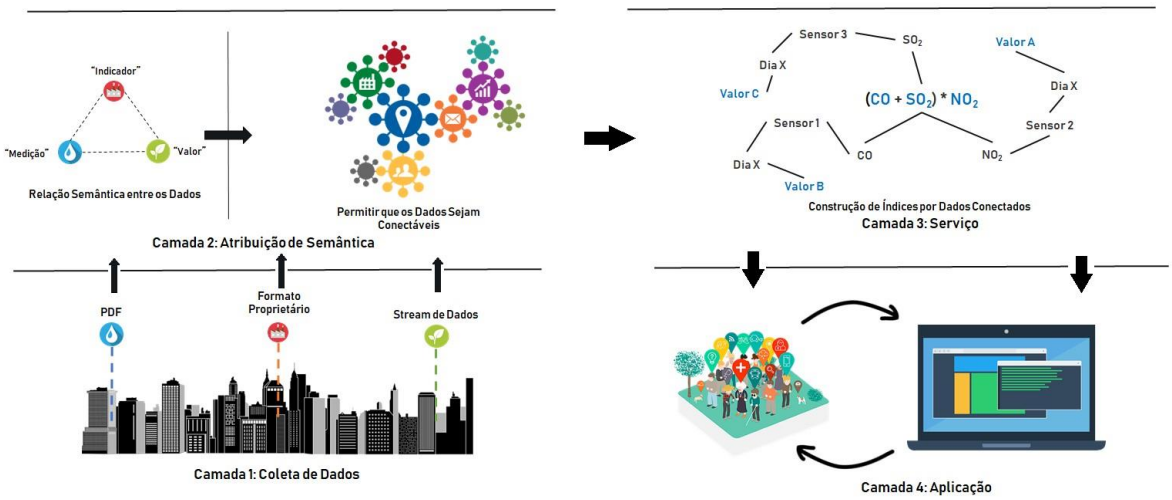


Figura 1: Camadas da S.A.Q.E.I

3.1. Camadas da Arquitetura

Para cada camada é realizada uma avaliação das suas funções, vantagens para a arquitetura e os dados esperados em sua entrada e saída.

3.1.1. Primeira camada: Coleta de dados

A primeira camada possui a função de coletar os dados de sensores de fontes privadas ou públicas como institutos ambientais de monitoramento ou repositórios de dados abertos municipais ou estaduais. Esses dados podem ser recebidos diretamente dos sensores via *streaming* de dados, neste caso o monitoramento e a entrada de medições e informações

são contínuos e a arquitetura deve ser capaz de tratar uma grande quantidade de dados de forma rápida e dedicada.

Na proposta defendida por Seric et al. (2011) a primeira camada atua sobre os sensores que fazem a leitura de dados dos fenômenos que os interessam. Assim, seu agente coletor captura estes dados e os armazena em um banco de dados. No projeto Common Scents (CS) conduzido pelo laboratório do Massachusetts *Institute of Technology* (M.I.T) os dados sobre as condições ambientais são oriundos de sensores móveis instalados em bicicletas. Na sua camada de tratamento de dados, são utilizadas bases de dados dispostas em servidores para a geração de dados integrados e padronizados.

Outro cenário possível é o de que as próprias instituições detentoras da informação tratem os dados coletados pelos sensores e os forneçam em diversos formatos, como arquivos CSV, PDF, formatos proprietários ou até mesmo imagens. Ainda deve-se ressaltar que esta camada poderá receber dados estáticos ou dinâmicos, de fontes distintas e em formatos diferentes. Tratar tantas formas de dados é uma função laboriosa e por isso buscou-se na literatura formas de se trabalhar com eles que possam ser implementados na S.A.Q.E.I.

3.1.2. Segunda Camada: Atribuição de semântica

A camada seguinte se refere à preparação para a conectividade dos dados pela atribuição de valor semântico aos mesmos baseando-se na ontologia. A ontologia desenvolvida tem o objetivo de definir conceitos e relações relativos à construção e monitoramento de índices ambientais, por meio de classes, axiomas e relações. Porém, sempre que possível deve fazer o reuso de conceitos já definidos em ontologias de alto nível. Foi o caso desta ontologia de domínio ao reutilizar conceitos presentes nas ontologias *time.owl*, *ssn.owl* desenvolvidas pela *World Wide Web Consortium* (W3C) e na *environment.owl* desenvolvida no projeto GCI pela Universidade de Toronto-CA. Entretanto, outros conceitos tiveram que ser definidos visando o controle das medições coletadas e a disponibilização de indicadores e índices derivados destes em classes da ontologia.

A ontologia é quem realiza a homogeneização do significado dos dados, i.e., que viabiliza o compartilhamento de informações que antes estavam dispersas em diferentes formatos e linguagens. Com os dados semanticamente conectados é possível realizar inferências lógicas. Desta forma, é possível agregar valor às informações já presentes, como propõe Sheth (2014) ao descrever o conceito de *smart data*, conforme mais dados instanciarem os conceitos e relações da ontologia. Portanto, esta camada pode receber dados crus ou tratados, e tem como saída um arquivo RDF com base na ontologia desenvolvida.

3.1.3. Terceira Camada: Serviços

Com base na ideia de Kon et al. (2016), a terceira camada deve ser vista como uma "*Middleware* de Serviços" desenvolvida para implementar os serviços e geração de informação para publicação. Ela se encarrega da lógica de aplicação, criação das relações de índices personalizados de usuários e cálculo de indicadores com valores de tempo de coleta dinâmicos e índices compostos de indicadores diversos.

A arquitetura deve permitir a representação de medições e valores ao longo do tempo de diversos indicadores ambientais. Além disso, deve permitir que os usuários possam monitorar os índices e também construir e desenvolver seus próprios índices ambientais. A saída da camada são medições para a construção de novos índices ambientais.

3.1.4. Quarta Camada: Aplicação

A última camada é a de aplicação e representa o consumo dos dados gerados pela arquitetura pelos usuários finais. Ela deve conter um terminal para consulta SPARQL para usuários com conhecimento para manipulação direta de ontologias. Para os que não possuem tal conhecimento, deve haver uma interface para construção, monitoramento e compartilhamento de índices ambientais, sendo este um requisito importante desta camada.

Os meios para se chegar à essa interface ainda estão sendo estudados, mas acredita-se que seja fundamental representar os dados conectados e o domínio que a ontologia representa sem se apoiar em competências técnicas. Com isso qualquer pessoa poderá criar iniciativas baseadas nas informações disponibilizadas.

4. Estudo de Caso: Monitoramento da Qualidade do Ar com os Dados de Sensores do Instituto Ambiental Paranaense (IAP)

Como prova de conceito (POC) da arquitetura S.A.Q.E.I, utilizamos o contexto local do Estado do Paraná. O IAP, entidade vinculada à Secretaria Estadual de Meio-Ambiente, realiza a coleta de dados sobre os agentes poluentes em várias regiões do estado. Atualmente, o monitoramento de índices ambientais ocorre em vários estados do Brasil, mas sem a integração de dados em tempo real e sem a padronização dos dados gerados.

Não existem no momento projetos de monitoramento que permitam aos cidadãos explorarem informações sobre indicadores a fim de que possam gerar os seus próprios. Com base nisso, descrevemos a seguinte situação: um cidadão deseja obter dados sobre a qualidade do ar em diferentes regiões da cidade de Curitiba para saber quais são as áreas de maior risco para pessoas com problemas respiratórios. A seguir mostramos como a arquitetura pode estar presente neste caso mais concreto. Para a POC, foi utilizado um protótipo da plataforma implementada na linguagem Java e com Apache Jena Framework.

4.1. Coleta dos Dados

Na primeira camada, os dados necessários são obtidos por meio de sensores das estações de coletas de dados ambientais que podem operar em modo automático, semiautomático ou manualmente. Esses dados atualmente, são disponibilizados em formato PDF[2], diferentemente do formato aberto, proposto pela S.A.Q.E.I. Devido a este formato, foi necessário executar uma preparação dos dados com PostGreSQL para estrutura dos dados em formato CSV (*Comma-separated Values*).

Para a POC utilizamos o cálculo do índice de qualidade de ar calculado pela função $(CO + SO_2) * NO_2$. A função é composta por três indicadores ambientais distintos:

[2] http://www.iap.pr.gov.br/pagina-1076.html

o CO (monóxido de carbono), o SO_2(dióxido de enxofre) e o NO_2 (dióxido de nitrogênio). Cada um desses indicadores possui um valor específico de acordo com o método de medição selecionado pelo usuário. Os sensores podem disponibilizar o "pior" valor do dia selecionado, a média de todas as coletas do dia ou até mesmo o "melhor" valor coletado. As referências de "pior" e "melhor" modificam de acordo com o indicador. Outro dado também a ser considerado é o local do sensor, cuja medição pode ser especificamente de uma estação ou uma média de todas as estações da cidade. A Tabela 1 mostra os dados coletados para a POC, indicadores de 3 regiões de Curitiba: Boqueirão, Santa Cândida e Cidade Industrial.

Coletas de dados de Indicadores do dia 01/04/2016			
	Boqueirão	**Santa Cândida**	**Cidade Industrial**
CO	3 µg/m3	1 µg/m3	7 µg/m3
SO2	1 µg/m3	1 µg/m3	24 µg/m3
NO2	1 µg/m3	16 µg/m3	1 µg/m3

Tabela 1: Dados coletados de 3 regiões de Curitiba

4.2. Atribuição Semântica

Os dados provenientes de sensores são interpretados e instanciados na ontologia desenvolvida para representar índices e indicadores ambientais. Foi desenvolvida uma ontologia para representar a semântica das medições com base em estudos sobre representação de índices e indicadores ambientais apresentados por Dahleh e Fox (2016) e Fox (2015). A ontologia representa um vocabulário comum para as medições de diferentes estações assim como possibilita a criação dos índices representando através das relações. A tabela 2 mostra a semântica associada às medições de monóxido de carbono (CO) das estações Boqueirão e Cidade Industrial (CIC). Essa atribuição semântica mostra a interoperabilidade que a ontologia permite no tratamento de dados oriundos de diferentes padrões de sensores em diferentes estações, que muitas vezes não são padronizados em um único formato.

As 6 triplas RDF da Tabela 2 são formadas pelos elementos: sujeito, predicado e objeto. Por exemplo, a medição mostrada na Tabela 2 do indicador CO na estação de coleta Boqueirão com o valor de 3 micrograma por metro cúbico recebe semântica pelo conjunto de triplas representadas nas linhas [1], [3] e [5] da Tabela 2.

Na tripla representada na linha [1] da Tabela 2 está representado semanticamente que a medição diz respeito ao indicador CO na estação Boqueirão. Na tripla na linha [3] na Tabela 2 está representando semanticamente que a medição do CO na estação Boqueirão aconteceu no dia 01/04/2016.

	Medições são relacionadas à uma estação de coleta:
[1]	< *AQ_ Measurement_CO, wasAttributedTo* , IAP_Station_Boqueirão>
[2]	< *AQ_ Measurement_CO, wasAttributedTo* , IAP_Station_CIC>
	Medições são relacionadas a seu dia e horário de coleta:
[3]	< *AQ_ Measurement_CO_Boqueirão* , *hasHappenedOn* , "2016-04-01T01:00:00">
[4]	< *AQ_ Measurement_CO_CIC* , *hasHappenedOn* , "2016-04-01T01:00:00">
	Medições são relacionadas a seu valor de coleta:
[5]	< *AQ_ Measurement_CO_Boqueirão, microgram_per_cubic_metre, 3*>
[6]	< *AQ_ Measurement_CO_CIC, microgram_per_cubic_metre, 7*>

Tabela 2: Exemplo de Atribuição de semântica

E finalmente, na tripla [5] da Tabela 2 a tripla está representado semanticamente o valor 3 para a medição em micrograma por metro cúbico de CO no Boqueirão no dia 01/04/2016. Analogamente acontece com as triplas das linhas [2], [4] e [6] para a estação da Cidade Industrial.

4.3. Serviços

A camada três faz o papel de provedora das informações geradas pelo processamento da camada anterior. Também, acrescenta o conteúdo que servirá de insumo para as aplicações consumidoras deste sistema viabilizarem a exteriorização dos dados. Nesta camada, a linguagem de *queries* SPARQL permite consultar os dados armazenados em formato de triplas RDF.

	Indicador recebe a média das medições realizadas no dia:
[1]	< *AQ_Indicator_of_CO, microgram_per_cubic_metre, 5*>

Tabela 3: Cálculo da função do indicador

A tabela 3 mostra a tripla que representa o valor de um indicador na arquitetura. Neste exemplo, representamos semanticamente que o indicador carbono (CO) tem valor de 5 microgramas por metro cúbico. Esse valor se refere à média das medições realizadas no dia para o carbono em uma estação de coleta específica. A arquitetura proposta foi capaz de representar diferentes índices ambientais de acordo com as particularidades requeridas pelo usuário. Além de relacionar a interface de uso da aplicação e o *back-end* da arquitetura, onde são definidos o vocabulário, classes e relações entre os dados e onde ocorre a coleta, conexão e atribuição semântica dos dados.

	Criação do Índice
	Atribuição a um usuário
[1]	< *AQ_ Index_A, wasAttributedTo* , User_A>
	Índice é relacionadas a seus indicadores:
[3]	< *AQ_ Index_a* , *hasIndicator* , *AQ_Indicator_of_CO*>
[4]	< *AQ_ Index_a, hasIndicator* , *AQ_Indicator_of_SO2*>
[5]	< *AQ_ Index_a, hasIndicator* , *AQ_Indicator_of_NO2*>

Tabela 4: Representação semântica do índice ambiental

A tabela 4 se trata da representação semântica do índice construído pelo usuário. A linha [1] representa que o índice criado pertence a um usuário, aqui tratado como User_A. Além disso, a partir da linha [3] são representadas as relações dos índices com os indicadores presentes nele, assim, descrevemos que o índice definido anteriormente possui os seguintes indicadores: CO, SO2 e NO2. Cada um desses indicadores já foi calculado e possui um valor de coleta, como mostrado na Tabela 3.

4.3. Aplicação

Nessa aplicação foi possível manipular os dados conectados através da ontologia para permitir o usuário construir novos índices ambientais e visualizá-los sem a dependência de uma linguagem técnica para consulta destes dados.

	Função Índice: **(CO + SO2) * NO2**
Cidade Iindustrial:	31 µg/m3
Boqueirão:	4 µg/m3
Santa Cândida:	32 µg/m3

Tabela 5: Resultado da função construída

A tabela 5 mostra os resultados para a função criada pelo usuário para três regiões da cidade de Curitiba: Cidade Industrial, Boqueirão e Santa Cândida. Podemos ver que o primeiro bairro e o último tiveram resultados altos e, portanto, não são adequados para pessoas com problemas respiratórios, que era a necessidade de informação do usuário na POC.

5. Conclusão

Foi apresentada a proposta de uma arquitetura para tratar dados oriundos de sensores referentes a medições ambientais para a criação e monitoramento ambiental por meio de indicadores de qualidade. O diferencial desta arquitetura é a manipulação dos dados através de ontologias que trazem semântica para dados dos sensores e ampla disponibilização para cidadãos em formato aberto para reutilização. Para a validação da proposta, foi desenvolvido um protótipo em Java e Jena cuja utilização foi através de uma prova de conceito (POC) utilizando indicadores do IAP na cidade de Curitiba para o controle de qualidade do ar.

Com a prova de conceito foi possível observar como cada camada da arquitetura proposta executa sua função, bem como, também foi possível visualizar as dificuldades e questões que exigem estudos para de fato tornar viável o uso da arquitetura, tais como: o tratamento para streaming de dados para grande fluxo de informação; a flexibilização da entrada de dados por diferentes fontes; a solução a ser adotada no caso dos dados não estarem em formato aberto e estruturado como o RDF, a interface para possibilitar a manipulação de dados semânticas sem exigir conhecimento de RDF.

A principal característica dessa arquitetura é que as camadas são independentes e possuem entradas e saídas próprias. Além disso, ela é capaz de integrar dados de diferentes fontes e em diferentes formatos, tratando seus valores e significados pela ontologia construída. E por fim, permitindo aos cidadãos utilizar estes dados, disponibilizados de forma aberta pela arquitetura e adquiridos por sensores, para construir seus próprios índices ambientais e compartilhá-los ou utilizá-los para tomada de decisão ou aplicação em outras ações sociais, políticas e tecnológicas que queiram.

Outra discussão gerada se trata da disponibilização dos dados em formato aberto e permitir a inclusão da população nesta ferramenta sem a necessidade de conhecimentos técnicos e científicos sobre dados conectados, ontologias e computação geral. Estuda-se, como trabalhos futuros, uma forma de conseguir aproximar a consulta de dados em ontologias de uma forma mais natural para permitir que a população geral tenha acesso à essas informações também e não tão somente uma comunidade específica de pesquisadores. Essa aspiração torna-se ainda mais importante quando se conta com a participação da população e gestores em uma solução *linked* data para cidades inteligentes.

Referências

Dahleh, D. & Fox, M. S. (2016). “An Environmental Ontology for Global City Indicators” (ISO 37120), 50.

Fox, M. S. (2015). “The role of ontologies in publishing and analyzing city indicators *Computers, Environment and Urban Systems”, Elsevier,* 54, 266-279.

Kon, F. et al. (2016) “Cidades Inteligentes: Conceitos, plataformas e desafios”. Jornadas de Atualização em Informática 2016—JAI, p. 17.

Lemos, A. (2013) “Cidades inteligentes”. GV-executivo, v. 12, n. 2, p. 46-49.

Métral, Claudine et al. (2007). “Ontologies for the Integration of Air Quality Models and 3D City Models”. CoRR abs/1201.6511: n. pag.

Montori, F., Bedogni, L. and Bononi, L. (2017) "A Collaborative Internet of Things Architecture for Smart Cities and Environmental Monitoring". In: IEEE Internet of Things Journal, vol. PP, no. 99, pp. 1-1.

OGC (2006). Candidate OpenGIS CityGML Implementation Specification (City Geography Markup Language). Approved discussion paper of the Open Geospatial Consortium, Inc. related to CityGML specification, 06-057r1.

Oprea M. M. (2009). “AIR_POLLUTION_Onto: an Ontology for Air Pollution Analysis and Control”. AIAI 2009. IFIP International Federation for Information Processing, vol 296, pp. 135–143, Springer, Boston, MA.

Oprea, M. (2005) “A case study of knowledge modelling in an air pollution control decision support system”. AI Commun. 18, 4 (December 2005), pp. 293-303.

Ribeiro, L. C. Q. (2015) “Estado da motorização individual no Brasil – Relatório 2015”. Observatório das Metrópoles. Universidade Federal do Rio de Janeiro - UFRJ. Instituto de Pesquisa e Planejamento Urbano e Regional - IPPUR. 2015

Shahanas, K. M. and Sivakumar, P. B. (2016) “Framework for a Smart Water Management System in the Context of Smart City Initiatives”. In: India Procedia Computer Science, 92, pp. 142 - 147

Sheth, Amit. (2014) "Smart data—How you and I will exploit Big Data for personalized digital health and many other activities." *Big Data (Big Data), 2014 IEEE International Conference on*. IEEE.

Spaargaren, G. (1997). The ecological modernization of production and consumption: Essays in environmental sociology. Spaargaren, 1997.

Yuntao, Y., Lili, L., Hongli, Z. and Yunzhong J. (2016) "The System Architecture of Smart Water Grid for Water Security", Procedia Engineering, Volume 154, 2016, pp. 361–368.

Ljiljana Šeric Darko Stipanicev, M. Š. (2011) Observer network and forest fire detection Information Fusion, 2011, Volume 12, Issue 3, 160-175

Monitoramento Ambiental de Cidades Urbanas: Detectando Outliers via Análise Fatorial Exploratória

Thiago I. A. Souza[1], **Andre L. L. Aquino**[2], **Danielo G. Gomes**[1]

[1]Universidade Federal do Ceará (UFC)
Grupo de Redes de Computadores, Engenharia de Software e Sistemas (GREat)
Av. Mister Hull, s/n – Campus do Pici – Bloco 942–A
60455-760 – Fortaleza – CE – Brasil

[2]Instituto de Computação – Universidade Federal de Alagoas (UFAL)
Caixa Postal 57.072–970 – Maceió – AL – Brasil

thiagosouza@great.ufc.br, alla@laccan.ufal.br, dgomes@great.ufc.br

Abstract. *In recent years, smart cities have emerged as a vast repository of data. Thus, the need arises to detect important events that are outside the standard of normality, called outliers. In this paper, we present a new outliers detection approach to intelligent urban environment monitoring data based on the Exploratory Factor Analysis (EFA), using the following procedures: first, we apply EFA generating a factorial-base structure; in the sequence, the distance of Mahalanobis is calculated on the factors extracted for the outliers detection. Real data from the Spanish cities of Elda and Rois validated our proposal and the EFA revealed the most influential factors in the detected outliers patterns.*

Resumo. *Nos últimos anos, as cidades inteligentes têm emergido como um vasto repositório de dados. Assim, surge a necessidade de detectar importantes eventos que estão fora do padrão de normalidade, os chamados outliers. Neste artigo, propomos uma nova abordagem de detecção de outliers para dados de monitoramento de ambientes urbanos inteligentes baseada na Análise Fatorial Exploratória (AFE), através dos seguintes procedimentos: primeiro, aplicamos AFE gerando uma estrutura fatorial-base; na sequência, a distância de Mahalanobis é calculada sobre os fatores extraídos para a detecção de outliers. Dados reais das cidades espanholas de Elda e Rois validaram nossa proposta e a AFE revelou os fatores mais influentes nos padrões de outliers detectados.*

1. Introdução

De acordo com a Perspectiva de Urbanização Mundial[1] publicada pelas Nações Unidas, mais da metade da população vive atualmente em áreas urbanas e em até 2050 cerca de 70% da população global residirá nas cidades. Ao mesmo tempo que a urbanização cresce em um ritmo acelerado modificando o tecido urbano das grandes cidades juntamente com sua geometria, o fenômeno da Tecnologia da Informação e Comunicação (TIC) também cresce de forma extraordinária, sendo estimado pela CISCO[2] que até o final de 2018 será mais de 10 bilhões de dispositivos mobile conectados, o que

[1]http://esa.un.org/unpd/wup/

[2]http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html

viabiliza a integração de múltiplas disciplinas, incluindo redes celulares de quinta geração (5G), redes ad roc heterogêneas, redes móveis híbridas, redes de sensores sem fio, dentre outras, que englobam a chamada Internet das Coisas (*Internet of Things*, IoT). Como resultado da evolução dessa infraestrutura tecnológica, surge o paradigma da cidade inteligente, que integra todos os serviços urbanos habilitados pela TIC em um sistema combinado, de modo que a cidade possa ser inteligente e seja capaz de facilitar o acesso a diferentes tipos de serviços, compartilhamento de informações, monitoramento urbano em tempo real, e assim por diante [Bi et al. 2017].

IoT é uma facilitadora-chave do desenvolvimento de uma vasta gama de aplicações em cidades inteligentes, viabilizando a implementação de uma grande quantidade de nós sensores dentro de uma determinada área, uma vez que esses nós se comunicam entre si para coletar dados e fornecer referências para cidades inteligentes em seus mais diversos serviços ofertados aos cidadãos como indústria, agricultura, segurança, transporte, saúde, ambiente, dentre outros [Qiu et al. 2017]. Dessa forma, IoT está produzindo cada vez mais uma grande quantidade de diferentes tipos de dados, que inclui a localização geográfica do nó sensor, monitoramento da qualidade de água, níveis de poluição sonora e ambiental, controle de resíduos, iluminação, tráfego de veículos, dados de detecção de eventos, e assim por diante. Entretanto, com o aumento do número de dados, padrões de comportamento que se desviam da normalidade do conjunto de observações também tem crescido, denominados na literatura de *outliers* [Camacho et al. 2016]. Em particular, melhorar a robustez da detecção de eventos com base em dados estratégicos, coletados a partir do monitoramento dos diversos setores das cidades torna-se um problema crítico dada a enorme quantidade e heterogeneidade dos dados.

Diante dessa problemática, propomos neste artigo uma nova abordagem de detecção de *outliers* para dados de monitoramento de ambientes urbanos inteligentes baseada na Análise Fatorial Exploratória (AFE), através dos seguintes procedimentos: 1) AFE é realizada sobre os dados para obter uma estrutura fatorial base na qual analisamos e nomeamos os fatores a partir dos padrões identificados; 2) a distância de Mahalanobis é calculada sobre os fatores extraídos para a detecção de eventos fora dos padrões de normalidade do conjunto observado. Dados reais das cidades da Espanha, Elda e Rois, coletados da plataforma de monitoramento ambiental Smart Citizen, foram utilizados para validar nossa abordagem.

O restante deste artigo está organizado como se segue. A Seção 2 apresenta os trabalhos relacionados. Na Seção 3 a metodologia proposta para a nova abordagem de detecção de *outliers* via AFE é apresentada. A metodologia proposta é validada para um conjunto de dados reais na Seção 4. Por fim, na Seção 5, são apresentadas as considerações finais e trabalhos futuros.

2. Trabalhos Relacionados

Vários esforços na literatura foram realizados na perspectiva de propor metodologias e esquemas de detecção de *outliers* a partir do monitoramento de determinadas variáveis ambientais no intuito de contribuir para o desenvolvimento das cidades inteligentes. Por exemplo, [Zanella et al. 2014] apresentou um levantamento abrangente de arquiteturas, protolocos e tecnologias para a proposta de uma metodologia baseada

em serviços web para o projeto da cidade inteligente de Padova (Itália), em que sua implementação engloba soluções que vislumbram o monitoramento e detecção de eventos anormais a partir de dados de iluminação pública e qualidade do ar. Um estudo sobre os fundamentos de IoT no desenvolvimento das cidades inteligentes foi realizado em [Jin et al. 2014], que analisou a ocorrência de *outliers* extraídos a partir do monitoramento de ruído. Já [Filipponi et al. 2010], em seu trabalho apresentou um esquema para implementação de serviços de informação para o monitoramento de áreas públicas e infraestruturas, testando sua abordagem em um cenário de transporte público (metrô) apresentando um sistema que auxilia na detecção de *outliers* e simplifica as comunicações em casos de emergência no transporte. [Souza et al. 2017] propôs uma nova abordagem de detecção de *outliers* a partir do monitoramento ambiental de dados coletados de diferentes e heterogêneos sensores através da aplicação de técnicas da estatística multivariada, como a Análise de Componentes Principais (PCA), trazendo como contribuição o diagnóstico das causas que geraram os eventos discrepantes, indicando quais variáveis ambientais contribuíram para o comportamento anômalo dos dados.

O presente trabalho diferencia-se dos anteriores por combinar a técnica multivariada Análise Fatorial Exploratória com a distância de Mahalanobis no intuito de extrair os fatores latentes mais representativos para a detecção de *outliers* a partir do sensoriamento de medidas ambientais heterogêneas de ambientes urbanos inteligentes.

3. Material e Métodos

Esta seção apresenta o delineamento metodológico no qual informamos como foram realizadas a coleta e a organização dos dados, a decorrente modelagem da análise fatorial exploratória e a detecção de *outliers*.

3.1. Coleta e Organização de Dados

A plataforma Smart Citizen[3], utilizada em pesquisas anteriores [McKercher et al. 2017, Souza et al. 2017], é uma ferramenta de *crowdsourcing* que coleta dados ambientais urbanos e tem a capacidade de enviar dados para a Internet. O dispositivo mede variáveis ambientais como temperatura, umidade, luminosidade, monóxido de carbono, dióxido de nitrogênio e níveis de ruído, podendo enviar dados via Wi-Fi que podem ser visualizados online em tempo real. Diante do exposto, as medidas de tais variáveis (registradas por um período de 31 dias do mês de Julho de 2017) de duas cidades da Espanha, Elda e Rois, foram coletadas e organizadas em matrizes, cada uma representando uma respectiva cidade. Assim, a análise fatorial exploratória é realizada sobre as matrizes geradas, cada uma com dimensões, 31 (tempo em dias) $\times$ 6 (variáveis ambientais).

3.2. Modelagem da Análise Fatorial Exploratória

A Análise Fatorial Exploratória (AFE) é um dos métodos da estatística multivariada que tem como objetivo principal a redução de dimensionalidade e identificar as relações subjacentes entre as variáveis medidas, determinando o número e a natureza apropriada dos fatores comuns (fatores latentes) necessários para explicar a matriz de correlação observada [Bartholomew and Knott 1999]. Assim, podemos distinguir

[3]https://smartcitizen.me/

ou classificar as variáveis de acordo com as contribuições dos fatores latentes para cada variável individualmente. O modelo expressa um vetor **x** m-dimensional (variáveis ambientais) como

$$\mathbf{x} = \mathbf{\Lambda f} + \mathbf{e}, \tag{1}$$

em que $\mathbf{\Lambda}$ $(m \times k)$ é a matriz de coeficientes ou fatores de carregamento e **f** é o vetor k-dimensional dos fatores comuns (fatores latentes), com $k \leq m$, e **e** é um vetor m-dimensional dos termos residuais do modelo (fatores específicos).

Para simplificar ainda mais nosso modelo original, adotamos um modelo de fatores ortogonais com base em três pressupos [Basilevsky 2009]: (i) a média do vetor dos fatores comuns **f** é zero e a matriz de covariância é a identidade; (ii) a média dos termos do vetor de erro **e** é zero e a matriz de covariância é diagonal; e (iii) os termos do vetor de erro não tem correlação com os fatores comuns.

Após a extração dos fatores latentes baseados no modelo da análise fatorial, um índice importante a ser obtido é a comunalidade. A comunalidade para qualquer variável pode ser interpretada como a proporção da variabilidade dessa variável explicada por todos os fatores comuns [Basilevsky 2009]. Assim, a comunalidade h_i para a *i*-ésima variável é definida como

$$h_i = \sum_{j=1}^{k} \lambda_{ij}^2, \tag{2}$$

na qual λ_{ij} é o coeficiente do fator de carregamento da *i*-ésima variável no *j*-ésimo fator comum.

Além das comunalidades, calculamos os escores dos fatores que são definidos como combinações lineares das variáveis observadas. A estimativa dos escores dos fatores com base na regressão é dada diretamente aqui sem derivação [Bartholomew and Knott 1999]

$$\mathbf{S} = \mathbf{\Lambda R}^{-1}\mathbf{X}, \tag{3}$$

em que **S** denota os escores dos fatores e **R** é a matriz de correlação das variáveis observadas, representadas pela matriz **X**.

3.3. Detecção de *Outliers*

A distância de Mahalanobis é uma estatística responsável por indicar variações desiguais, identificando valores ou eventos discrepantes de um determinado conjunto de observações [Hotelling 1947]. Portanto, uma vez que a detecção de *outliers* visa estabelecer se um novo padrão é significativamente diferente de um padrão anterior, utilizamos tal medida combinada com a AFE na detecção de eventos discrepantes no cenário de ambientes urbanos inteligentes. A distância de Mahalanobis nesta pesquisa é dada por

$$D_M^2 = (\mathbf{s} - \mu)^T \mathbf{\Sigma}^{-1} (\mathbf{s} - \mu), \tag{4}$$

na qual **s** é o vetor de escores dos fatores calculados na Equação 3, μ é a média dos escores dos fatores, $\mathbf{\Sigma}$ é a matriz de covariância dos escores e D_M^2 é a estatística de desvio.

Portanto, neste trabalho a distância de Mahalanobis D_M é calculada sobre os vetores **s** de escores dos fatores extraídos da AFE de cada cidade e comparada com um valor limite calculado. Os limites aproximados de controle da distância de Mahalanobis, com um nível de confiança α podem ser determinados de diferentes maneiras, aplicando-se os pressupostos de distribuição de probabilidade [Tracy et al. 1972]

$$T_\alpha = \frac{d(n^2-1)}{n(n-d)} F_\alpha(d, n-d), \tag{5}$$

em que $F_\alpha(d, n-d)$ é o limite superior do percentil da distribuição F com d e $n-d$ graus de liberdade. Assim, se $D_M^2 > T_\alpha$, então as observações são consideradas *outliers*, caso contrário, normais.

4. Resultados

A análise fatorial exploratória foi realizada sobre o conjunto de dados que engloba as 6 variáveis observadas durante um período de 31 dias do mês de Julho de 2017 de duas cidades da Espanha, Elda e Rois, em que uma solução de dois fatores foi derivada para cada cidade. Nas subseções seguintes serão apresentados e discutidos os critérios de validação da AFE bem como a rotulagem de cada fator extraído e suas interpretações, e a etapa de detecção de *outliers* baseada na análise fatorial. Todas as simulações realizadas nesta pesquisa foram implementadas no *software* MATLAB, utilizando um processador *Intel Core I5* com velocidade de 2.66GHz e 4GB de memória.

4.1. Validação dos Resultados

Análises preliminares foram realizadas para examinar a adequação dos dados à AFE (Tabela 1). Para tanto, no intuito de verificar se as variáveis analisadas são correlacionadas entre si, gerando a hipótese de a matriz de correlação das variáveis ser identidade, os testes de Kaiser-Meyer-Olkin (KMO) e Bartlett foram aplicados sobre os dados de cada cidade. Conforme observado na Tabela 1, o valor do teste KMO se mostrou significativo tanto para os dados da cidade de Elda (0,71), quanto da cidade de Rois (0,75), garantindo uma boa adequação da amostra a aplicação da AFE, uma vez que possui valor superior a 0,6 [Bartholomew and Knott 1999]. Já o teste de Bartlett rejeitou a hipótese de que a matriz de correlação seria a matriz identidade [Basilevsky 2009]. Desta forma, ambos os métodos mostraram que os dados são adequados para aplicação da AFE.

Tabela 1. Testes de Validação da AFE.

Cidade	Adequação da Amostra - KMO	Esfericidade de Bartlett
Elda	0,71	489,08
Rois	0,75	435,89

4.2. Análise das Comunalidades

A variância extraída de cada nova variável foi comparada com as variâncias das variáveis originais, verificando o quanto de variância comum (comunalidade), existe entre

as variáveis observadas e as que foram obtidas através da AFE. Desta forma, conforme observado na Tabela 2, os valores de comunalidade das variáveis apresentam valores superiores a 0,6, indicando que todas as variáveis, para ambas as cidades, apresentam elevada representatividade dentro dos fatores extraídos pela AFE.

Tabela 2. Valores de Comunalidade das Variáveis Ambientais das Cidades.

Cidade	Variáveis	Extração	Cidade	Variáveis	Extração
Elda	Temperatura	0,84	Rois	Temperatura	0,74
	Umidade	0,89		Umidade	0,69
	Luminosidade	0,81		Luminosidade	0,67
	NO_2	0,93		NO_2	0,85
	CO	0,95		CO	0,92
	Ruído	0,99		Ruído	0,74

4.3. Seleção e Interpretação dos Fatores

Para a seleção do número de fatores, foi utilizado o critério da variância explicada [Sundberg and Feldmann 2016, Souza et al. 2017] cujos primeiros dois fatores explicam cerca de 73% da variância total para a cidade de Elda (Tabela 3), e cerca de 83% da variância para a cidade de Rois (Tabela 4). Além da variância explicada, foi utilizado também o critério de Kaiser [Kaiser 1966, Souza et al. 2017], que diz que os fatores a serem considerados devem apresentar autovalores acima da unidade ($\lambda > 1$, segunda coluna da Tabela 3 e Tabela 4).

Tabela 3. Distribuição da Variância Explicada da AFE - Cidade de Elda.

Fatores	Autovalores (λ)	% Variância	% Variância Acumulativa
1	2,89	48,17	48,17
2	**1,51**	**25,18**	**73,35**
3	0,99	16,69	90,05
4	0,32	5,44	95,49
5	0,18	3,13	98,62
6	0,08	1,39	100

Tabela 4. Distribuição da Variância Explicada da AFE - Cidade de Rois.

Fatores	Autovalores (λ)	% Variância	% Variância Acumulativa
1	2,17	46,23	46,23
2	**1,61**	**36,90**	**83,14**
3	0,98	7,41	90,55
4	0,57	5,66	96,22
5	0,47	3,13	99,35
6	0,17	0,67	100

Os fatores de carregamento permitem que uma correlação possa ser estabelecida entre as variáveis observadas e os fatores extraídos. Desta forma, tanto para a cidade de Elda (Tabela 5) quanto para a cidade de Rois (Tabela 6), todas as cargas com valores superiores a 0,6 estão destacadas em negrito. Nesta pesquisa, a análise desses fatores se

estabeleceu a partir do cruzamento de cargas elevadas com as demais variáveis. Assim, a partir do padrão observado deste cruzamento, nomeamos os fatores extraídos de cada cidade que obtiveram os maiores valores, conforme discutido a seguir:

- **Fatores de Carregamento da Cidade de Elda -** Analisando os valores dos fatores das cargas desta cidade (Tabela 5), ambos têm em comum o fato de se referirem prioritariamente às variáveis climáticas, uma vez que para as variáveis, temperatura e umidade, os fatores apresentaram os maiores valores para o Fator I. Dessa forma, para fins de análise nomeamos o Fator I como **Condições Climáticas**.
- **Fatores de Carregamento da Cidade de Rois -** Os fatores de carregamento para esta cidade (Tabela 6) apresentam características relacionadas à poluição da cidade. Isto devido ao fato das variáveis monóxido de carbono (CO) e dióxido de nitrogênio (NO_2) apresentarem os maiores fatores de carregamento para o Fator II. Assim, o Fator II recebeu a denominação de **Qualidade do Ar**.

Tabela 5. Fatores de carregamento da AFE - Cidade de Elda.

Variáveis Ambientais	Fator I	Fator II
Temperatura	**0,99**	0,59
Umidade	**0,95**	0,53
Luminosidade	0,03	0,15
NO_2	0,11	0,167
CO	0,41	0,09
Ruído	0,20	0,11

Tabela 6. Fatores de carregamento da AFE - Cidade de Rois.

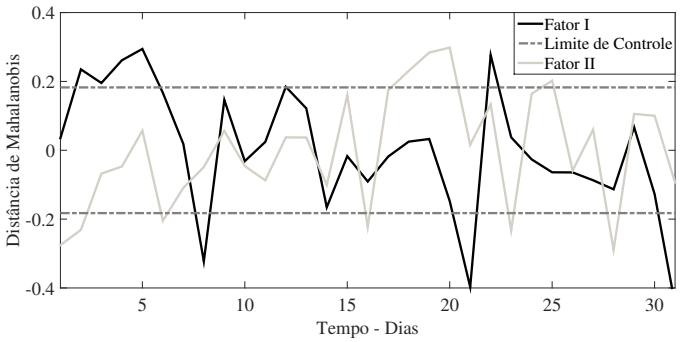
Variáveis Ambientais	Fator I	Fator II
Temperatura	0,39	0,38
Umidade	0,43	0,56
Luminosidade	0,19	0,15
NO_2	0,54	**0,91**
CO	0,49	**0,95**
Ruído	0,01	0,32

Os fatores derivados do modelo exploratório fatorial obtidos neste estudo podem ser utilizados como ferramenta para identificar padrões de eventos relacionados as variáveis ambientais que estejam fora do padrão de normalidade. Como é amplamente conhecido que uma cidade com cidadãos pouco saudáveis dificilmente se torna uma cidade inteligente, uma vez que tais variáveis impactam diretamente na vida dos cidadãos, esses fatores podem revelar padrões de anormalidade que permaneceriam invisíveis frente a uma análise de dados que apenas explorasse a natureza descritiva dos dados.

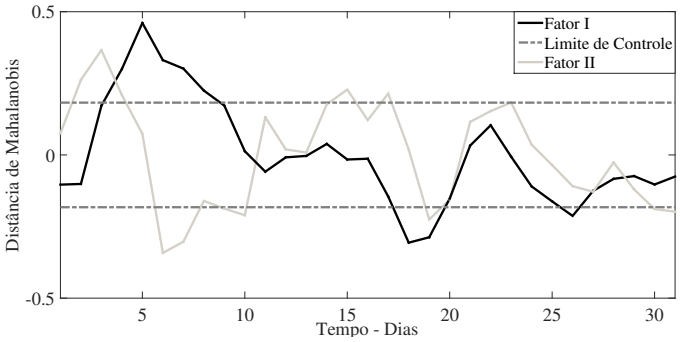
4.4. Detecção de *Outliers* via Análise Fatorial Exploratória

A partir dos fatores extraídos, de acordo com os critérios de variância explicada e Kaiser, calculamos a distância de Mahalanobis para ambos os fatores extraídos de cada cidade, no intuito de serem utilizados como uma distribuição de referência empírica para

estabelecer uma região gráfica de controle para o monitoramento do comportamento das variáveis ambientais. Assim, se os valores da estatística permanecem dentro das regiões de controle, não há evidências de que o processo em análise sofreu algum tipo de mudança. Entretanto, caso os valores em algum instante sejam traçados fora do limiar de controle, há evidências de que o processo sofreu alguma tipo de alteração. Portanto, quando os fatores extraídos têm uma interpretação clara, o gráfico estatístico que descreve o comportamento dos fatores fornece uma ilustração visual útil para analisar os perfis das variáveis ambientais das cidades.



(a)



(b)

Figura 1. Detecção de *outliers* para: (a) Cidade de Elda; (b) Cidade de Rois.

Os resultados da detecção de *outliers* através da distância de Mahalanobis aplicada sobre os dois fatores extraídos de cada cidade são apresentados na Figura 1 e Tabela 7. É importante destacar que o parâmetro T_α, calculado na Equação 5 da Seção 3.3, foi utilizado nesta pesquisa para delimitar a fronteira de monitoramento, no intuito de inferir qual região do gráfico está fora dos limites de controle. Para a cidade de Elda, verificamos conforme a Tabela 7 que $32,25\%$ e $29,03\%$ de eventos do Fator I e Fator II, respectivamente, estavam fora das fronteiras estabelecidas pelo limite de controle. Já a cidade de Rois, para o Fator I e Fator II, respectivamente, os percentuais foram de $25,80\%$ e $41,94\%$ (Tabela 7) de eventos que ultrapassaram os limites de controle. Esses resultados permitem-nos analisar o dia em que determinado evento anormal aconteceu, e assim verificar o instante do ocorrido, podendo a informação ser utilizada pelo orgão público responsável por monitorar padrões ambientais bem como servir de *insights* para as tomadas de decisões futuras por parte dos gestores públicos.

Na Figura 1a, T_α é o parâmetro delimitador dos valores da distância de Mahala-

Tabela 7. Distribuição de *outliers* por fator.

	Fator I (%)	Fator II (%)
Elda	32,25	29,03
Rois	25,80	41,94

nobis sobre os dois fatores extraídos. Observamos que na maioria dos dias não há alertas para eventos fora dos limites, contudo o Fator I supera em cerca de 3% no número de *outliers* em relação ao Fator II. Este resultado permite-nos inferir que o Fator Condições Climáticas (conforme nomeado na subseção 4.3) foi o responsável por influenciar na geração de *outliers* para a cidade de Elda, revelando um importante comportamento discrepante por parte das variáveis ambientais, temperatura e umidade, em determinados dias do mês de Julho. Este fator pode ser útil para revelar padrões de conforto ou desconforto climático da cidade.

Na Figura 1b, observa-se novamente determinados instantes em que o valor da distância de Mahalanobis é superior ao limite de controle T_α, sinalizando alertas de ocorrência de eventos discrepantes. Verificamos que apesar do Fator I registrar o maior pico no dia 5 de Julho, o Fator II supera em cerca de 16% na quantidade de *outliers* detectados, sendo o fator responsável por influenciar significativamente na geração de tais eventos. Portanto, o Fator Qualidade do Ar (conforme denominado na subseção 4.3), aponta para o comportamento discrepante das variáveis monitoradas relacionadas a poluição do ar, monóxido de carbono (CO) e dióxido de nitrogênio (NO_2), em determinados períodos do mês de Julho de 2017 da cidade de Rois. A análise deste fator pode ser útil para apontar tendências dos níveis de qualidade do ar, contribuindo para o monitoramento efetivo dos índices de poluição ambiental da cidade.

5. Conclusão

Neste artigo propomos uma nova abordagem de detecção de *outliers* para dados de monitoramento de ambientes urbanos inteligentes baseada na Análise Fatorial Exploratória (AFE). Através dos resultados alcançados com a aplicação da AFE, uma estrutura fatorial-base foi obtida revelando os fatores latentes mais rerpesentativos, os quais foram nomeados, a saber: **Fator Condições Climáticas** e **Fator Qualidade do Ar**. A partir dos fatores latentes extraídos pelo modelo multivariado, a distância de Mahalanobis foi calculada sobre os escores dos fatores, tomando os valores da estatística como um recurso de identificação de eventos discrepantes, caso ultrapassem o limite de controle estabelecido. Padrões de *outliers* foram identificados para ambos os fatores: para o Fator Condições Climáticas, constatamos que as variáveis ambientais, temperatura e umidade, foram as responsáveis por gerar o comportamento discrepante; para o Fator Qualidade do Ar, as variáveis ambientais, monóxido de carbono e dióxido de nitrogênio, foram as que influenciaram no comportamento anômalo dos dados.

Como perspectivas de trabalhos futuros, sugere-se: (i) aumentar o número de amostras e de cidades; (ii) realizar uma análise de correlação entre variáveis ambientais; (iii) utilizar os fatores da AFE para a análise de *cluster*.

Agradecimentos

Thiago Iachiley agradece a CAPES, André L. L. Aquino agradece ao CNPq, FAPEAL e FAPESP e Danielo G. Gomes agradece ao CNPq (respectivamente, processos 88882.183548/2018-01, 311878/2016-4 e 432585/2016-8) pelo apoio financeiro.

Referências

Bartholomew, D. J. and Knott, M. (1999). *Latent Variable Models and Factor Analysis.* Arnold Publishers.

Basilevsky, A. T. (2009). *Statistical factor analysis and related methods.* John Wiley and Sons.

Bi, Y., Lin, C., Zhou, H., Yang, P., Shen, X., and Zhao, H. (2017). Time-constrained big data transfer for sdn-enabled smart city. *IEEE Communications Magazine*, 55:44–50.

Camacho, J., Villegas, A. P., Teodoro, P. G., and Fernandez, G. M. (2016). Pca-based multivariate statistical network monitoring for anomaly detection. *Computers and Security*, 59:118–137.

Filipponi, L., Vitaletti, A., Landi, G., Memeo, V., Laura, G., and Pucci, P. (2010). Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors. In *Sensor Technologies and Applications, SENSORCOMM*, pages 281–286. Fourth International Conference.

Hotelling, H. (1947). *Multivariate quality control.* In: Techniques of statistical analysis. NewYork: McGraw-Hill.

Jin, J., Gubbi, J., Marusic, S., and Palaniswami, M. (2014). An information framework for creating a smart city through internet of things. *IEEE Internet of Things Journal*, 1:112–121.

Kaiser, H. F. (1966). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, 20:141–151.

McKercher, G. R., Salmond, J. A., and Vanos, J. K. (2017). Characteristics and applications of small, portable gaseous air pollution monitors. *Environmental Pollution*, 223:102–110.

Qiu, T., Liu, J., Si, W., Han, M., Ning, H., and Atiquzzaman, M. (2017). A data-driven robustness algorithm for the internet of things in smart cities. *IEEE Communications Magazine*, 55:18–23.

Souza, T. I. A., Magalhães, D. M. V., and Gomes, D. G. (2017). Aplicando estatística multivariada para detecção e diagnóstico de anomalias em dados urbanos. *Anais do I Workshop de Computação Urbana (CoUrb)*, 1:72–85.

Sundberg, R. and Feldmann, U. (2016). Exploratory factor analysis—parameter estimation and scores prediction with high-dimensional data. *Journal of Multivariate Analysis*, 148:49–59.

Tracy, N. D., Young, J. C., and Mason, R. L. (1972). Multivariate control charts for individual observations. *Expert Systems With Applications*, 24:88–95.

Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things Journal*, 1:22–32.

Integração, Relacionamento e Representação de Dados em Cidades Inteligentes: Uma Revisão de Literatura

Larysse Silva, José Alex Lima, Nélio Cacho, Eiji Adachi, Frederico Lopes, Everton Cavalcante

Universidade Federal do Rio Grande do Norte (UFRN)
Natal-RN, Brasil

larysse.savanna@ufrn.edu.br, j.alex.medeiros@gmail.com, neliocacho@dimap.ufrn.br, {eijiadachi, fred}@imd.ufrn.br everton@dimap.ufrn.br

Resumo. *Uma característica notável de cidades inteligentes é o aumento da quantidade de dados gerados produzidos pelos mais diversos dispositivos e sistemas computacionais, ampliando assim os desafios do desenvolvimento de software que envolva a integração de grandes volumes de dados. Nesse contexto, este artigo apresenta uma revisão de literatura a fim de identificar as principais estratégias utilizadas no desenvolvimento de soluções de integração, relacionamento e representação de dados em cidades inteligentes. Este estudo selecionou e analisou de forma sistemática onze artigos publicados entre os anos de 2015 e 2017. Os resultados obtidos evidenciam lacunas com relação a soluções para a integração contínua de fontes de dados heterogêneas a fim de dar suporte ao desenvolvimento de aplicações e tomada de decisão.*

Abstract. *A notable characteristic of smart cities is the increase in the amount of available data generated by several devices and computational systems, thus augmenting the challenges related to the development of software that involves the integration of larges volumes of data. In this context, this paper presents a literature review aimed to identify the main strategies used in the development of solutions for data integration, relationship, and representation in smart cities. This study systematically selected and analyzed eleven studies published from 2015 to 2017. The achieved results reveal gaps regarding solutions for the continuous integration of heterogeneous data sources towards supporting application development and decision-making.*

1. Introdução

A pesquisa e desenvolvimento de soluções em cidades inteligentes são iniciativas mundiais que levam a explorar melhor os recursos de uma cidade a fim de melhorar a qualidade dos serviços para os cidadãos. Os últimos avanços da Tecnologia da Informação (TI) têm proporcionado uma diversidade de tecnologias de *hardware* e *software* que resultam na produção de uma quantidade massiva de dados. Um relatório publicado em 2014 pela *International Data Corporation* (IDC) comprova esse fato ao informar que o volume total de dados gerados e transmitidos no mundo em 2013 foi de 4.4 ZB e a previsão para 2020 é que esse número chegue a ser dez vezes maior [Turner et al. 2014].

De um modo geral, inclusive no contexto de cidades inteligentes, a produção, coleta e disponibilização de dados geralmente são realizadas de forma distribuída, isolada

e sem um padrão definido. Isso ocorre devido à grande heterogeneidade de *stakeholders* envolvidos nas soluções para cidades inteligentes, tais como cidadãos, governo, indústria, academia etc., que geram e consomem dados e informações relacionadas a seus próprios interesses. Outro fator que contribui para essa diversidade é a ausência de políticas relacionadas ao compartilhamento e disponibilização desses dados, que geralmente possuem formatos definidos de acordo com a facilidade ou a capacidade técnica dos seus respectivos produtores [Souza et al. 2017]. Outro aspecto importante diz respeito à natureza desses dados, que são diversos, podem não estar devidamente estruturados e conter ruídos ou imprecisões que afetam de forma negativa a sua utilização por aplicações e usuários finais.

Devido ao aumento exponencial da quantidade de dados gerados por empresas, pessoas e dispositivos computacionais, o termo *Big Data* vem sendo utilizado para descrever enormes conjuntos de dados que, diferente dos dados tradicionais, geralmente incluem dados não estruturados que software de processamento de dados tradicional ainda não consegue lidar de forma adequada [Chen et al. 2014]. Tais dados são adquiridos a partir de diversos tipos de atividades, a exemplo da captação de medidas por sensores, publicações de usuários em redes sociais, envio de *e-mails* e transações bancárias ou mesmo dados extraídos de arquivos de *log* de servidores Web.

O principal objetivo de uma solução de *Big Data* é oferecer uma abordagem abrangente para o tratamento de volumes de dados caóticos para tornar as aplicações mais eficientes e precisas, transformando dados brutos em informação de valor agregado e conhecimento para os usuários. Além disso, o paradigma de *Big Data* também traz novas oportunidades para descobrir novos valores, auxiliar na obtenção e compreensão mais aprofundada dos valores ocultos, bem como integrar e relacionar dados de diferentes fontes, possibilitando o gerenciamento dos conjuntos de dados de forma efetiva.

Tão importante quanto gerenciar grandes volumes de dados é a sua integração para que seja possível extrair informações relevantes e transformá-las em conhecimentos práticos. Entretanto, atualmente é comum que a produção de dados ocorra de forma não-padronizada, tendo em vista que os sistemas são desenvolvidos de acordo com as necessidades e interesses dos desenvolvedores ou proprietários do sistema, que muitas vezes não levam em consideração outros atores que possam estar interessados nos mesmos dados. Outro fato é que o formato dos dados gerados é definido de acordo com a facilidade ou a capacidade técnica dos desenvolvedores, fazendo com que a integração dos dados seja um grande desafio para aplicações que precisam realizar consultas em diversas fontes de dados independentes e heterogêneas.

Nesse contexto, este trabalho tem como objetivo realizar uma revisão de literatura a fim de investigar as principais estratégias e ferramentas que permitem a integração, relacionamento e representação de grandes volumes de dados no contexto de cidades inteligentes. Para esse fim, foi adotada uma metodologia bem definida que permitiu a busca, seleção e análise sistemática dos estudos disponíveis na literatura [Kitchenham e Charters 2007]. A ideia é que os resultados obtidos permitam não só ter uma visão geral acerca das estratégias atualmente existentes para a integração dados gerados por diversos sistemas no contexto de cidades inteligentes, mas também elencar questões importantes a serem endereçadas em pesquisa e desenvolvimento no futuro.

O restante deste artigo está estruturado da seguinte forma. A Seção 2 discute brevemente alguns trabalhos relacionados a esta pesquisa. A Seção 3 apresenta a metodologia adotada neste trabalho em termos das questões de pesquisa a serem respondidas e as estratégias para busca e seleção dos estudos. A Seção 4 descreve como os estudos relevantes foram selecionados. A Seção 5 provê uma síntese resultante da análise dos estudos selecionados como respostas às questões de pesquisa estabelecidas. A Seção 6 levanta possíveis ameaças à validade da revisão de literatura realizada. Por fim, a Seção 7 apresenta algumas conclusões a partir dos resultados encontrados e direções para trabalhos futuros.

2. Trabalhos Relacionados

É possível encontrar na literatura trabalhos relacionados ao processamento de dados em cidades inteligentes, porém há uma escassez de estudos ligados diretamente a integração, relacionamento e representação de dados em larga escala. A seguir são apresentados alguns trabalhos relacionados encontrados na literatura.

Macêdo *et al.* (2017) fizeram um estudo comparativo sobre ferramentas existentes para publicação e gerenciamento de dados abertos em cidades inteligentes. Como resultado, os autores concluíram que as ferramentas apresentadas não foram capazes de atingir adequadamente critérios como facilidade de uso por usuários sem conhecimento técnico especializado, possibilidade de mapeamento objeto-relacional e qualidade dos dados em termos de completude, confiabilidade e precisão. Por fim, os autores propuseram como trabalho futuro a implementação de uma nova ferramenta visando diminuir a complexidade e os custos da publicação e do gerenciamento de dados abertos.

Chen *et al.* (2015) realizaram uma revisão de literatura sobre grandes volumes de dados no contexto da Internet das Coisas (IoT). Os autores forneceram uma maneira sistemática de analisar a mineração de dados em visão de conhecimento, visão técnica e visão de aplicação, incluindo classificação, agrupamento, análise de associação, análise de série temporal e análise isolada, incluindo os casos mais recentes de aplicações na pesquisa. Os autores analisaram ainda os algoritmos mais recentes de análise de dados a fim de aplicá-los a soluções de *Big Data* e discutiram desafios de pesquisa futura.

3. Metodologia

Uma revisão de literatura permite analisar estudos disponíveis em um determinado domínio e permitem responder a questões de pesquisa sobre o atual estado da arte da literatura ou estado da prática. Contudo, a fim de conferir valor científico a tal revisão, ela precisa ser feita seguindo um procedimento rigoroso e sistemático com vistas a minimizar possível viés e tornar o processo reprodutível por outros pesquisadores. Para isso, é necessário estabelecer um protocolo bem definido com questões de pesquisa e critérios explícitos para avaliar e selecionar os estudos disponíveis na literatura, protocolo esse que precisa ser seguido de forma estrita ao longo de todo o processo [Kitchenham e Charters 2007].

A revisão de literatura apresentada neste trabalho foi conduzida em conformidade com metodologias bem estabelecidas na literatura [Biolchini e Travassos 2005, Kitchenham e Charters 2007]. Como ilustra a Figura 1, as etapas realizadas foram basicamente

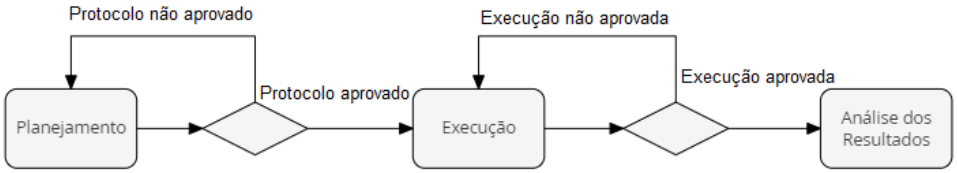


Figura 1. Fases do processo de revisão de literatura

três: (i) *planejamento*, na qual foi elaborado um protocolo definindo as questões de pesquisa a serem respondidas, a estratégia de busca adotada, os critérios utilizados para a seleção dos estudos e os métodos para extração e síntese de dados; (ii) *execução*, na qual os estudos foram identificados, selecionados e avaliados de acordo com o protocolo estabelecido, e; (iii) *análise dos resultados*, na qual as informações extraídas dos estudos selecionados foram agregadas considerando as questões de pesquisa, bem como feita uma análise e discussão acerca dos resultados encontrados.

Questões de pesquisa. O objetivo principal desta revisão de literatura foi identificar estudos relevantes que apresentassem contribuições na elaboração de estratégias para integração, relacionamento e representação de grandes volumes de dados de diferentes fontes no contexto de cidades inteligentes. Com esse objetivo em vista, foram definidas as seguintes questões de pesquisa (QPs):

QP1: Quais as principais estratégias adotadas para o desenvolvimento de sistemas de integração, relacionamento e representação de dados em cidades inteligentes?
QP2: Quais as principais ferramentas utilizadas na integração, relacionamento e representação de dados em cidades inteligentes?
QP3: Quais os principais desafios encontrados no desenvolvimento de sistemas de integração, relacionamento e representação de dados?

Estratégia de busca. Para recuperar os estudos relacionados ao objetivo central deste trabalho, foi utilizado um processo automatizado de busca sobre três bases eletrônicas de publicações científicas, a saber, ACM Digital Library[1], IEEEXplore[2] e Google Scholar[3]. Com base nas questões de pesquisa, foi elaborada a seguinte *string* de busca:

```
(data integration E data representation E data relationship) E
                (smart city OU smart cities)
```

Critérios de seleção. Critérios de seleção foram utilizados para avaliar cada um dos estudos recuperados a fim de incluir apenas estudos de fossem de fato relevantes para responder às QPs e excluir aqueles que não contribuíssem para respondê-las.

Foram considerados os seguintes critérios de inclusão (CIs):

CI1: O tema do estudo está relacionado a integração, relacionamento e representação de dados em cidades inteligentes.

[1] http://dl.acm.org
[2] http://ieeexplore.ieee.org
[3] http://scholar.google.com

CI2: O estudo apresenta uma estratégia para a integração, relacionamento e representação de no mínimo dois tipos de dados diferentes.

Foram estabelecidos ainda os seguintes critérios de exclusão (CEs):

CE1: O estudo não está relacionado ao tema de integração de dados.
CE2: O estudo descreve uma abordagem sem detalhes claros e suficientes sobre como realizar a integração, relacionamento e representação de dados.
CE3: O estudo não foi publicado entre janeiro de 2015 e dezembro de 2017.
CE4: O estudo é uma duplicação de outro resultado já recuperado ou um trabalho mais recente em comparação a outro anteriormente publicado.
CE5: O resumo e/ou o texto completo do estudo não está disponível.
CE6: O estudo não foi publicado em Inglês, que é o idioma mais comumente utilizado em publicações científicas.

Nesta revisão de literatura, um estudo foi considerado relevante se este não atendeu a nenhum dos CEs e atendeu a pelo menos um IC.

4. Processo de Seleção

Para recuperação dos estudos a partir das bases eletrônicas de publicação, a *string* de busca foi adaptada a fim de torná-la compatível com as especificidades dos mecanismos de busca de cada uma das bases eletrônicas. Feito isso, o procedimento de busca automatizada foi realizado sobre cada base eletrônica de acordo com a *string* de busca adaptada. A busca automatizada foi limitada apenas aos campos de título, resumo e palavras-chave.

Após recuperar os estudos a partir das bases eletrônicas, o processo de seleção foi conduzido em três etapas. Na Etapa 1, o título e do resumo do estudo foram lidos e analisados em conformidade com os critérios de seleção estabelecidos no protocolo. Na Etapa 2, os estudos filtrados na etapa anterior tiveram as seções de introdução e conclusão lidas e analisadas para verificar se o tema abordado no estudo era realmente compatível com o tópico de interesse deste estudo. Por fim, na Etapa 3, os estudos filtrados na etapa anterior foram lidos por completo, seguindo-se atividades de extração de dados para apoiar a elaboração de respostas às questões de pesquisa anteriormente estabelecidas.

Após a aplicação da *string* de busca nas bases eletrônicas, foram encontradas 290 publicações científicas no total, sendo 264 da IEEEXplore, 25 da ACM Digital Library e 1 do Google Scholar. A Tabela 1 apresenta uma evolução da seleção dos estudos na execução das três etapas do processo de seleção. A primeira coluna apresenta a quantidade de artigos identificados na Etapa 1, a segunda coluna exibe os artigos selecionados na Etapa 2 e, por fim, a quantidade de artigos extraídos na Etapa 3.

Tabela 1. Resultados das etapas do processo de seleção dos artigos

Base eletrônica	**Etapa 1**	**Etapa 2**	**Etapa 3**
IEEEXplore	264	38	8
ACM Digital Library	25	4	3
Google Scholar	1	0	0
Total	**290**	**42**	**11**

5. Resultados

Esta seção sumariza os resultados da revisão de literatura realizada considerando as questões de pesquisa e os dados extraídos/sintetizados a partir dos estudos selecionados. As Seções 5.1 a 5.3 apresentam respostas a cada uma das QPs inicialmente definidas no protocolo em termos de estratégias, ferramentas e desafios para a integração, relacionamento e representação de dados em cidades inteligentes.

5.1. Estratégias (QP1)

Dentre as principais estratégias identificadas nesta pesquisa, podem ser citadas:

Souza *et al.* (2017) desenvolveram um *middleware* de dados chamado *Smart Geo Layers* baseado em dados geoespaciais que visa unificar os dados fornecidos por diversas fontes em ambientes de cidades inteligentes, permitindo que usuários de diversas organizações possam compartilhar e consumir dados. Os autores desenvolveram um modelo unificado de dados para facilitar a integração de dados de diferentes tipos e formatos. A arquitetura da solução proposta pelos autores é composta por elementos que dispõem de *APIs RESTfull* para consumir e enviar dados para o *middleware*, um *context broker* que é de um componente intermediário para armazenar camadas registradas no *Smart Geo Layers*, um componente de segurança para realizar autenticação de usuários a fim de proteger dados privados e também foi desenvolvido um componente responsável por traduz as operações fornecidas pelas *APIs* para as linguagens de consulta específicas do sistema de banco de dados (NoSQL ou geoespacial). O banco de dados *NoSQL* armazena atributos não-geográficos das entidades armazenadas no *Smart Geo Layers*, enquanto o banco de dados geoespacial armazena valores de atributos geográficos. Os autores informam que essa separação dos dados visa garantir a escalabilidade, pois as consultas que usam funções geográficas são mais complexas e demoradas do que as consultas com atributos apenas textuais.

Chakraborty *et al.* (2017) destacam o uso da abordagem *Extract-Transform-Load* (ETL, do inglês, Extração, Transformação e Carregamento) para integração de dados. O ETL é um processo de armazenamento de dados que extrai dados de fontes externas, transforma-os em necessidades operacionais, que podem incluir verificações de qualidade e carrega-os no banco de dados de destino. Esse processo é dividido em três fases:

- Extração: nessa primeira fase ocorre o processo de extração dos dados das fontes. Normalmente esses dados são extraídos em formato simples como csv, xls e txt ou por meio de um cliente *RESTfull*.
- Transformação: essa fase compreende a limpeza dos dados, ou seja, remoção de dados duplicados, verificação de violação de integridade, classificação e agrupamento de dados, etc.
- Carregamento: essa fase envolve a persistência dos dados na base consolidada.

Com base no estudo elaborado pelo autor sobre a abordagem ETL, é possível identificar que essa estratégia possui diversas limitações como, por exemplo, a impossibilidade de automatizar as soluções que fazem uso do ETL, havendo a necessidade da intervenção manual principalmente na etapa de transformação dos dados. Outra lacuna encontrada nessa estratégia é a capacidade limitada para extrair dados de diferentes fontes ao mesmo tempo, pois embora na maioria das ferramentas de ETL a consulta possa ser

criada e utilizada, o que é muito semelhante à consulta SQL, só é possível extrair dados de fonte de dados única.

You *et al.* (2017) fizeram uso da abordagem *Informed Design Platform* (IDP) para realizar a coleta, armazenando, limpeza, análise, integração, mineração e visualização de dados. Essa abordagem é composta por cinco componentes, a saber, (i) *objetos*, (ii) um coletor para reunir dados dos objetos, (iii) uma plataforma central de gerenciamento de dados e permitir uma integração de sistemas fracamente acoplada, (iv) de serviços distribuídos modularizados e reutilizáveis para processar dados de várias fontes e (v) um orquestrador de serviços.

5.2. Ferramentas (QP2)

Dentre as principais ferramentas utilizadas nos trabalhos encontrados, pode-se destacar:

O Clover ETL, que é uma plataforma ETL de integração de dados baseada em Java para rápido desenvolvimento e automação de transformações de dados, limpeza de dados, migração de dados e distribuição de dados em aplicações, bancos de dados e armazenamento em nuvem.

O Talend, que é uma ferramenta de integração de dados de código aberto desenvolvido pela Talend e projetado para combinar, converter e atualizar dados em vários locais nos negócios. O Talend Open Studio para Integração de Dados funciona como um gerador de código, produzindo scripts de transformação de dados e programas subjacentes em Java.

O Pentaho Data Integration (PDI) é o componente do Pentaho responsável pelos processos de ETL e pode ser usado para migrar dados entre aplicativos ou bancos de dados, exportar dados de bancos de dados para arquivos simples, carregar dados maciçamente em bancos de dados, limpeza de dados e integração de dados de aplicações. Todo processo no PDI é criado com uma ferramenta gráfica onde o usuário especifica o que fazer sem escrever linhas de código, tornando-o orientado a metadados. Apache Hadoop é uma ferramenta para processamento distribuído de grandes conjuntos de dados em computadores usando modelos de programação simples.

A ferramenta GUIDES é apresentada por Balasubramani *et al.* (2017) como uma nova estrutura de conversão e gerenciamento de dados para sistemas de infraestrutura urbana subterrânea que permite que administradores municipais, trabalhadores e contratados, juntamente com o público em geral e outros usuários, consultem dados digitalizados e integrados para tomar decisões mais inteligentes.

O FIWARE foi utilizado por Souza *et al.* (2017) e é uma plataforma genérica e extensível capaz de lidar com os requisitos essenciais em cidades inteligentes. Essa plataforma oferece componentes que fornecem um ambiente para suportar o desenvolvimento de aplicações que precisam integrar dados heterogêneos, aplicações lógicas e elementos da interface do usuário baseados na Web, entre outras funcionalidades como realizar a autenticação e o gerenciamento de identidades e credenciais de usuários, organizações e aplicações, implementação de interfaces que permitem acessar, editar, visualizar e analisar informações de multicamadas que podem ser representadas por dados geoespaciais.

Sobre as limitações das ferramentas mencionadas, é possível citar que as ferramentas CloverETL e Pentaho não fornecem suporte web semântico para a visualização

do modelo de dados, isso se torna uma limitação tendo em vista que a possibilidade de o usuário selecionar as classes e propriedades adequadas para fornecer relacionamentos entre elas através da visualização é uma parte essencial de um ETL semântico.

5.3. Desafios (QP3)

Os principais desafios encontrados nas soluções de integração, relacionamento e representação de dados são em relação a heterogeneidade, inconsistência, instabilidade e atualização frequente dos dados. A heterogeneidade geralmente é causada por diversos fatores, como dados autônomos, diferenças nas fontes de dados, técnicas de coleta e armazenamento de dados, etc. A inconsistência dos dados é causada principalmente por imperfeições que acabam afetando na qualidade desses dados. Um exemplo disso é quando dois conjuntos possuem valores diferentes para o mesmo atributo. A instabilidade se refere a variabilidade no formato, esquema e outros aspectos dos dados o que pode levar a resultados inválidos no processo de integração. A atualização frequente dos dados se torna um problema devido a frequência de atualização influenciar no rastreamento de eventos. Por exemplo, os dados meteorológicos e de tráfego são disponibilizados a cada hora, enquanto os limites administrativos estão disponíveis em um ciclo anual [Shivaprabhu et al. 2017].

Chakraborty *et al.* (2017) descrevem que o principal desafio ao usar *frameworks* ETL é a necessidade de um especialista de domínios nas fases de extração, transformação e carregamento dos dados para definir o esquema para a origem dos dados já que não existe uma ferramenta ETL capaz de mapear e definir o esquema de origem ou ontologia e carregar os dados automaticamente a partir dos metadados da fonte de dados. Os autores também relatam que *frameworks* ETL são propensos a erros devido ao ruído dos dados e afirmam que é realmente difícil definir regras na estrutura ETL para remoção de dados.

Diversos autores também relatam que tiveram problemas com banco de dados em relação as diferentes semânticas de dados [Chakraborty et al. 2017, Shivaprabhu et al. 2017, Souza et al. 2017]. Outras dificuldades enfrentadas estão relacionadas a compreensão do domínio de como geometrias GIS podem ser mapeadas e como elas podem ser exploradas em outros domínios (como gráficos), a abstração teórica de como diferentes camadas de pontos, linhas, geometrias podem ser mapeadas para multigramas, entre outros.

6. Ameaças à Validade

A fim de garantir alta qualidade e valor científico aos resultados obtidos, um protocolo foi estabelecido com as questões de pesquisa a serem respondidas e critérios explícitos para selecionar e avaliar os estudos. Mesmo assim, potenciais ameaças à validade deste estudo ainda são inevitáveis e podem afetar os resultados obtidos. Nesta seção são discutidas algumas dessas limitações e que estratégias foram adotadas para mitigá-las.

Incompletude do estudo. A principal ameaça à validade deste estudo diz respeito a sua incompletude, uma vez que estudos relevantes podem não ter sido recuperados e selecionados. Para reduzir essa ameaça, foram utilizadas três importantes bases eletrônicas de publicação dentre as existentes, porém ainda há limitações. Primeiro, alguns estudos podem não ter sido recuperados devido a limitações técnicas dos mecanismos de busca das próprias bases eletrônicas. Segundo, as bases eletrônicas escolhidas não representam uma lista exaustiva de fontes de publicação, de modo que outras bases poderiam ser

também incluídas. Terceiro, não foi realizado um processo de *snowballing*, uma técnica que consiste na verificação das listas de referências de cada um dos estudos selecionados a fim de encontrar estudos adicionais que não foram recuperados pela busca automática [Wohlin 2014]. Quarto, apesar de o fato de considerar estudos publicados entre 2015 e 2017 possibilitar focar apenas nos estudos mais recentes, essa decisão pode afetar a completude deste trabalho. Como trabalho futuro, este estudo poderá ser revisado e atualizado para cobrir essas limitações.

Incoerências e inconsistências. Para evitar possíveis incoerências e inconsistências, foi estabelecido um protocolo bem definido para orientar todo o processo de revisão da literatura. Além disso, o método de seleção dos estudos com base nos critérios de inclusão e exclusão foi escolhido com a finalidade de manter o máximo de coerência e fidelidade ao tema proposto. Esses critérios foram discutidos e planejados de maneira cuidadosa a fim de diminuir os riscos de exclusão de estudos relevantes para a pesquisa.

Viés na extração de dados. A existência de viés na extração de dados a partir dos estudos selecionados pode resultar em imprecisões, afetando assim a análise dos estudos. Mais ainda, nem todos os estudos descrevem de forma clara e suficiente as informações que seriam extraídas dos estudos para responder às questões de pesquisa, de modo que foi necessário, em alguns casos, realizar inferências durante a síntese dos resultados. A fim de minimizar tal viés, o protocolo estabelecido procurou ser seguido de maneira estrita.

7. Conclusão

A partir da revisão de literatura apresentada neste trabalho, foi possível fazer um levantamento das principais estratégias e ferramentas utilizadas no desenvolvimento de soluções de integração, relacionamento e representação de grandes volumes de dados de fontes diversas no contexto de Cidades Inteligentes. É possível observar que há um número reduzido de estudos relacionados a este tema de gerenciamento de dados e ainda há limitações nas soluções disponíveis. Foi possível observar também que as principais dificuldades enfrentadas pelos desenvolvedores são em relação a integração contínua das fontes de dados heterogêneas para suportar o desenvolvimento de aplicações e tomada de decisões.

Como trabalhos futuros, será elaborada uma estratégia baseada nos estudos encontrados nesta pesquisa na tentativa de criar uma solução de integração e gerenciamento de dados com menos limitações e capaz de integrar, relacionar e representar dados provenientes de sistemas independentes mantidos por diferentes organizações, a fim de permitir o compartilhamento de dados gerados por diferentes departamentos.

Referências

Balasubramani, B. S., Belingheri, O., Boria, E. S., Cruz, I. F., Derrible, S., Siciliano, M. D. (2017). “GUIDES – Geospatial Urban Infrastructure Data Engineering Solutions”, Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. USA: ACM.

Biolchini, J., Mian, P. G., Natali, A. C. C., Travassos, G. H. (2005) Systematic review in Software Engineering. Technical report, COPPE/Federal University of Rio de Janeiro, Brazil.

Chakraborty, J., Padki, A., Bansal, S. K. (2017) “Semantic ETL – State-of-the-art and

open research challenges", Proceedings of the 11th IEEE International Conference on Semantic Computing. USA: IEEE, pp. 413-418

Chen, F., Deng, P., Wan, J., Zhang, D., Vasilakos, A. V., Rong, X. (2015) "Data Mining for the Internet of Things: Literature review and challenges". International Journal of Distributed Sensor Networks 11(8).

Chen, M., Mao, S., Liu, Y. (2014) "Big Data: A survey". Mobile Networks and Applications 19(2), pp. 171-209.

Kitchenham, B., Charters, S. (2007) Guidelines for performing systematic literature reviews in Software Engineering. Technical report, Keele University, United Kingdom.

Macêdo, J., Cacho, N., Lopes, F. (2017) "A comparative study of tools for smart cities open data publication and management", Proceedings of the 2017 IEEE Summer School on Smart Cities. USA: IEEE – a ser publicado.

Shivaprabhu, V. R., Balasubramani, B. S., Cruz, I. F. (2017) "Ontology-based instance matching for geospatial urban data integration", Proceedings of the 3rd ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics. USA: ACM.

Souza, A., Pereira, J., Oliveira, J., Trindade, C., Cavalcante, E., Cacho, N., Batista, T., Lopes, F. (2017) "A data integration approach for smart cities: The case of Natal", Proceedings of the 3rd IEEE International Smart Cities Conference. USA: IEEE.

Turner, V., Reinsel, D., Gatz, J. F., Minton, S. (2014) The digital universe of opportunities. IDC White Paper, EMC, USA.

Wohlin, C. (2014) "Guidelines for snowballing in systematic literature studies and a replication in Software Engineering", Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering. USA: ACM.

You, L., Tuncer, B., Xing, H. (2017) "Mining place design knowledge from multi-source data in an informed design platform", Proceedings of the 2017 IEEE International Conference on Big Knowledge. USA: IEEE, pp. 276-283.

Uso de aprendizado supervisionado para análise de confiabilidade de dados de *crowdsourcing* sobre posicionamento de ônibus

Diego Vieira Neves, Felipe Cordeiro Alves Dias, Daniel Cordeiro

[1]Escola de Artes, Ciências e Humanidades
Universidade de São Paulo

Abstract. *Intelligent Transportation Systems allows sensors and GPS devices to monitor public transport systems in Smart Cities. Capturing and processing this data should, in theory, allow systems to make the public transport more reliable and predictable for the citizens, which would improve the quality of life of the urban population and the environment. Insufficient or low-quality data, nevertheless, may prevent its use on such real-time systems. This work studies the use of data obtained from crowdsourcing as an alternative to augment this data. In order to mitigate the uncertainties introduced by the crowdsourced data, this work proposes a reliability model for crowdsourced data conceived for the São Paulo bus-based public transport system.*

Resumo. *Sistemas de Transportes Inteligentes permitem o uso de sensores e equipamentos de GPS para monitorar os sistemas de transportes públicos em Cidades Inteligentes. A captura e processamento desses dados permite, em tese, que o cidadão possa utilizar o transporte público com confiabilidade e previsibilidade, o que melhoraria a qualidade de vida da população urbana e o meio ambiente. Contudo, diversos fatores podem fazer com que esses dados sejam insuficientes ou de baixa qualidade para uso em tempo real. Este trabalho estuda o uso de dados obtidos via colaboração coletiva (crowdsourcing) como complemento dessas informações. Para mitigar as incertezas introduzidas pelo uso de crowdsourcing, este trabalho propõe um modelo de análise de confiabilidade dos dados coletados especializado para o sistema de transporte público (por ônibus) do município de São Paulo.*

1. Introdução

O conceito de Cidades Inteligentes (SC — *Smart Cities*) é utilizado para descrever cidades que promovem a integração de diversas tecnologias para solucionar os problemas ocasionados pelo crescimento populacional e, desta forma, criar respostas inovadoras e alinhadas às necessidades da população [De Santis et al. 2014, Batista et al. 2016]. Uma característica importante relacionada a SC que requer respostas inovadoras é a *mobilidade urbana*, que poderia ser melhorada com a disponibilização de serviços de transporte coletivo de qualidade [Figueiredo 2016].

Um problema observável nas cidades brasileiras é a qualidade dos seus serviços de transporte público, especialmente quando nos referimos ao modal ônibus. As prestadoras de serviços têm dificuldades para estabelecer e cumprir horários de itinerários das linhas de ônibus nas grandes metrópoles. A falta de um sistema confiável leva o usuário a não optar pela utilização desses serviços, o que agrava problemas urbanos sociais e ambientais.

As iniciativas em SC têm como objetivo resolver problemas como o mencionado anteriormente. Tais propostas utilizam dados coletados por meio de sistemas distribuídos, per-

tencentes as estruturas tecnológicas utilizadas por essas cidades como, por exemplo, os Sistemas de Transporte Inteligentes (ITS — *Intelligent Transport System*), que realizam a coleta de diversas informações por meio de sensores [Magalhães 2008]. Entretanto, as limitações decorrentes do uso de dispositivos físicos acoplados aos meios de transporte, tais como, indisponibilidade do sinal GPS ou falha na conexão 3G/4G, contribuem para que parte dos dados coletados pelos ITS não possa ser utilizada [Zhu et al. 2011, Pons et al. 2015].

Dados obtidos com *crowdsourcing* são eficazes no contexto de transporte público, pois permitem a obtenção de novas informações em tempo real [Cerotti et al. 2016]. Informações sobre a localização dos ônibus disponibilizadas voluntariamente pelos passageiros poderiam ser combinadas às informações obtidas pelos dispositivos acoplados aos ônibus urbanos, o que contribuiria para a criação de ITS eficientes e confiáveis [Figueiredo 2016, Batista et al. 2016].

Contudo, os dados informados por meio de *crowdsourcing* podem ser subjetivos, imprecisos e incorretos [Misra et al. 2014]. Portanto, cada nova informação obtida via *crowdsourcing* precisa ser analisada e sua confiabilidade assegurada antes de ser incorporada à base de dados de um ITS. Este trabalho propõe um novo modelo para a análise de confiabilidade desses dados por meio de métricas escolhidas para o contexto do modal ônibus, com técnicas de aprendizagem de máquina que permitem identificar de forma eficiente registros com informações não confiáveis (ausentes ou erradas).

2. Fundamentação Teórica

2.1. Cidades Inteligentes

O conceito de Cidades Inteligentes engloba iniciativas com o objetivo de atenuar e gerir eficientemente problemas urbanos que afetam a qualidade de vida da população, decorrentes do constante crescimento populacional [Caprotti 2016]. O termo SC tem sido utilizado para descrever cidades que buscam continuadamente o desenvolvimento urbano, mediante a políticas que estimulam a participação da população por meio de inteligências distintas, como a *(i)* humana, *(ii)* coletiva e *(iii)* artificial (por meio das TICs — Tecnologias da Informação e Comunicação) [Cury and Marques 2016, Batista et al. 2016].

No contexto de mobilidade urbana, um dos objetivos das SC é resolver os problemas relacionados à disponibilização de serviços de transportes com qualidade para a população [Figueiredo 2016]. Os ITS e as técnicas de *crowdsourcing* podem ser descritos com exemplos de soluções que contribuem para aumentar a qualidade dos serviços de transportes oferecidos e a atrair novos usuários [Sussman 2008, Misra et al. 2014, Cerotti et al. 2016].

2.2. Sistemas Inteligentes de Transporte

Os Sistemas de Transportes Inteligentes têm como objetivo coletar as informações sobre os veículos da rede de transporte público para utilizar esses dados em aplicações inteligentes [Figueiredo 2016]. Diferentes soluções em ITS são utilizadas para: *(i)* identificar a localização dos veículos através de sensores ou GPS, *(ii)* transmitir e receber grandes quantidades de dados, *(iii)* processar grandes quantidades de informações e *(iv)* utilizar essas informações para melhorar as condições do tráfego [Sussman 2008].

Um grupo de ITS relevante para esse trabalho são os Sistemas Avançados de Transporte Público (APTS - *Advanced Public Transportations Systems*), que podem ser descritos como um conjunto de aplicações que aumentam a qualidade, segurança e eficiência dos

sistemas de transporte públicos [Figueiredo 2005, Hwang et al. 2006]. Para facilitar o gerenciamento do transporte público, os APTS utilizam a estrutura de um sistema de Localização Automática de Veículos (AVL — *Automatic Vehicle Location*), que possibilita o rastreamento dos veículos em tempo real [Chowdhury and Sadek 2003].

2.3. Crowdsourcing

O termo *crowdsourcing* (*colaboração coletiva* ou *contribuição colaborativa*) se refere ao uso de uma rede distribuída de voluntários dispostos a resolver problemas, desenvolver novas tecnologias, contribuir com dados, etc. [Howe 2006]. As iniciativas baseadas neste conceito estão se tornando essenciais para as infraestruturas das SC, uma vez que possibilitam a captura de diversas informações que normalmente não poderiam ser capturadas por meio da utilização de abordagens e técnicas tradicionais [Cullina et al. 2015, Cerotti et al. 2016].

O uso de *crowdsourcing* em problemas de mobilidade urbana já foi investigado por outros trabalhos [Pedersen et al. 2013, Misra et al. 2014, Cullina et al. 2015, Cerotti et al. 2016]. Tais trabalhos se centraram no problema de como construir Sistemas de Informação ao Usuário (SIU) eficientes, confiáveis e em como permitir melhor interação entre usuários e gestores de serviços de transporte público. Os dados necessários para a criação destes sistemas são fornecidos pelos próprios usuários, por meio dos seus respectivos dispositivos móveis, o que possibilita a atualização constante dos dados. Nenhum dos trabalhos mencionados, entretanto, preocupou-se com a qualidade e confiabilidade dos dados fornecidos pelos usuários.

O aumento da capacidade computacional dos dispositivos móveis possibilitaram aplicações que utilizam técnicas de *crowdsourcing* [Pedersen et al. 2013]. Contudo, assim como ocorre com outras abordagens, os dados obtidos por meio de *crowdsourcing* também apresentam informações não confiáveis, que comprometem a confiabilidade dos dados coletados. Diversos autores [Mashhadi and Capra 2011, Allahbakhsh et al. 2013, Mousa et al. 2015, Daniel et al. 2018] abordam a necessidade da definição e adoção de métodos e técnicas para avaliar a qualidade e garantir a confiabilidade dos dados obtidos com o uso de *crowdsourcing*. Para mitigar as incertezas introduzidas pelo uso de *crowdsourcing*, este trabalho propõe o uso de técnicas de aprendizagem de máquina para criar um modelo de análise de confiabilidade e qualidade desses dados.

2.4. Técnicas de Aprendizagem de Máquina

Aprendizagem de Máquina (ML — *Machine Learning*) são técnicas que utilizam conceitos de inteligência artificial e/ou métodos estatísticos para realizar o reconhecimento de padrões [Mitchell 1997]. São utilizadas no desenvolvimento de sistemas inteligentes capazes de adquirir conhecimento de forma automática por meio da análise de um conjunto de dados [Ghotra et al. 2015]. Os algoritmos de classificação utilizados por essa abordagem são capazes de realizar aprendizagem interativa, mediante a análise de um conjunto de dados chamados de *amostra* [Bishop 2006]. Tais algoritmos podem ser utilizados para resolver problemas de classificação, regressão, *clustering* ou extração de regras, possibilitando, deste modo, a construção de modelos de predição [Ghotra et al. 2015].

Além dos algoritmos utilizados para criar o modelo, outros elementos considerados são a parametrização e complexidade computacional dos algoritmos, as variáveis que serão utilizadas pelo modelo, etc [Mitchell 1997, Ghotra et al. 2015]. A seleção

do melhor algoritmo para a criação de um modelo de predição é uma atividade considerada complexa, devido aos fatores que podem influênciar no desempenho do modelo desenvolvido [Ghotra et al. 2015]. As técnicas mais utilizadas por estudos correlatos [Wu et al. 2004, Wang et al. 2009, Biagioni et al. 2011, Altinkaya and Zontul 2013, Kormáksson et al. 2014] são:

- **Árvore de Decisão**: algoritmo composto por uma estrutura no formato de árvore, aonde cada nó interno representa um determinado teste em uma característica de um registro, e os arcos representam o resultado do teste realizado. A predição da variável-alvo é feita através de regras de decisão simples, inferidas pelos dados de treinamento [Ghotra et al. 2015, Witten et al. 2016].
- **K-Nearest Neighbour**: estrutura que realiza aprendizagem por analogia, ou seja, esse algoritmo relaciona cada um dos registros do conjunto de dados a um ponto em um espaço $m - dimensional$, sendo m o número de atributos de entrada que descrevem o conjunto de dados. Para classificar um novo registro, a similaridade com outros registros já conhecidos é calculada por meios da distância de tais registro ao novo registro [Witten et al. 2016].
- **Regressão logística**: método utilizado para entender a relação entre um conjunto de variáveis independentes (ou explicativas) e uma variável dependente (ou resposta) e construir um modelo que explique essa associação [Hosmer Jr et al. 2013].
- **Análise de discriminante linear**: técnica utilizada para identificar as características que discriminam uma determinada classe ou grupo de dados, e, assim, elaborar previsões a respeito de uma nova observação, identificando o grupo mais adequado a que ela deverá pertencer, em função de suas características [Hair et al. 2009].
- **Gaussian Naive Bayes**: algoritmo baseado no *Teorema de Bayes* e que é utilizado para calcular a probabilidade da ocorrência de um evento, baseando-se em probabilidades obtidas da análise dos eventos passados [Mitchell 1997, Witten et al. 2016].
- **Máquinas de Vetores Suporte**: método supervisionado para predição de rótulos utilizando técnicas de regressão e classificação. Baseia-se na construção de hiperplanos ou espaços dimensionais infinitos [Vapnik 1998].

3. Análise de confiabilidade de dados de *crowdsourcing*

3.1. Procedimentos de coleta e análise dos dados

O modelo de análise de dados de *crowdsourcing* para o transporte público utiliza duas bases de dados: a da *SPtrans*, que contém dados disponibilizados pela empresa São Paulo Transporte S. A. (SPTrans[1]); e a *base de dados crowdsourcing*, disponibilizada pela *startup* Scipopulis[2], especializada em mobilidade urbana.

A primeira base de dados contém os registros referentes aos trajetos realizados pelos ônibus da rede de transporte público da cidade São Paulo. Esses dados foram coletados pelo sistemas AVL instalados nos veículos. A SPTrans disponibiliza os dados coletados em tempo real para os desenvolvedores, acrescidos de informações como: identificação da linha, identificação e localização dos veículos, horários em que foram realizados os registros, entre outras.

Já a segunda base contém dados privados disponibilizados pela *startup* Scipopulis,

[1]São Paulo Transporte S. A.: `http://www.sptrans.com.br/`
[2]Scipopulis: `http://scipopulis.com/`

com informações disponibilizadas voluntariamente pelos usuários do aplicativo Coletivo[3]. Este aplicativo possui uma funcionalidade que permite que usuários informem se um determinado ônibus já passou enquanto o usuário esperava pelo seu ônibus. Quando o passageiro informa que o ônibus chegou no ponto de parada onde está o usuário, o sistema registra informações como (*i*) horário (data e hora atual em que registro foi realizado), (*ii*) localização (latitude e longitude obtida pelo GPS do usuário), (*iii*) identificador do ponto de parada e (*iv*) número da linha do ônibus. A Figura 1(a) apresenta uma visão geral da distribuição dos dados coletados por *crowdsourcing*.

(a) (b)

Figura 1. (a) Mapa de distribuição de localizações das contribuições coletivas, (b) Coleta de dados de posicionamento dos ônibus.

Os dados coletados são referentes aos trajetos realizados por 785 linhas de ônibus da cidade de São Paulo, durante o período de 1º de novembro à 31 de dezembro de 2016. As bases contém mais de 25 milhões registros, totalizando 12 GB de dados. A Figura 1(b) apresenta de forma esquemática como os dados são coletados. Contudo, é importante ressaltar que nem todos os registros contidos nas bases foram utilizados.

Os dados coletados passaram pelo seguinte processo de preparação e seleção, antes de serem utilizados: (i) *consolidação dos dados* em uma única base dos dados referentes aos trajetos realizados pelos ônibus; (ii) *estruturação e normalização* para remoção de dados duplicados, com falta de informação; (iii) *classificação manual* da confiabilidade do dado de acordo com o seguinte protocolo de validação:

- **Distância do usuário em relação ao ponto de ônibus** — (a) diretamente proporcional à probabilidade do dado ser confiável, se inferior a 300 metros ou (b) não confiável, caso contrário;
- **Distâncias anteriores e posteriores do veículo em relação ao ponto de parada informado pelo usuário** — (a) diretamente proporcional à probabilidade do dado ser confiável, se inferior a 10 quilômetros ou (b) não confiável, caso contrário;
- **Velocidade média entre as localizações anteriores e posteriores do veículo em relação a localização do ponto de ônibus informado pelo usuário** — (a) diretamente proporcional à probabilidade do dado ser confiável, se superior 10 km/h e inferior a 80 km/h ou (b) não confiável, caso contrário;
- **Tempos e distâncias dos registros realizados pelos usuários em relação à localização e aos tempos anteriores e posteriores do AVL** — (a) diretamente proporcional à probabilidade do dado ser confiável, se a diferença de tempo entre o registro capturado pelo sistema AVL em relação ao registro informado pelo usuário for inferior a 15 minutos ou (b) não confiável, se superior 15 minutos, e

[3]Aplicativo Coletivo: `https://www.facebook.com/appcoletivo`

(c) diretamente proporcional à probabilidade do dado não ser confiável, quando o tempo e a distância entre os registros capturados pelo equipamento AVL em relação ao registro informado pelo usuário forem inversamente proporcionais.

Por fim, após a execução das etapas anteriores chegamos ao conjunto de dados final contendo 971 registros que servirão como base para o modelo proposto. O conjunto de dados selecionado considera as seguintes informações: identificadores da linha, do veículo, do ponto de parada; tipo de trajeto realizado (ida ou volta); distância do usuário em relação ao ponto de parada obtida por meio das coordenadas de latitude e longitude capturadas a partir dos dispositivos móveis dos passageiros; os tempos em que foram realizados os registros pelos usuários (data e hora); distância da localização anterior do veículo em relação ao ponto de parada obtida por meio das coordenadas de latitude e longitude capturadas a partir dos sistemas AVL instalados nos veículos; os tempos em que foram capturadas as informações do sistema AVL (data e hora); velocidade média necessária para o deslocamento do veículo entre a localização anterior registrada pelo sistema AVL até a nova localização informada pelo usuário; e classificação da qualidade da informação (confiável = 1 e não confiável = 0).

3.2. Descrição do modelo de análise da confiabilidade

O modelo proposto tem como objetivo realizar, de forma automatizada, a análise de confiabilidade dos dados fornecidos pelos usuários (com *crowdsourcing*) sobre o horário de passagem dos ônibus nos pontos de paradas. Para cada registro informado por um usuário, o modelo deverá ser capaz de identificar inconformidades e/ou anomalias que possam influenciar ou comprometer a qualidade dos dados coletados.

Este trabalho avalia o uso de 6 algoritmos de predição diferentes: (i) Árvore de Decisão, (ii) *K-Nearest Neighbour*, (iii) Regressão logística, (iv) Análise de discriminante linear, (v) *Gaussian Naive Bayes*, e (vii) Máquinas de Vetores Suporte. Para a implementação dos algoritmos foi utilizada a biblioteca *Scikit-Learn*[4] [Pedregosa et al. 2011] e em cada modelo implementado, a confiabilidade do conjunto de dados foi analisada considerando-se 4 variáveis, sendo uma variável resposta (y) e 3 variáveis independentes (X), à saber:

- **Classificação**: variável resposta, indica a qualidade da informação dos registros analisados;
- **Distância do usuário**: variável independente que indica a distância entre a localização do usuário em relação ao ponto de parada informado pelo passageiro;
- **Distância anterior**: variável independente que indica a distância entre o ponto de parada informado pelo usuário em relação à localização anterior do veículo;
- **Velocidade**: variável independente que indica a velocidade média necessária para o deslocamento entre a localização anterior do veículo e o ponto de parada informado pelo usuário.

A divisão entre o conjunto de dados de treinamento e testes foi feita utilizando-se validação cruzada (*cross-validation*) [Kohavi 1995], que é usada para avaliar a capacidade de generalização do modelo (classificação de dados desconhecidos) [Witten et al. 2016]. Essa técnica ajuda a evitar os problemas de *overfitting* e *underfitting*, garantido um conjunto

[4] *Scikit-Learn* é uma biblioteca de código open source que possui diversos algoritmos de aprendizado de máquina implementados em linguagem de programação Python.

de dados adequado para o modelo desenvolvido [Witten et al. 2016]. Em particular, foi utilizado o método *K-fold cross-validation*, que consiste em dividir o conjunto de dados aleatoriamente em k segmentos chamados de *folds*. Os modelos desenvolvidos foram treinados e testados com a interação de 10-*folds*, conforme recomendações de diversos autores [Kohavi 1995, Witten et al. 2016].

Avaliou-se também o desempenho do algoritmo em termos de tempo de execução e de avaliação do modelo. Na literatura podem ser encontradas inúmeras medidas que são utilizadas individualmente ou em conjunto para avaliação de modelos de predição [Kohavi 1995, Witten et al. 2016]. Neste estudo, consideramos as seguintes medidas de desempenho: Matriz de Confusão, Acurácia, Precisão, Sensibilidade, Especificidade, *F-score* e *MCC* (*Matthews correlation coefficient*). O tempo de execução é apresentado como a média de 30 execuções.

4. Análise dos resultados

A Tabela 1 retrata as matrizes de confusão geradas para cada um dos 6 modelos utilizados, bem como os indicadores de desempenho para cada modelo. A matriz de confusão consolida a quantidade de registros realmente pertencentes a cada uma das classes (Não confiável e Confiável). Com isso, é possível calcular os indicadores de cada algoritmo como, por exemplo, precisão, sensibilidade, especificidade, *F-score*, MCC e acurácia.

Tabela 1. Indicadores de desempenho dos modelos

Algoritmos	Matriz de confusão			Indicadores de desempenho geral					Indicadores de treinamento		Indicadores de validação	
Árvores de Decisão	Classes	Não confiável	Confiável	Precisão	Sensibilidade	Especificidade	F-score	MCC	Acurácia	Tempo médio de execução (s)	Acurácia	Tempo médio de execução (s)
	Não confiável	88	0	1.00	0.99	1.00	0.99	0.99	0.9882 ± 0.02	0.040699	0.9966 ± 0.01	0.003127
	Confiável	1	203									
K-Nearest Neighbors	Classes	Não confiável	Confiável	Precisão	Sensibilidade	Especificidade	F-score	MCC	Acurácia	Tempo médio de execução (s)	Acurácia	Tempo médio de execução (s)
	Não confiável	71	1	0.98	0.80	0.99	0.88	0.85	0.9117 ± 0.04	0.053361	0.9349 ± 0.05	0.003896
	Confiável	18	202									
Regressão Logística	Classes	Não confiável	Confiável	Precisão	Sensibilidade	Especificidade	F-score	MCC	Acurácia	Tempo médio de execução (s)	Acurácia	Tempo médio de execução (s)
	Não confiável	76	4	0.95	0.85	0.98	0.89	0.86	0.9190 ± 0.06	0.113797	0.9418 ± 0.05	0.009664
	Confiável	13	199									
Gaussian Naive Bayes	Classes	Não confiável	Confiável	Precisão	Sensibilidade	Especificidade	F-score	MCC	Acurácia	Tempo médio de execução (s)	Acurácia	Tempo médio de execução (s)
	Não confiável	74	0	1.00	0.83	1.00	0.91	0.88	0.9352 ± 0.05	0.026670	0.9486 ± 0.04	0.002109
	Confiável	15	203									
Análise Discriminante Linear	Classes	Não confiável	Confiável	Precisão	Sensibilidade	Especificidade	F-score	MCC	Acurácia	Tempo médio de execução (s)	Acurácia	Tempo médio de execução (s)
	Não confiável	21	0	1.00	0.24	1.00	0.38	0.42	0.7570 ± 0.06	0.043261	0.7671 ± 0.05	0.002278
	Confiável	68	203									
Máquinas de Vetores Suporte	Classes	Não confiável	Confiável	Precisão	Sensibilidade	Especificidade	F-score	MCC	Acurácia	Tempo médio de execução (s)	Acurácia	Tempo médio de execução (s)
	Não confiável	5	0	1.00	0.05	1.00	0.10	0.19	0.7186 ± 0.06	0.938610	0.7123 ± 0.06	0.107141
	Confiável	84	203									

Os resultados mostram que os algoritmos de Árvores de Decisão, *Gaussian Naive Bayes* e Regressão Logística apresentaram os melhores indicadores. O modelo desenvolvido utilizando o algoritmo de Árvores de Decisão previu 100% dos registros classificados como confiáveis e 99,65% dos registros considerados como não confiáveis. A acurácia total do modelo foi de 99,66%. Além disso, o modelo apresentou excelentes indicadores

de desempenho com 100% de precisão, 97,75% de sensibilidade, 100% de especificidade, 98,86% de *F-score* e 98,39% de MCC.

Os algoritmos de *Gaussian Naive Bayes* e Regressão Logística apresentaram valores de acurácia similares. Porém, a partir da análise dos demais indicadores, identificamos que o algoritmo de *Gaussian Naive Bayes* apresentou o melhor desempenho. Este modelo obteve acurácia de 94,86%, e previu 100% dos registros classificados como confiáveis e 83,15% dos registros considerados como não confiáveis, além de apresentar indicadores de desempenho com 100% de precisão, 83,15% de sensibilidade, 100% de especificidade, 90,80% de *F-score* e 87,99% de MCC. Já o algoritmo de regressão logística apresentou acurácia de 94,18% e foi capaz de prever 98,03% dos registros classificados como confiáveis e 85,39% dos registros considerados como não confiáveis, e obteve indicadores de desempenho com 95% de precisão, 85,39% de sensibilidade, 98,03% de especificidade, 89,94% de *F-score* e 86,10% de MCC.

O modelo criado com o *K-Nearest Neighbour* identificou 99,50% dos registros classificados como confiáveis e 79,77% dos registros considerados como não confiáveis. A acurácia total do modelo foi de 93,49%. Esse modelo apresentou indicadores com 98,61% de precisão, 79,78% de sensibilidade, 99,51% de especificidade, 88,20% de *F-score* e 84,67% de MCC. Os algoritmos de Análise de Discriminante Linear e Máquinas de Vetores Suporte tiveram o menor desempenho dentre os outros modelos. O modelo com o algoritmo de Análise de Discriminante Linear obteve acurácia total 76,71% e identificou 100% dos registros classificados como confiáveis, contudo o modelo identificou apenas 23,59% dos registros considerados como não confiáveis. Esse algoritmo apresentou indicadores com 100% de precisão, 23,60% de sensibilidade, 100% de especificidade, 38,18% de *F-score* e 42,04% de MCC. Do mesmo modo, o modelo com o algoritmo de Máquinas de Vetores Suporte apresentou acurácia total de 71,23% e, embora tenha identificado 100% dos registros classificados como confiáveis, classificou apenas 5,61% dos registros como não confiáveis, e apresentou os seguintes indicadores, 100% de precisão, 5,62% de sensibilidade, 100% de especificidade, 10,64% de *F-score* e 19,93% de MCC.

5. Conclusão e Trabalhos Futuros

No contexto de transporte público, um dos problemas existentes é a pontualidade dos ônibus em relação aos horários de parada pré estabelecidos. A integração eficiente das diversas tecnologias existentes nas SC tem permitido a criação e identificação de inúmeras oportunidades para solucionar (ou amenizar) problemas como o mencionado, que afetam a qualidade dos serviços de transporte disponibilizado para a população. Contudo, a predição dos tempos de chegadas de ônibus é uma atividade que depende de diversos fatores aleatórios (ambiente estocástico). Como consequência, a precisão das predições realizadas pode ser prejudicada, uma vez que os erros de predição são ocasionados por fatores que não podem ser controlados como, por exemplo: atrasos em intersecções sinalizadas, número de passageiros em pontos de parada, etc.

A utilização de técnicas de *crowdsourcing* possibilita a obtenção de novos dados ou até mesmo a correção de informações existentes. No entanto, assim como em outras abordagens, dados de *crowdsourcing* também apresentam informações não confiáveis que podem comprometer a confiabilidade das atividades de predições realizadas. Este trabalho investigou o uso de aprendizado supervisionado para analisar a confiabilidade dos dados obtidos por *crowdsourcing*. Analisamos o desempenho de 6 algoritmos de ML em termos

de eficiência computacional e em qualidade de classificação. A análise dos resultados obtidos mostram que os modelos desenvolvidos utilizando os algoritmos de Árvores de Decisão, *Gaussian Naive Bayes* e Regressão Logística foram os mais adequados para análise de dados de mobilidade urbana. Acreditamos que a qualidade da classificação e o bom desempenho computacional apresentado pelos algoritmos os tornariam candidatos à serem utilizados na prática.

Como trabalhos futuros, pretendemos analisar a confiabilidade de outras formas de obtenção de dados como, por exemplo, as informações de mídias sociais e analisar o uso desses algoritmos na análise de fluxos de dados obtidos em tempo real.

6. Agradecimentos

Os autores agradecem à empresa Scipopulis pelos dados fornecidos. Esta pesquisa é parte do INCT da Internet do Futuro para Cidades Inteligentes financiado pelo CNPq, proc. 465446/2014-0, CAPES, proc. 88887.136422/2017-00 e FAPESP, proc. 2014/50937-1.

Referências

Allahbakhsh, M., Benatallah, B., Ignjatovic, A., Motahari-Nezhad, H. R., Bertino, E., and Dustdar, S. (2013). Quality control in crowdsourcing systems: Issues and directions. *IEEE Internet Computing*, 17(2):76–81.

Altinkaya, M. and Zontul, M. (2013). Urban bus arrival time prediction: A review of computational models. *International Journal of Recent Technology and Engineering*, 2(4):164–169.

Batista, D. M., Goldman, A., Hirata, R., Kon, F., Costa, F. M., and Endler, M. (2016). Interscity: Addressing future internet research challenges for smart cities. In *7th International Conference on the Network of the Future (NOF)*, pages 1–6.

Biagioni, J., Gerlich, T., Merrifield, T., and Eriksson, J. (2011). Easytracker: automatic transit tracking, mapping, and arrival time prediction using smartphones. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pages 68–81. ACM.

Bishop, C. M. (2006). Pattern recognition and machine learning (information science and statistics) springer-verlag new york. *Inc. Secaucus, NJ, USA*.

Caprotti, F. (2016). *Eco-cities and the transition to low carbon economies*. Springer.

Cerotti, D., Distefano, S., Merlino, G., and Puliafito, A. (2016). A crowd-cooperative approach for intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*.

Chowdhury, M. A. and Sadek, A. W. (2003). *Fundamentals of intelligent transportation systems planning*. Artech House.

Cullina, E., Conboy, K., and Morgan, L. (2015). Measuring the crowd: a preliminary taxonomy of crowdsourcing metrics. In *Proceedings of the 11th International Symposium on Open Collaboration*, page 7. ACM.

Cury, M. J. F. and Marques, J. A. L. F. (2016). A cidade inteligente: uma reterritorialização/smart city: A reterritorialization. *Redes*, 22(1).

Daniel, F., Kucherbaev, P., Cappiello, C., Benatallah, B., and Allahbakhsh, M. (2018). Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Comput. Surv.*, 51(1):7:1–7:40.

De Santis, R., Fasano, A., Mignolli, N., and Villa, A. (2014). Smart city: fact and fiction. *Munich Personal RePEc Archive*.

Figueiredo, G. D. S. (2016). Técnicas de simulação para apoio à decisão em planejamento urbano. Master's thesis, Universidade Federal do Rio de Janeiro.

Figueiredo, L. M. B. (2005). *Sistemas inteligentes de transporte*. PhD thesis, Univ. do Porto.

Ghotra, B., McIntosh, S., and Hassan, A. E. (2015). Revisiting the impact of classification techniques on the performance of defect prediction models. In *Proceedings of the 37th International Conference on Software Engineering-Volume 1*, pages 789–800. IEEE Press.

Hair, J. F., Black, W. C., Babin, B. J., Anderson, R. E., and Tatham, R. L. (2009). *Análise multivariada de dados*. Bookman Editora.

Hosmer Jr, D. W., Lemeshow, S., and Sturdivant, R. X. (2013). *Applied logistic regression*, volume 398. John Wiley & Sons.

Howe, J. (2006). The rise of crowdsourcing. *Wired magazine*, 14(6):1–4.

Hwang, M., Kemp, J., Lerner-Lam, E., Neuerburg, N., and Okunieff, P. (2006). Advanced public transportation systems: state of the art update 2006. Technical Report FTA-NJ-26-7062-06.1, U.S. Department of Transportation.

Kohavi, R. (1995). The power of decision tables. In *European conference on machine learning*, pages 174–189. Springer.

Kormáksson, M., Barbosa, L., Vieira, M. R., and Zadrozny, B. (2014). Bus travel time predictions using additive models. In *IEEE Intl. Conference on Data Mining*, pages 875–880. IEEE.

Magalhães, C. T. d. A. (2008). Avaliação de tecnologias de rastreamento por gps para o monitoramento do transporte público por ônibus. Master's thesis, Univ. Federal do Rio de Janeiro.

Mashhadi, A. J. and Capra, L. (2011). Quality control for real-time ubiquitous crowdsourcing. In *Proceedings of the 2Nd International Workshop on Ubiquitous Crowdsouring*, UbiCrowd '11, pages 5–8, New York, NY, USA. ACM.

Misra, A., Gooze, A., Watkins, K., Asad, M., and Le Dantec, C. (2014). Crowdsourcing and its application to transportation data collection and management. *Transportation Research Record: Journal of the Transportation Research Board*, 2414:1–8.

Mitchell, T. M. (1997). Machine learning. 1997. *Burr Ridge, IL: McGraw Hill*, 45(37):870–877.

Mousa, H., Mokhtar, S. B., Hasan, O., Younes, O., Hadhoud, M., and Brunie, L. (2015). Trust management and reputation systems in mobile participatory sensing applications. *Comput. Netw.*, 90(C):49–73.

Pedersen, J., Kocsis, D., Tripathi, A., Tarrell, A., Weerakoon, A., Tahmasbi, N., Xiong, J., Deng, W., Oh, O., and de Vreede, G.-J. (2013). Conceptual foundations of crowdsourcing: A review of is research. In *Hawaii International Conference on System Sciences*. IEEE.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pons, I., Monteiro, J., and Speicys Cardoso, R. (2015). Big data para análise de métricas de qualidade de transporte: metodologia e aplicação. Technical report, ANTP.

Sussman, J. S. (2008). *Perspectives on intelligent transportation systems (ITS)*. Springer.

Vapnik, V. N. (1998). *Statistical learning theory*, volume 1. Wiley New York.

Wang, J.-n., Chen, X.-m., and Guo, S.-x. (2009). Bus travel time prediction model with ν-support vector regression. In *Intelligent Transportation Systems, 2009. ITSC'09. 12th Intern. IEEE Conf. on*, pages 1–6. IEEE.

Witten, I. H., Frank, E., Hall, M. A., and Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

Wu, C.-H., Ho, J.-M., and Lee, D.-T. (2004). Travel-time prediction with support vector regression. *IEEE transactions on intelligent transportation systems*, 5(4):276–281.

Zhu, T., Ma, F., Ma, T., and Li, C. (2011). The prediction of bus arrival time using global positioning system data and dynamic traffic information. In *Wireless and Mobile Networking Conference (WMNC), 2011 4th Joint IFIP*, pages 1–5. IEEE.

Um Ambiente de Apoio à Decisão baseado em *Data Warehouse* para a Área de Segurança Pública do Estado do Rio de Janeiro

Wagner Santos[1], **Daniel de Oliveira**[2]

[1]Polícia Militar do Estado do Rio de Janeiro (PMERJ)

[2]Instituto de Computação – Universidade Federal Fluminense (IC/UFF)

pesquisa_ei@pmerj.rj.gov.br, danielcmo@ic.uff.br

Resumo. *As novas abordagens propostas para cidades inteligentes possuem o potencial de melhorar a vida dos cidadãos em diversos aspectos. Um desses aspectos é o aprimoramento das políticas de segurança pública, em especial nos grandes centros metropolitanos. A análise integrada de dados históricos de ocorrências de crimes ajuda, sem dúvida, na tomada de decisão, na previsão e até mesmo na prevenção de ocorrências. No entanto, o acesso a esses dados históricos de forma integrada pode não ser trivial, pois se faz necessário integrar bases de dados de diferentes organizações (e.g., bombeiros, polícia civil, polícia militar, etc). Neste artigo, apresentamos uma abordagem que visa integrar dados das diferentes organizações associadas à segurança pública de forma a prover um ambiente de apoio a decisão que possa de fato auxiliar especialistas em segurança a delinear suas ações. O arcabouço proposto é denominado CHORD (Criminal dasHbOaRd Decision making). Nesse artigo, instanciamos o CHORD para o cenário do estado do Rio de Janeiro, que vem sofrendo nos últimos anos com um aumento significativo nos índices de criminalidade. A abordagem proposta é uma ferramenta promissora para auxiliar os departamentos de polícia na prevenção de crimes.*

1. Introdução

O conceito de *Cidades Inteligentes* (*i.e., Smart Cities*) tem ganhado bastante relevância na comunidade acadêmica nos últimos anos. Diversas iniciativas de sucesso podem ser encontradas tanto no Brasil quanto fora dele, como por exemplo o programa MIT *City Science*[1]. As Cidades Inteligentes podem ser definidas como sistemas complexos que envolvem pessoas com uma variedade de *expertises*, interagindo e utilizando um conjunto de serviços e ambientes para melhorar o desenvolvimento da cidade e, consequentemente, a qualidade de vida dos seus cidadãos [Shapiro 2006, Allwinkle and Cruickshank 2011].

De acordo com a Fundação Getúlio Vargas (FGV)[2], podemos caracterizar o nível de inteligência de uma cidade de acordo com nove dimensões, a saber: governança, administração pública, planejamento urbano, tecnologia, meio ambiente, conexões internacionais, coesão social, capital humano e economia. Em especial, nesse artigo, nos concentramos em uma dimensão específica: a administração pública, e mais detalhadamente, em políticas de segurança na administração pública.

Cidades inteligentes são, *a priori*, cidades seguras. Dessa forma, fornecer soluções para reduzir a criminalidade urbana é uma das principais prioridades. A criminalidade urbana é uma questão antiga, ainda que seja um problema em aberto, em muitos países, em especial o Brasil [Baldwin 1975, Sociales 2001, Gribanova et al. 2017]. O problema do aumento da criminalidade tem afetado praticamente todas as cidades brasileiras não distinguindo

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

[2]http://fgvprojetos.fgv.br/noticias/o-que-e-uma-cidade- Inteligente

etnia, sexo ou classe social. A crescente emigração para as grandes cidades em busca de oportunidades de emprego, alavancada pela crise econômica nos últimos anos, tem gerado um aumento significativo dos índices criminais nos grandes centros urbanos. De acordo com o Instituto de Segurança Pública do estado do Rio de Janeiro[3], os roubos de rua têm seguido uma trajetória inconstante, mas ascendente, no período entre janeiro de 2010 e dezembro de 2017. O mesmo vêm acontecendo com homicídios no mesmo período [4], o que corrobora com o estudo realizado pelo Conselho Cidadão para a Segurança Pública e Justiça, o Brasil ocupa o décimo lugar das cidades mais violentas do mundo [5].

Tais fatos têm levado diversas autoridades da segurança pública – nos diversos níveis governamentais – e a sociedade civil a refletirem sobre o desenvolvimento de ferramentas para criação de políticas públicas eficazes com o propósito de reduzir a violência. Uma das soluções propostas pela Polícia Militar do Estado do Rio de Janeiro foi a criação do Escritório de Gestão da Qualidade (EGQ). O EGQ possui o intuito de estudar as ocorrências de crimes a fim de desenvolver análises complexas capazes de oferecer subsídios aos gestores no momento da tomada de decisão. Entretanto, o EGQ ainda não possui um ambiente analítico que seja capaz de integrar dados das diversas esferas governamentais de forma a apoiar o processo de tomada de decisão para elaboração de políticas públicas.

Nas últimas décadas, uma abordagem capaz de prover tal capacidade analítica, chamada de *Data Warehouse* (DW), tem sido amplamente utilizada em diversos domínios [Golfarelli and Rizzi 2009], sejam eles acadêmicos ou comerciais [Inmon 1992]. DWs são bases de dados multidimensionais que integram informações de diversas fontes a fim de facilitar a análise de dados, reunindo e consolidando informações de diversos *Data Marts* (DM) [Inmon 1992]. Estes são um subconjunto dos DWs que possuem um objetivo específico, orientado por assunto, variante no tempo, e não volátil. Uma das maiores vantagens dos DMs frente aos bancos de dados transacionais é que eles possuem dados previamente sumarizados, *e.g.,* dados agregados por mês ou ano [Inmon 1992], o que acelera e facilita o processo de análise dos dados.

Dessa forma, o objetivo desse artigo é propor um ambiente de apoio à tomada de decisão baseado em *Data Warehouse* denominado CHORD (Criminal dasHbOaRd Decision making) que seja capaz de integrar informações de várias esferas governamentais e que possibilite a sumarização dos indicadores estratégicos criminais para análise pelos membros do EGQ. Além disso, tal ambiente tem como objetivo estar alinhado com o Sistema Integrado de Metas do estado do Rio de Janeiro, sendo capaz de estabelecer uma comparação entre os valores reais dos indicadores de crimes e suas metas pré-estabelecidas pela Secretaria de Segurança Pública.

Esse artigo se encontra organizado em 5 seções além da Introdução. A Seção 2 apresenta o referencial teórico. A Seção 3 apresenta a abordagem proposta chamada CHORD. A Seção 4 apresenta a avaliação experimental. A Seção 5 discute trabalhos relacionados, e, finalmente, a Seção 6 conclui o artigo e discute trabalhos futuros.

2. Referencial Teórico

Esta seção apresenta os principais conceitos associados a abordagem proposta tanto no que tange conceitos de segurança pública quanto de *Data Warehouse*.

[3]http://www.isp.rj.gov.br

[4]https://www.ispgeo.rj.gov.br

[5]http://www.seguridadjusticiaypaz.org.mx/biblioteca/prensa/send/6-prensa/239-las-50-ciudades-mas-violentas-del-mundo-201

2.1. O Sistema Integrado de Metas do Estado do Rio de Janeiro

O Sistema Integrado de Metas (SIM) foi desenvolvido pela Subsecretaria de Planejamento e Integração Operacional (SSPIO), órgão subordinado a Secretária Estadual de Segurança Pública do estado do Rio de Janeiro - SESEG/RJ, sendo publicado no Decreto Estadual nº 41.931 de 26 de novembro de 2009.

Este sistema tem como objetivo desencadear ações integradas entre as Polícias Civil e Militar com o propósito de prevenir e controlar a ocorrência de crimes do estado do Rio de Janeiro por meio do controle de indicadores estratégicos criminais. Esses indicadores agrupam delitos em categorias, são elas:

- *Letalidade Violenta*: composto pelos crimes de Homicídio Doloso, Latrocínio (roubos seguidos de morte), Lesão Corporal Seguida de Morte e Homicídio Decorrente de Oposição à Intervenção Policial;
- *Roubo de Veículo*: Composto pelos crimes de roubo de veículo e roubo de moto;
- *Roubo de Rua*: composto pelos crimes de Roubo a Transeunte, Roubo em Coletivo e Roubo de Aparelho Celular;

Ainda segundo o decreto, para melhorar o controle dos indicadores estratégicos criminais, o estado foi dividido em 137 Circunscrições Integradas de Segurança Pública (CISP), com numerações não sequencias agrupadas em 39 Áreas Integradas de Segurança Pública (AISP), que por sua vez estão agrupadas dentro de sete Regiões Integradas de Segurança Pública (RISP). A Figura 1 apresenta um exemplo da organização do estado do Rio de Janeiro para controle dos indicadores estratégicos.

Figura 1. Divisão Territorial do Estado do Rio de Janeiro em RISP, AISP e CISP - Fonte: www.sistemademetas.rj.gov.br

Dessa forma, as polícias Civil e Militar adaptaram seus departamentos e comandos a fim de criar equivalências em suas estruturas especiais e a estrutura especial da Secretaria de Segurança Pública. No que tange a Polícia Militar do estado do Rio de Janeiro, esta se organizou da seguinte forma: (i) Criou sete Comandos de Policiamento de Área (CPA), cada um sendo equivalente a uma RISP; (ii) Criando 39 Batalhão de Polícia Militar agrupados em CPA, sendo cada batalhão equivalente uma AISP; e (iii) Cada batalhão possui em sua área de policiamento uma quantidade de delegacias, sendo que cada delegacia equivalente a uma CISP.

2.2. Uma Breve Introdução à Análise Criminal

Com os indicadores estratégicos, o Escritório de Gestão da Qualidade realiza o monitoramento utilizando técnicas oriundas de diversas áreas de conhecimento, denominada

Análise Criminal (AC). A AC serve como fonte para a confecção dos relatórios gerenciais que oferecem apoio às tomadas de decisão. De acordo com [Magalhães 2007], a AC possui três vertentes (Estratégica – Tática – Administrativa), sendo dessa forma o maior vetor de produção de conhecimento específico para a Gestão da Segurança Pública, possibilitando revelar com clareza as características do crime a partir de eventos desconexos.

A Análise Criminal Administrativa (ACD) refere-se à aplicações de técnicas estatísticas descritivas que mostram o comportamento dos indicadores ao longo de um determinado período (*i.e.* série histórica). A realização de comparação dentre períodos sazonais tem como objetivo a verificação da variação dos indicadores, além da produção do conhecimento voltada a diversos públicos (Cidadãos, Gestores Públicos, Instituições Pública, Organismos Internacionais, Organizações Não-Governamentais, etc.) devidamente selecionados pelos gestores.

A Análise Criminal Tática (ATC) produz conhecimento a médio prazo que são utilizados nas atividades de polícia ostensiva e investigativa. Na atividade de polícia ostensiva são produzidos relatórios que indicarão as áreas de maior incidência dos delitos e, por vezes, indicando os *modi operandi* dos autores dos delitos, bem como o perfil dos envolvidos (vítima e autor) e a correlação entre eles. No que tange a atividade investigativa, o mesmo conhecimento agregado de mais informações sobre os envolvidos (vítima e autor) possibilita a busca da solução dos casos investigados.

A Análise Criminal Estratégica (ACE) diz respeito à produção do conhecimento tendo como direcionamento o estudo dos fenômenos criminais a longo prazo. Nesta vertente, os relatórios servem como base para o planejamento e desenvolvimento de soluções capazes de produzir políticas públicas por meio da interação entre instituições que possuem influência na segurança pública.

Em todas as vertentes da análise criminal, o analista realiza inicialmente o processo de extração de dados oriundo de um ambiente transacional (OLTP), onde os usuários executam várias tarefas, tais como: inclusões, alterações, exclusões e pesquisas, além de dados oriundos de planilhas eletrônicas, textos, mapas entre outros tipos de arquivos.

Em um terceiro momento, o analista realiza a inferência fundamentada na aplicação de equações estatísticas e probabilísticas, bem como o uso de técnicas de geoprocessamento a fim de responder as perguntas (”O que?”, ”Onde?”, ”Quando?”, ”Quem?”e ”Como?”) necessárias ao encontro da solução para os eventos criminosos.

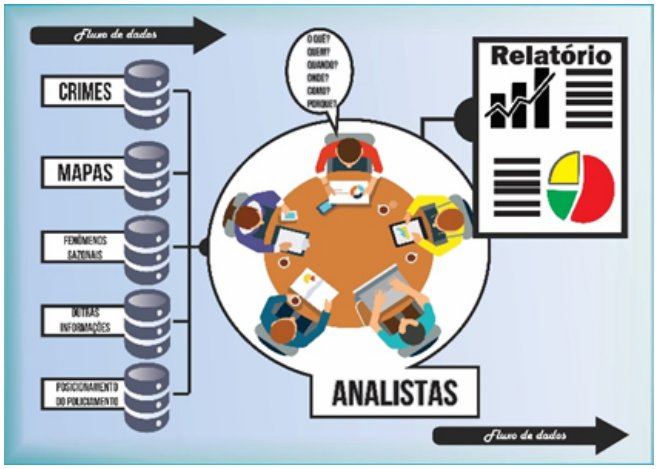


Figura 2. Fluxo de Dados no Processo de Análise Criminal

A última etapa é a construção dos relatórios e/ou *dashboards*. Neles são expostos os resultados encontrados nas análises realizadas pelos analistas utilizando técnicas básicas de visualização de dados (*i.e.* exposição de gráficos com métricas estatísticas como: contagens, totais, diferenças absolutas, diferenças percentuais, frequências acumuladas e percentuais acumulados). A Figura 2 apresenta o fluxo dos dados ao longo do processo de análise.

2.3. *Data Warehousing* e Modelagem Dimensional

Segundo [Kimball and Ross 2002], a modelagem dimensional é uma técnica de projeto de bancos de dados que visa apoiar consultas analíticas. Faz-se uso de redundâncias planejadas dos dados para melhorar o desempenho das consultas [Kimball and Ross 2002, Inmon 1992]. O modelo dimensional de um banco de dados é composto pelas tabelas *Fato* com suas respectivas *Dimensões*. As dimensões podem ser compartilhadas por tabelas fato diferentes. Existem dois modelos de implementação e um banco de dados dimensional: o Modelo Estrela [Kimball and Ross 2002] e o Modelo Floco de Neve [Inmon 1992]. O Modelo Estrela possui a tabela fato centralizada com as suas respectivas dimensões no seu entorno. Nesse modelo, a tabela fato possui chaves estrangeiras para todas as suas dimensões, sendo um modelo desnormalizado. O Modelo Floco de Neve é uma variação do Modelo Estrela, no qual todas as dimensões são normalizadas, fazendo com que sejam geradas quebras na tabela original ao longo de hierarquias existentes em seus atributos.

Um Data Warehouse (DW) é constituído pela união dos Data Marts (DMs). Assim, como nos DMs, um DW preferencialmente deve ser modelado de forma dimensional, pois em comparação com um banco de dados transacional e normalizado, a modelagem dimensional produz modelos mais previsíveis e compreensíveis, facilitando a utilização e assimilação pelos usuários finais (no contexto desse artigo, os analistas de segurança), além de possibilitar consultas com alto desempenho [Kimball and Ross 2002]. Portanto, a modelagem dimensional possui uma estrutura simplificada, mais próxima da visão que o físico tem do seu domínio, facilitando assim a compreensão, de forma que os próprios físicos possam criar suas consultas. Apesar de terem uma estrutura diferente de bancos de dados transacionais, os bancos de dados dimensionais podem ser modelados sobre Sistemas de Gerência de Bancos de Dados (SGBDs) relacionais como o mySQL ou o PostsgreSQL.

Segundo [Raslan and Calazans 2014] antes que os dados sejam sumarizados e inseridos no DW, eles passam por um processo denominado ETL (*Exract*, *Transform*, *Load*). A extração envolve a leitura e compreensão dos dados de origem e cópia destes para a *staging area* para serem manipulados posteriormente. Na fase de transformação há o processo de filtragem dos dados, correção de possíveis erros de digitação, tratamento de conteúdos ausentes de atributos, a combinação de dados de diversas origens, exclusão de atributos que não fazem parte do domínio, além da criação de novas chaves (*surrogate keys*). A carga de dados (*load*) é o processo de inserção dos dados no DW após o processo de transformação, sendo portando, um processo interativo e incremental ao longo do tempo. O ETL apresenta os mesmos processos realizados manualmente pelos analistas do EGQ atualmente.

3. Abordagem Proposta: CHORD

O CHORD (Criminal dasHbOaRd Decision making) é um ambiente de apoio a decisão baseado em DW para o Escritório de Gestão da Qualidade (EGQ). Utilizando um ambiente que integra dados de diferentes esferas governamentais, os especialistas em segurança do EGQ são capazes de delinear políticas de segurança mais eficazes. A arquitetura do CHORD é apresentada na Figura 3. O CHORD é composto por cinco componentes principais: (i) as fontes de dados, (ii) o componente ETL, (iii) o ODS (*Operational Data Store*), (iv) o *Data*

Warehouse CHORD-DW, que é composto por vários *Data Marts*, e (v) o *Dashboard*. A seguir discutiremos cada um dos componentes do CHORD.

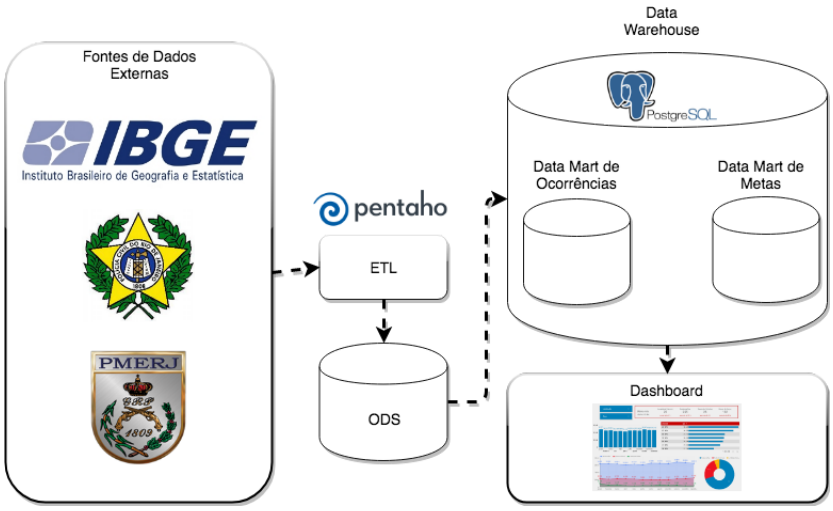


Figura 3. Arquitetura Conceitual do CHORD

3.1. Fontes de Dados

As fontes de dados representam bases de dados de diversas organizações que são utilizadas para alimentar o CHORD-DW. A principal fonte de informações para o CHORD-DW deriva do Sistema Gerencial *Web* (Gweb). O Gweb é um ambiente transacional que armazena diariamente as informações dos registros nas delegacias de Polícia Civil, sendo possível extrair do sistema as principais informações de crimes que compõem o Sistema Integrado de Metas além de todas as variantes de ocorrências intituladas como roubo. No Gweb há a opção de extração de dados para arquivos no formato .CSV com estrutura predefinida. Além dos dados do Gweb, o CHORD também importa dados das metas que devem ser atingidas dentro de um período de tempo por todos os gestores. As metas são publicadas semestralmente no diário oficial do estado do Rio de Janeiro, além de estarem disponíveis no sítio do Instituto de Segurança Pública/RJ. Outra fonte importante de dados é a base de dados do IBGE de onde são importadas informações sobre a população residente, a população flutuante da região, etc.

3.2. O Componente ETL

O componente ETL é o responsável por executar todo o processo de extração de informações e carga no CHORD-DW. O processo se inicia identificando as origens dos dados que servirão como base para compor o *Operational Data Store* (ODS), tais como: a relação de comandantes das Áreas Integradas de Segurança Pública, a relação dos indicadores que serão monitorados, os títulos de crimes que os compõem, a origem das ocorrências de crimes e o mapa da área em que as ocorrências serão monitoradas. Essas informações chegam até os analistas por meio de planilhas eletrônicas extraídas do Gweb. A ferramenta escolhida para realizar o processo de ETL foi o Pentaho, uma vez que a suíte Pentaho é uma ferramenta livre que está em expansão na comunidade de *Business Intelligence*. O Pentaho verifica se existem atualizações ou novas informações a serem importadas das múltiplas fontes de dados, e, caso haja, o mesmo realiza o processo de carga no ODS. O *ODS* é uma base de dados intermediária que contém dados das várias fontes. Diferentemente do CHORD-DW, o ODS não tem o objetivo de armazenar o histórico de dados e de servir como base para consultas mais complexas.

3.3. Construção dos *Data Marts* do CHORD-DW

O CHORD-DW é um componente-chave do CHORD, uma vez que todo o processo analítico depende dos dados que são integrados. Nessa sub-seção apresentamos como os DMs que compõem o CHORD-DW foram modelados. [Monteiro et al. 2013] afirma que os DWs não necessitam ser construídos de uma vez, pois a complexidade se torna bastante elevada. Em vez disso, sugere-se abordar processos de negócios de maneira incremental, e, assim, conceber o DW em partes. Neste sentido, optou-se em construir o DW utilizando a metodologia *bottom-up*, onde a construção do CHORD-DW começa a partir de uma pequena parte (assunto), *i.e.* DMs. A seguir são descritos os processos de construção dos DMs para o acompanhamento dos crimes ocorridos no estado do Rio de Janeiro, bem como a comparação com as metas estabelecidas para os indicadores.

A Figura 4 apresenta o Modelo CHORD-DW. Foi utilizada a modelagem dimensional estrela [Kimball and Ross 2002], sendo a tabela fato de ocorrências de crimes representada pela tabela *fatoOcorrencia*, responsável por armazenar os fatos, que no contexto desse artigo são as ocorrências de crimes, armazenados, respectivamente, nos atributos *valorLetalidadeViolenta*, *valorRouboVeiculo* e *valorRouboRua*. Ainda na tabela *fatoOcorrencia*, os atributos *idArea*, *idTempo* e *idTitulo* são chaves estrangeiras para as tabelas que representam as dimensões tempo, área e título, respectivamente.

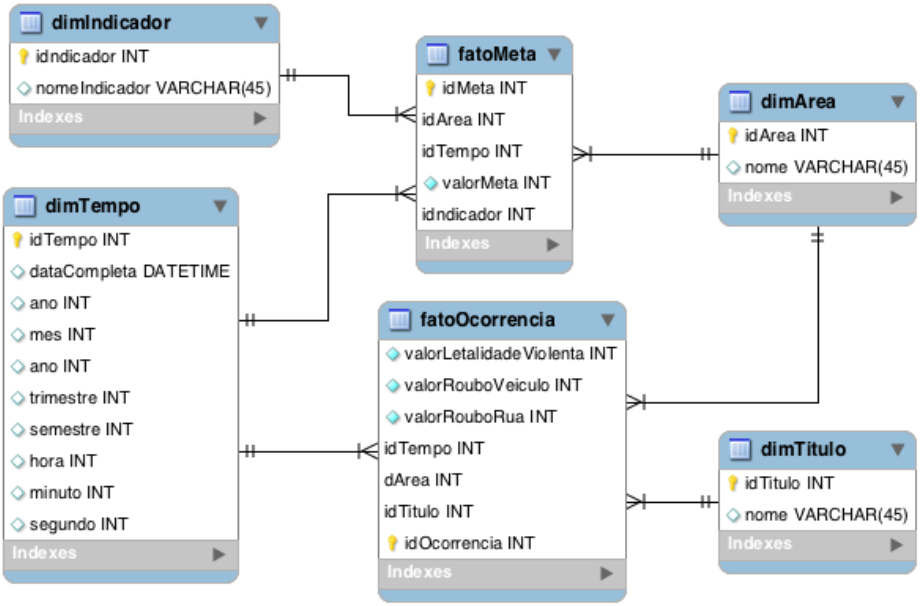


Figura 4. Modelagem do *Data Warehouse* do CHORD

A tabela fato que corresponde às metas que devem ser cumpridas por cada área se encontra representada pela tabela *fatoMeta*, responsável por armazenar as metas de cada área ao longo do tempo. Os valores das metas são armazenados no atributo *valorMeta*. Ainda na tabela *fatoMeta*, os atributos *idArea*, *idIndicador* e *idTempo* são chaves estrangeiras para as tabelas que representam as dimensões tempo, indicador e área, respectivamente.

A dimensão área *dimArea* agrupa todas as divisões e subdivisões do estado do Rio de Janeiro que estão previstas no Decreto Estadual nº 41.931 de 26 de novembro de 2009. Com exceção da Circunscrição Integrada de Segurança Pública, todas as divisões possuem um comandante da Polícia Militar. A tabela *dimTempo* representa a dimensão tempo, sendo responsável por armazenar todas as possíveis granularidades de tempo para um determinado

fato. O atributo *dataCompleta* representa a data completa (até milissegundos). Os atributos *ano, mes, dia, trimestre, semestre, hora, minuto* e *segundo* são as representações numéricas de partes da data completa. O atributo *idTempo* é a chave primária da tabela. A tabela *dimIndicador* associa a qual indicador um determinado crime está associado e possui o atributo *nomeIndicador*. A tabela *dimTitulo* representa os títulos de crimes que estão impactando diretamente os números do indicador em que pertence.

4. Avaliação Experimental

De forma a avaliar o CHORD, realizamos uma avaliação experimental com uma amostra de ocorrências de crimes ocorridos no estado do Rio de Janeiro durante os anos 2015, 2016 e 2017. A tabela *fatoOcorrencia* possui aproximadamente 500.000 registros. A Tabela 1 exibe três consultas e seus tempos de execução associados (por limitações de espaço apresentamos apenas três consultas). Tal medição tem como objetivo demonstrar a eficiência adquirida em relação ao banco de dados transacional existente no EGQ. Para cada consulta apresentamos seu tempo de execução em milissegundos obtido por meio da sua execução via *Dashboard* do CHORD, apresentado na Figura 5.

Tabela 1. Desempenho das Consultas no CHORD

Consulta	Tempo de Execução
Consultar o somatório dos valores dos índices de criminalidade durante todos os anos	7.678 ms
Consultar o somatório dos valores dos índices de criminalidade durante o ano de 2017	3.144 ms
Consultar o somatório dos valores dos índices de criminalidade durante o ano de 2016 no 18^o BPM	2.432 ms

Na primeira consulta foi explorada a agregação dos valores dos campos *valorLetalidadeViolenta*, *valorRouboVeiculo* e *valorRouboRua* em todos os anos. A segunda consulta explora agregação dos valores dos campos *valorLetalidadeViolenta*, *valorRouboVeiculo* e *valorRouboRua* durante o ano de 2017. A terceira consulta explora agregação dos valores dos campos *valorLetalidadeViolenta*, *valorRouboVeiculo* e *valorRouboRua* durante o ano de 2016 no 18^o Batalhão de Polícia, que compreende os bairros de Jacarepaguá, Pechincha, Freguesia, Tanque, Vila Valqueire, Taquara, Curicica, Cidade de Deus, Anil e Gardênia Azul. Todas as consultas se beneficiam de agregações pré-calculadas presentes no CHORD-DW e, por isso, apresentam um tempo de execução bastante reduzido. É importante ressaltar que a amostra contém apenas três anos, e os tempos de execução tendem a aumentar quando todos os dados estiverem carregados no CHORD-DW. Este experimento reforça a necessidade de se utilizar agregações pré-calculadas neste cenário.

5. Trabalhos Relacionados

A segurança pública é uma prioridade absoluta tanto para a indústria quanto para a comunidade científica, principalmente devido aos benefícios potenciais que as abordagens propostas nessa área podem proporcionar à sociedade e aos seus cidadãos.

A abordagem mais proeminente nesse sentido é o $PredPol$[6], um *software* comercial usado pelo Departamento de Polícia de Atlanta. O $PredPol$ considera a história dos crimes para prever a incidência de novos crimes na cidade. Apesar de seu aparente sucesso, já que

[6]http://www.predpol.com/

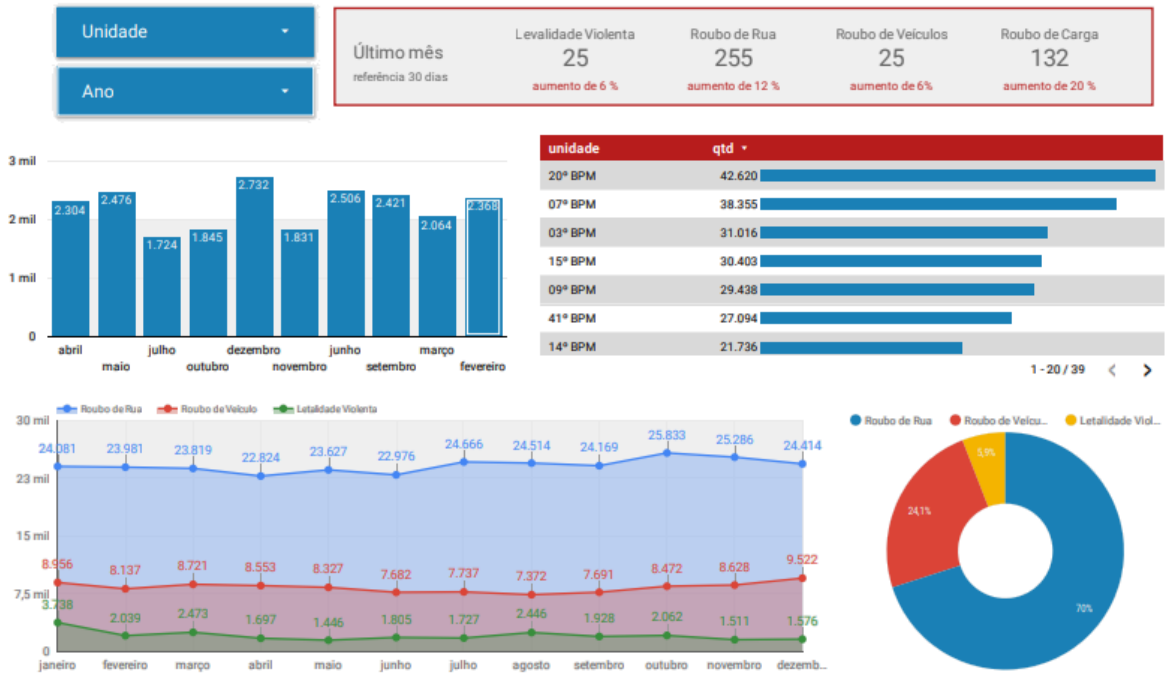


Figura 5. Um fragmento do *Dashboard* do CHORD

a utilização do sistema foi capaz de ajudar na redução de 32% dos arrombamentos naquela cidade, não está claro como o sistema armazena seus dados ou se integra diferentes bases de dados.

Outras iniciativas mais simples usadas pelos departamentos de polícia em todo o mundo incluem os mapas de crime fornecidos pela Polícia do Reino Unido[7] e do departamento de Polícia de Nova York[8]. No Brasil, podemos citar o portal ”Onde fui Roubado”[Roubado 2013] que é uma abordagem *crowdsourcing* para coleta e visualização de ocorrências de crimes na *Web*. No entanto, tais abordagens apresentam apenas ocorrências criminais em mapas visuais, sem que seja realizado o cruzamento de tais dados com outras bases de dados.

6. Conclusões e Trabalhos Futuros

Infelizmente, nos últimos anos, os cidadãos dos grandes centros urbanos, em especial do estado do Rio de Janeiro, sofreram com ondas de violência significativamente maiores. Ambientes e soluções que fornecem capacidade analítica aos gestores têm o potencial de fornecer soluções para aliviar este problema.

Neste artigo, demos um passo nessa direção, propondo o ambiente CHORD (Criminal dasHbOaRd Decision making). O CHORD é baseado no conceito de DW que integra dados de bancos de dados de diversas esferas governamentais. O CHORD-DW, que é o componente-chave do ambiente CHORD, segue uma modelagem dimensional do tipo estrela [Kimball and Ross 2002] que permite que dados sejam armazenados de forma pré-calculada no CHORD-DW, acelerando assim consultas que eram bastante lentas anteriormente. Foi realizada uma avaliação experimental do CHORD utilizando-se um sub-conjunto dos dados

[7]https://www.police.uk
[8](https://maps.nyc.gov /crime/)

de ocorrencias de crimes nos anos de 2015, 2016 e 2017 (aproximadamente 500.000 registros), e constatou-se que o tempo de execução de todas as consulta é aceitável.

Como sugestão de trabalho futuro, podemos considerar a identificação de padrões nos dados do CHORD, utilizando-se de técnicas de mineração de dados e aprendizado de máquina. A aplicação de técnicas de mineração de dados possibilita a indução de padrões interpretáveis que podem auxiliar os departamentos de polícia na prevenção de crimes.

Agradecimentos

Os autores agradecem a CNPq, CAPES e FAPERJ por financiarem parcialmente o trabalho aqui apresentado.

Referências

Allwinkle, S. and Cruickshank, P. (2011). Creating smart-er cities: An overview. *Journal of urban technology*, 18(2):1–16.

Baldwin, J. (1975). Urban criminality and the 'problem'estate. *Local Government Studies*, 1(4):12–20.

Golfarelli, M. and Rizzi, S. (2009). *Data Warehouse Design: Modern Principles and Methodologies*. McGraw-Hill, Inc., New York, NY, USA, 1 edition.

Gribanova, G., Vulfovich, R., et al. (2017). Modern city safety as a complex problem. *Public administration issues*, (5):83–100.

Inmon, W. H. (1992). *Building the Data Warehouse*. John Wiley & Sons, Inc., New York, NY, USA.

Kimball, R. and Ross, M. (2002). *The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling*. John Wiley & Sons, Inc., New York, NY, USA, 2nd edition.

Magalhães, L. C. (2007). Análise criminal e mapeamento da criminalidade–gis. *Anais do Fórum Internacional de Gabinetes de Gestão Integrada. São Luís, Maranhão*.

Monteiro, A. V. G., Pinto, M. P. O., and da Costa, R. M. E. M. (2013). Uma aplicação de data warehouse para apoiar negócios. *Cadernos do IME-Série Informática*, 16:48–58.

Raslan, D. A. and Calazans, A. T. S. (2014). Data warehouse: conceitos e aplicações. *Universitas: Gestão e TI*, 4(1).

Roubado, O. F. (2013). Disponível em: http://ondefuiroubado. com. br. *Acessado em*, 27.

Shapiro, J. M. (2006). Smart cities: quality of life, productivity, and the growth effects of human capital. *The review of economics and statistics*, 88(2):324–335.

Sociales, P. (2001). Crime as a social cost of poverty and inequality: a review focusing on developing countries. *Facets of Globalization*, page 171.

Plataforma ROTA: Histórico, Desafios e Soluções para Segurança Pública em Cidades Inteligentes

Gustavo A. Carvalho, Pedro P. Barbosa Neto, Nélio Cacho, Eiji Adachi, Frederico Lopes

Universidade Federal do Rio Grande do Norte (UFRN)
Natal-RN, Brasil

gustavo_carvalho@ufrn.edu.br, pedro_paivaneto@hotmail.com,
neliocacho@dimap.ufrn.br, eijiadachi@imd.ufrn.br, fred@imd.ufrn.br

Resumo. *Em meio ao problema da segurança pública existente no Brasil, esse artigo apresenta o histórico de desenvolvimento e implantação da plataforma ROTA, uma solução de cidades inteligentes criada com o objetivo de operar sobre essa problemática, discorrendo sobre suas aplicações e o propósito de cada uma delas. Os resultados obtidos pela introdução da solução são exibidos e interpretados, expondo os avanços causados pela plataforma. Por fim, são apresentados novos desafios que deverão ser enfrentados pelo ROTA de forma a alcançar o seu funcionamento ideal.*

Abstract. *Amidst the problem of public safety in Brazil, this paper presents a developement and deployment history of the ROTA Platform, a smart city solution created to address problem, besides discussing its applications and the purpose of each one. The results obtained from the introduction of the solution are shown and explained, thereby exposing advances caused by the platform. The paper also presents new challenges to be handled by ROTA for achieving its ideal behavior.*

1. Introdução

Os últimos anos têm visto um crescimento cada vez maior da população urbana ao redor do mundo. Desde 2009, a quantidade de indivíduos vivendo em zonas urbanas já ultrapassa a população que vive em zonais rurais [UNDESA 2015], e com o tempo esse fenômeno só tende a se intensificar. Estimativas dizem que até 2050 a população urbana terá alcançado o número de 6,4 bilhões de pessoas ao redor do mundo [Herald 2014].

O desenvolvimento acelerado das cidades traz consigo não só benefícios, mas também uma série de problemas relacionados com o crescimento acelerado e não-planejado das áreas urbanas. Entre esses problemas, está o do crescimento das taxas de criminalidade. A segurança pública no Brasil é uma questão recorrente quando se trata de áreas a serem melhoradas. De acordo com dados do Atlas da Violência[1], produzido pelo Instituto de Pesquisa Econômica Aplicada (IPEA) em parceria com o Fórum Brasileiro de Segurança Pública (FBSP), em 2005 a taxa de homicídios no Brasil era de 26,1 para cada 100 mil habitantes, aumentando em 2015 para aproximadamente 28,9. Esses números são aproximadamente quatro vezes superiores à média global. No estado do Rio Grande do Norte, de 2005 à 2015, a taxa de homicídios aumentou de 13,5 para 44,9, uma variação de aproximadamente +232,0%, a pior registrada dentre todos os estados brasileiros.

[1] http://ipea.gov.br/atlasviolencia/

Em meio a essa realidade, a Universidade Federal do Rio Grande do Norte (UFRN) junto à Secretaria da Segurança Pública e da Defesa Social do Estado do Rio Grande do Norte (SESED-RN) apresentam neste artigo o histórico, as soluções apresentadas e os desafios enfrentados durante o desenvolvimento da plataforma ROTA. A plataforma ROTA é uma solução relacionada ao conceito de cidades inteligentes que busca melhorar a segurança pública da cidade de Natal através da integração de diversas fontes de informação e da integração de novas tecnologias na infraestrutura de segurança já existente.

Esse artigo descreve o estado atual da plataforma ROTA e de suas aplicações, além de apresentar os principais resultados alcançados desde a sua implantação. Por fim, são discutidas tanto novas funcionalidades para a plataforma ROTA como também desafios que ainda podem ser enfrentados pela mesma.

2. Aplicações de Segurança para Cidades Inteligentes

A problemática da violência e da falta de segurança é uma que ocorre no mundo todo e da qual sempre se buscam novas soluções para tentar resolvê-la. No âmbito de cidades inteligentes, plataformas como INSPEC^{2}T [Leventakis et al. 2016], Polícia Popular[2] e Emergência RJ[3] vêm ganhando notoriedade por permitirem o registro de denúncias e ocorrências diretamente a partir de qualquer smartphone, reduzindo a "carga" das linhas diretas com a polícia, permitindo um maior detalhamento da ocorrência com a inclusão de arquivos multimídia e de informações adicionais como horário e local das ocorrências.

Por mais que as aplicações mencionadas permitam uma forma digitalizada de cadastrar denúncias e ocorrências, os sistemas policiais em si continuam funcionando de forma tradicional. De fato, nenhuma delas ajuda o trabalho policial na gestão de recursos pelo fato de apenas armazenarem os dados relacionados a ocorrências, sem que haja a organização desses dados para assim transmitir informações relevantes que podem ser utilizadas no dia-a-dia das forças policiais.

3. A Plataforma ROTA

Neste contexto atual, a partir de uma iniciativa da Universidade Federal do Rio Grande do Norte (UFRN) e da Secretaria de Segurança Pública e da Defesa Social do Rio Grande do Norte (SESED-RN), surge a plataforma ROTA, cujo objetivo é melhorar a segurança pública através de softwares projetados para aprimorar a infra-estrutura de segurança por trás da cidade. A plataforma ROTA fornece tecnologias para coletar, processar, compartilhar, armazenar e analisar uma vasta quantidade de dados vindos de diversas fontes, transformando dados sortidos em informações [Lopes et al. 2016].

A plataforma ROTA tem como principal objetivo auxiliar o gerenciamento das viaturas policiais nas rondas através da cidade. De forma a atender essa necessidade, a ROTA conta com uma série de aplicativos que cumprem as mais diversas necessidades, como: rastreamento das viaturas a partir de localização GPS, gerenciamento e visualização de todas viaturas ativas, análise de áreas com maior risco de ocorrências, e cadastro de novas ocorrências diretamente para a plataforma. Devido ao foco voltado às viaturas da cidade,

[2]http://www.policiapopular.com/

[3]https://goo.gl/T4cdic

a maior parte das funcionalidades da ROTA são voltadas para os membros da Polícia Civil. Porém, ainda assim, a população em geral ainda é a principal fonte de denúncia de ocorrências, sendo a principal usuária do aplicativo de cadastro de ocorrências, que é o único aberto ao público.

A ROTA é composta por três camadas de processamento de dados (Figura 1): A Camada de Integração, a Camada Analítica, e a Camada Facade. A camada de Integração é responsável por unificar os dados disponibilizados por diversas fontes de informação da cidade de Natal, como o órgão de trânsito (DETRAN-RN), o Instituto Técnico-Científico de Polícia (ITEP), e a Secretária de Estado da Segurança Pública e Defesa Social (SESED/RN), para criar informações capazes de ser utilizadas pelas outras camadas da plataforma. Notavelmente, essa camada também é a responsável por adequar os dados para serem propriamente processados pela camada analítica, que a partir disso gerará ainda mais informações a partir dos dados já coletados.

Os dados alimentados à camada analítica são processados pelo motor analítico para gerar e correlacionar informações sobre a frequência e a localização das ocorrências, e os padrões das rotas de patrulha. As novas informações adquiridas permitem um visão mais acurada dos padrões de movimentação da cidade, além de permitir um maior grau de planejamento sobre quais áreas necessitam de maior proteção em quais horários. Ainda na camada analítica, a base de dados de objetos móveis HERMES [Pelekis and Theodoridis 2014] foi utilizada para detectar padrões e dados geométricos que auxiliam na obtenção das informações já citadas.

Por fim, a camada Facade provê todas as informações necessárias para o funcionamento dos aplicativos finais da plataforma ROTA. Devido ao grande número de requisições que podem ser necessárias simultaneamente, a camada Facade foi criada seguindo o modelo chamado de Computação em Nuvem. Tal paradigma permite que os diversos recursos necessários sejam disponibilizados aos usuários sob demanda com o mínimo de esforço computacional [Mell and Grance 2011].

Nas seções a seguir, apresentamos em mais detalhes os principais elementos da arquitetura da plataforma ROTA.

3.1. ROTA Viatura

Como tradicionalmente toda a comunicação entre as viaturas policiais e a central de operação é realizada por meio da faixa de rádio policial, o repasse de ordens e informações pelo rádio é a única forma de coordenar as viaturas através da cidade. Esse processo envolve um diálogo constante entre a central e cada uma dos veículos policiais, o que o torna lento e inviável quando um grande número de viaturas estão ativas simultaneamente. A ROTA Viatura [Lopes et al. 2016] surge, então, com o objetivo principal de disponibilizar um meio de comunicação alternativo entre a central e as viaturas, permitindo uma maior flexibilidade ao trabalho da polícia militar nas rondas realizadas através da cidade.

O ROTA Viatura propõe uma solução moderna a esse problema de organização informacional. Através de um aplicativo instalado nos tablets Android presentes em cada uma das viaturas da cidade, é possível adquirir uma maior rapidez na transmissão das informações nos dois sentidos entre a viatura policial e o centro de operações. Por exemplo, a central pode enviar todas as informações sobre uma ocorrência instantaneamente

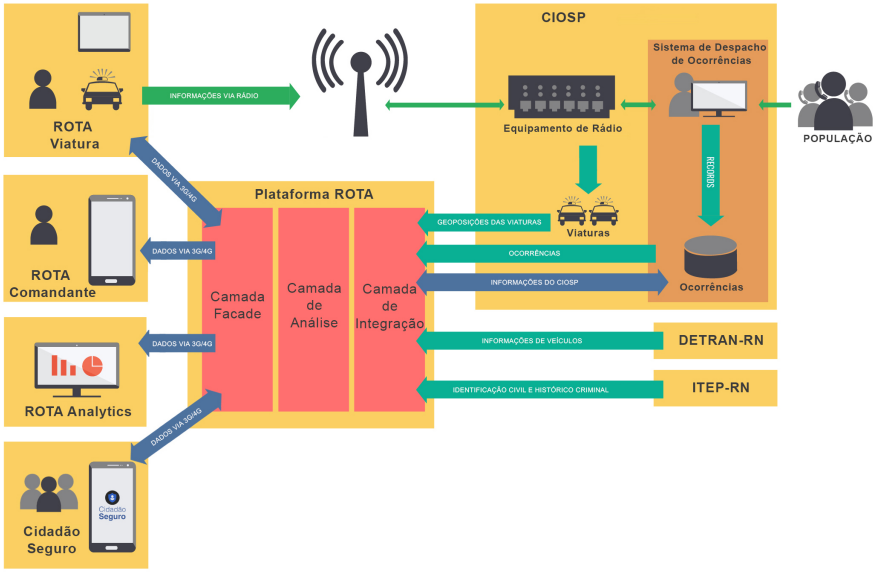


Figura 1. Arquitetura do ROTA

para a viatura policial selecionada, enquanto ao mesmo tempo outra viatura pode requisitar a permissão de reabastecimento, ambos sem a utilização do rádio policial.

Além do mais, a partir dos dados de localização GPS disponibilizados pelos tablets rodando a aplicação, é possível saber com precisão onde cada viatura se encontra em qualquer dado momento. Esses dados permitem um gerenciamento eficaz da frota policial, pois é possível saber a posição e a direção de qualquer veículo sem nem ao menos ser necessário o uso do rádio policial. Isso permite que a central de operações possa gerenciar toda a frota policial de forma ágil, enquanto, ao mesmo tempo, permitindo que o canal de rádio fique livre para transmissões importantes.

3.2. ROTA Comandante

Em contraparte ao ROTA Viatura, o ROTA Comandante [Lopes et al. 2016] é o aplicativo da plataforma ROTA que busca facilitar o trabalho da central de monitorar e comandar todas as viaturas circulando pela cidade. O trabalho de determinar a localização de cada viatura, que antes era realizado exclusivamente através da comunicação por rádio, agora passa a ser feito pelo serviço de localização do ROTA Viatura. O ROTA Comandante, que assim como o ROTA Viatura foi criado para especificamente para os tablets Android portados pela polícia, serve como um agregador para essas informações, exibindo os dados de todas as viaturas ativas diretamente sobre o mapa da cidade, tudo em tempo real (Figura 2).

Além disso, o ROTA Comandante também permite a visualização das ocorrências registradas, tanto as que ainda estão em execução quanto as já concluídas recentemente. Com isso, é possível atender as ocorrências ativas rapidamente com as viaturas mais próximas, e ao mesmo tempo também coordenar a patrulha da cidade para cobrir as áreas em que houveram mais ocorrências registradas nos últimos dias.

A partir das informações apresentadas, o comandante pode verificar quais áreas da

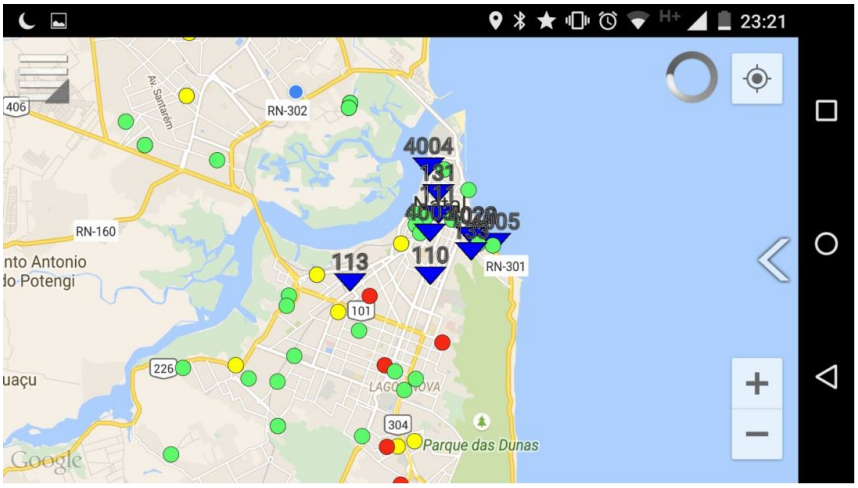


Figura 2. Tela do ROTA Comandante listando viaturas e ocorrências

cidade já estão sendo cobertas e quais ainda carecem de mais viaturas, permitindo, com isso, uma redistribuição do posicionamento das viaturas. Por fim, é possível ainda indicar a rota por qual cada uma delas deve percorrer para satisfazer a sua nova zona de cobertura.

3.3. ROTA Analytics

Em cidades inteligentes, no âmbito da segurança, o uso inteligente dos recursos disponíveis é crucial. Nesse contexto, a aplicação ROTA Analytics [Junior 2017] foi criada, e possui como objetivo principal disponibilizar uma previsão da incidência de crimes na cidade.

Essa ferramenta é integrada com o ROTA Viatura, e ajuda o supervisor de patrulha a elaborar uma lista de locais a serem visitados durante a ronda, como também o tempo que deve ser levado em cada um desses lugares. Outra ferramenta importante do ROTA Analytics é mostrar um mapa de calor que subdivide as áreas de atuação dos diferentes distritos de polícia, chamados de Áreas Integradas de Segurança Pública (AISPs), e demonstrar pela diferença de cores entre essas áreas várias métricas específicas, como predição de crimes, densidade de escolas, e outra ferramentas importantes relacionadas à criminalidade (Figura 3).

3.4. ROTA Cidadão Seguro

Desde 1995, com a criação do disque-denúncia no Rio de Janeiro, esse era o único método disponível no Brasil para o cidadão entrar em contato com a polícia a respeito de denúncias ou ocorrências. Atualmente, com o aumento vertiginoso tanto da população quanto da parcela que possui acesso a meios de comunicação [OGLOBO 2011], essa ferramenta se encontra sobrecarregada. Nesse contexto, foi criado o aplicativo ROTA Cidadão Seguro [Moreira 2017b] (ou apenas Cidadão Seguro), um projeto que possui como principal proposta o desenvolvimento de um canal de comunicação direta com o Centro Integrado de Operações de Segurança Pública (CIOSP), que ”consiste de uma sala de controle responsável por responder chamadas de um número de emergência para os serviços da polícia, bombeiros, e ambulâncias, similar ao 911 nos EUA ou 112 na Europa”[Mendonça 2016].

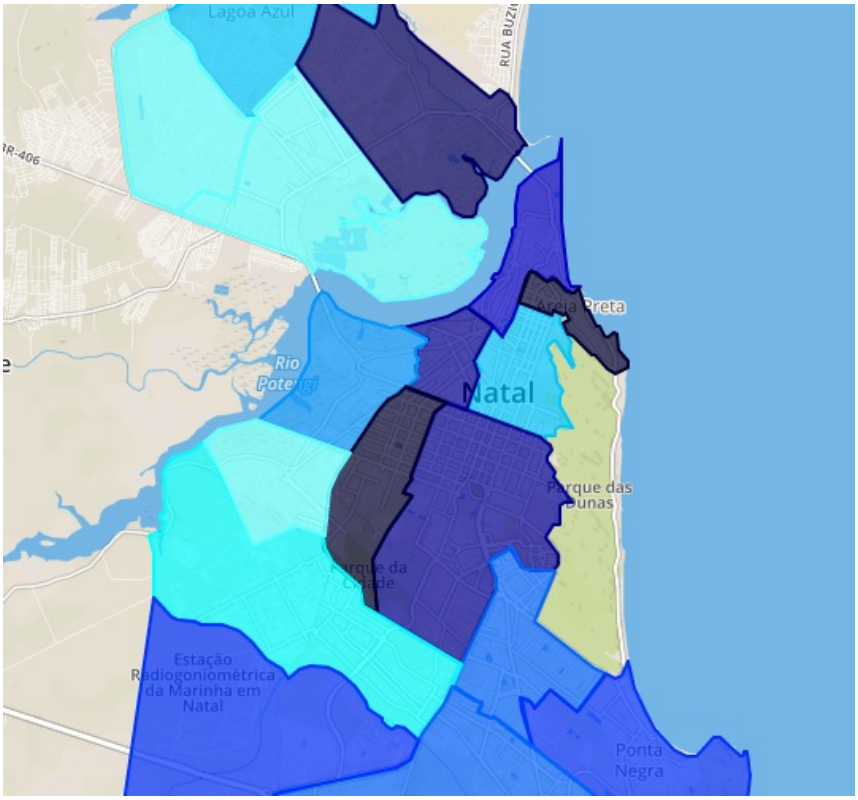


Figura 3. Mapa de Natal dividido pelos departamentos de polícia

Ao utilizar a aplicação, o usuário deverá passar por diferentes módulos para concluir o cadastro de sua ocorrência/denúncia. A partir da criação da ocorrência (Figura 4), devem ser informados dados como tipo de ocorrência, localização, e outras informações relevantes para os policiais que irão averiguá-las, como pontos de referência. Após a criação, o usuário poderá acessar a aplicação para verificar o status de suas ocorrências e assim ter uma forma de feedback, o que não era possível através do disque denúncia. A aplicação está disponível tanto para aplicativos Android quanto iOS, alcançando dessa forma o maior número de usuários possível.

3.5. Resultados

A plataforma ROTA trata diversos campos de atuação da segurança pública, além de promover soluções que beneficiam os diferentes atores participantes do processo de criação, gerenciamento, e solução de ocorrências. Esses atores vão desde o cidadão, inserido no processo de registro de denúncias e ocorrências pelo cidadão seguro, aos comandantes, que utilizam os dados analisados pelo Analytics para definir novas rotas para as viaturas utilizando o ROTA Comandante. Essas mudanças afetam diretamente a rotina dos policiais presentes nas viaturas, que podem visualizar informações importantes, como as rotas de patrulha e dados sobre as ocorrências, através do ROTA Viatura. Os benefícios e resultados trazidos por esse processo serão retratados a seguir.

Inicialmente, da aplicação Cidadão Seguro, apesar de ainda não estar disponível para a população em geral, espera-se resultados promissores. A possibilidade de adição de arquivos multimídia, que antes não estava disponível através das linhas telefônicas,

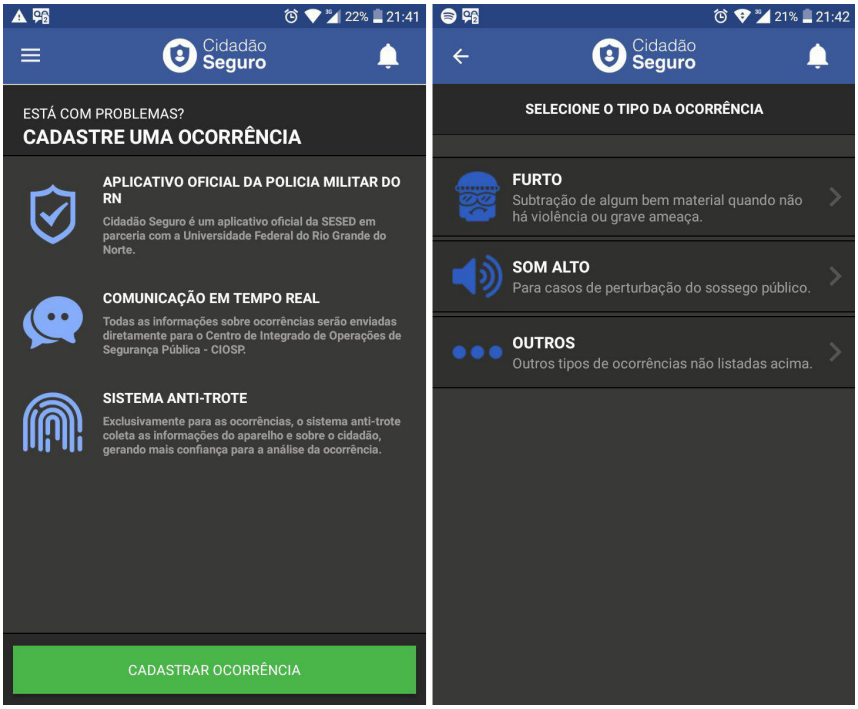


Figura 4. Tela de registro de ocorrências do aplicativo Cidadão Seguro

traz uma riqueza maior de detalhes para os policiais que estarão verificando a denúncia em questão.

Após a criação de novas ocorrências, os policiais militares encarregados de fazer a patrulha e averiguar as mesmas se beneficiam da agilidade e confiabilidade das informações passadas, disponibilizadas pela aplicação do ROTA Viatura. Isso é demonstrado através dos dados coletados desde o início da utilização da aplicação, que foi implantada inicialmente em apenas 19 viaturas, e depois foi expandida pela SESED-RN, chegando a um total de 127 viaturas após um período de 16 meses. Além disso, foram atendidas 1.103 ocorrências, 54.261 buscas de placas e 1.059 buscas de identidades civis nesse mesmo período[Moreira 2017a].

Seguindo o fluxo de resolução das ocorrências, a aplicação ROTA Analytics cumpre o seu objetivo, se tornando uma nova fonte de informações a serem consultadas e que podem ser utilizadas no planejamento de rotas nas viaturas. Assim, otimizando a gestão de recursos policiais nas áreas mais carentes de segurança da cidade.

Com os dados fornecidos pelo ROTA Analytics, em conjuntos com outras ferramentas de análise de dados, agora os comandantes podem, além de gerenciar de forma prática como será efetuada a patrulha, acompanhar a localização delas em tempo real.

De uma forma geral, os resultados obtidos desde a criação da plataforma ROTA mostram que a polícia está aberta a se adaptar para o uso de novas ferramentas que se propõem a agilizar e facilitar o trabalho em suas rotinas.

4. Novos Desafios

Por mais que a plataforma ROTA já esteja em operação e tenha diversas das suas aplicações já sendo utilizadas por parte da polícia e dos órgãos de segurança, ainda não

se pode dizer que as tecnologias estão sendo utilizadas de forma ideal. ”Tentativas de implementação de tecnologia muitas vezes falham por causa de questões humanas e organizacionais - em vez de tecnológicas - que foram negligenciadas ou subestimadas no desenvolvimento de soluções de TIC”[Griffith 1998]. Enquanto as aplicações podem ser compelidas a serem adotadas por parte da polícia, o mesmo não pode ser feito para a população, que só virá a utilizar as aplicações caso confie na sua credibilidade.

Segue como exemplo o aplicativo Cidadão Seguro. O Cidadão Seguro surgiu com o objetivo principal de facilitar as denúncias por parte da população para a polícia e, com isso, passar uma sensação de segurança para todos os usuários do aplicativo. Porém, embora a aplicação promova uma maior facilidade no uso das funcionalidades propostas, não se pode dizer que houve uma melhora significativa no sentimento de segurança do usuário, na sua interação com os órgãos de segurança, ou confiança que o mesmo tem em relação ao trabalho da polícia. ”Um dos princípios chave para o policiamento democrático é que a polícia deve operar a partir do melhor interesse da população. Assim sendo, é essencial que a polícia assegure a confiança da população faça o que fizerem”[Emsley 1983].

De forma a contornar essa situação, diversas cidades europeias passaram a utilizar as mídias sociais como forma de aproximar a população da polícia, criando um relação de confiança entre ambos. ”As mídias sociais não apenas auxiliam o policiamento comunitário, como também estimulam a introdução do policiamento comunitário. A comunicação autêntica e individual tem um impacto positivo na relação entre um policial e um membro da comunidade e, consequentemente, também no Policiamento Comunitário”[Meijer A 2013]. Ainda segundo [P. Saskia Bayerl and Markarian 2017], ”a colaboração da polícia com a comunidade pode evoluir para um policiamento bem-sucedido, mais eficaz e eficiente. Isso pode ser alcançado informando a comunidade sobre como ela pode ajudar e explicando-a os limites legais para suas contribuições”. Chegando a dizer até que ”A mídia social pode ser considerada um elemento básico no policiamento comunitário”.

As experiências adquiridas nos projetos europeus podem com isso serem trazidas e integradas também na plataforma ROTA. Um meio de interação descontraído entre a população e a policia, mesmo que realizado fora das redes sociais mais populares, seria uma ferramenta altamente efetiva em aproximar entre ambas as partes, reforçando o policiamento comunitário e facilitando ainda mais o trabalho da polícia. Essa ferramenta teria, além disso, o benefício de reestabelecer a relação de confiança entre a população e a força policial, relação essa que vem comprometida conforme os índices de violência aumentam.

5. Conclusão

Esse artigo apresentou um breve histórico sobre a plataforma ROTA, uma solução de cidades inteligentes para o problema da segurança pública na cidade de Natal, apresentando seus aplicativos e a finalidade de cada um deles. Além disso, os resultados obtidos com a implantação dos sistemas foram apresentados e discutidos, dando destaque às mudanças proporcionadas pela ROTA e como a cidade se adequou a elas. Por fim, foi apresentado à plataforma um novo desafio que, apesar de envolver em maior parte somente a população, exerce uma grande influência em como a ROTA será visto pelos habitantes de Natal e em o quão efetivas serão as aplicações que têm como foco principal a população da cidade.

Referências

Emsley, C. (1983). *Policing and its context, 1750-1870 / Clive Emsley*. Macmillan London.

Griffith, T. L. (1998). Cross-cultural and cognitive issues in the implementation of new technology: focus on group support systems and bulgaria. *Interacting with Computers*, 9(4):431 – 447. Shared Values and Shared Interfaces: The Role of Culture in the Globalisation of Human-Computer Systems.

Herald (2014). City population to reach 6.4bn by 2050 url: http://www.heraldglobe.com/news/223727231/city-population-to-reach-64bn-by-2050, acesso em: 2018-03-31.

Junior, A. A. (2017). A predictive policing application to support patrol planning in smart cities.

Leventakis, G., Papalexandratos, G., Kokkinis, G., Charalambous, E., and Koutras, N. (2016). Towards efficient law enforcement decision support systems in the area of community policing: The use of mobile applications. In *2016 European Intelligence and Security Informatics Conference (EISIC)*, pages 198–198.

Lopes, F., Coelho, J., Cacho, N., Loyola, E., Tayrony, T., Andrade, T., Medonça, M., Oliveira, M., Estaregue, D., and Moura, B. (2016). Rota: A smart city platform to improve public safety.

Meijer A, T. M. (2013). Social media strategies: understanding the differences between north american police departments.

Mell, P. M. and Grance, T. (2011). Sp 800-145. the nist definition of cloud computing. Technical report, Gaithersburg, MD, United States.

Mendonça, M. (2016). Improving public safety at fingertips: A smart city experience.

Moreira, B. C. (2017a). Plataforma e soluções para segurança pública em cidades inteligentes. TCC (Graduação) - Curso de Ciência da Computação, Departamento de Informática e Matemática Aplicada, Universidade Federal do Rio Grande do Norte, Natal.

Moreira, B. C. (2017b). Towards civic engagement in smart public security.

OGLOBO (2011). Número de celulares no brasil é maior que o de habitantes url: https://oglobo.globo.com/economia/numero-de-celulares-no-brasil-maior-que-de-habitantes-2924116, acesso em 2018-03-31.

P. Saskia Bayerl, Rusa Karlovic, B. A. and Markarian, G. (2017). *Community Policing – A European Perspective: Strategies, Best Practices and Guidelines*. Springer International Publishing.

Pelekis, N. and Theodoridis, Y. (2014). *Mobility Data Management and Exploration*.

UNDESA (2015). Department of economic and social affairs, population division. world urbanization prospects: The 2014 revision.

Uma Plataforma para Apoio à Segurança em Campus Inteligente

Silvino Medeiros, Ícaro França, Eiji Adachi, José Alex Lima, Frederico Lopes, Everton Cavalcante, Nélio Cacho

Universidade Federal do Rio Grande do Norte (UFRN)
Natal-RN, Brasil

{silvinogustavo, icazevedo10, j.alex.medeiros}@gmail.com,
{eijiadachi, fred}@imd.ufrn.br, {everton, neliocacho}@dimap.ufrn.br

Resumo. *Esse artigo versa sobre as necessidades encontradas no contexto de segurança na Universidade Federal do Rio Grande do Norte (UFRN), em Natal-RN, e propõe o SIGOc – Sistema Integrado de Gestão de Ocorrências, uma plataforma que objetiva otimizar o modo atual de operação para o gerenciamento de ocorrências nos campi da UFRN. O SIGOc consiste em dois aplicativos para dispositivos móveis e um sistema Web, permitindo que a comunidade universitária reporte ocorrências e, ao mesmo tempo, apoiando o trabalho das equipes de segurança do campus para um rápido atendimento de ocorrências. Além disso, a solução proposta visa endereçar os desafios encontrados nesse contexto em três frentes principais, a saber, gerencial, informacional e comunicacional.*

Abstract. *This paper deals with needs related to safety in the Federal University of Rio Grande do Norte (UFRN), in Natal, and proposes SIGOc – Integrated System for Occurrence Management, a platform that aims to optimize the current operation mode to manage occurrences in the UFRN campuses. SIGOc consists of two mobile applications and a Web system, enabling the university community to report occurrences while supporting the work of campus safety staff for fast occurrence handling. Moreover, the proposed solution seeks to address challenges found in this context on three main fronts, namely managerial, informational, and communicational.*

1. Introdução

Na visão de Barrionuevo *et al.* (2012), uma cidade inteligente utiliza todos os recursos tecnológicos disponíveis de maneira coordenada e inteligente para desenvolver centros urbanos que são integrados, habitáveis e sustentáveis. Nesse mesmo contexto, Lacinák *et al.* (2017) afirmam que toda cidade inteligente precisa ser uma cidade segura e que centros urbanos com essa característica devem possuir, dentre outros mecanismos, sistemas inteligentes para monitoramento, procura, detecção e identificação de crimes e eventos que ameacem a segurança pública.

Nos últimos anos, houve um aumento significativo nos índices de criminalidade nas grandes cidades brasileiras. Segundo dados do Atlas da Violência[1], a taxa de homicídios a cada 100 mil habitantes no Brasil foi de 26,1 para 28,9 no período de 2005 a

[1] http://ipea.gov.br/atlasviolencia/

2015, taxa aproximadamente quatro vezes maior do que a média do planeta. Nesse mesmo período, essa taxa passou de 13,5 para 44,9 homicídios no Estado do Rio Grande do Norte, representando uma variação de aproximadamente 231,99%, alcançando a maior variação entre todas as unidades federativas do País.

Com esse aumento da criminalidade e o crescimento da população acadêmica nos *campi* universitários brasileiros, os desafios de segurança enfrentados por uma cidade refletem-se também dentro desses espaços. Afinal, os *campi* universitários estão localizados dentro das cidades e tipicamente não há barreiras físicas que delimitem ou impeçam o livre acesso e trânsito de pessoas pelas dependências desses *campi*. A despeito das semelhanças entre cidades e *campi* universitários, a questão de segurança desses espaços guarda algumas particularidades. Enquanto nas cidades as responsabilidades da segurança pública ficam bem definidas entre Polícia Civil (policiamento investigativo) e Polícia Militar (policiamento ostensivo), tais responsabilidades não são tão claras assim nas dependências dos *campi* universitários. Por um lado, alguns defendem que, por ser papel da Polícia Militar (definido no Art. 144 da Constituição Federal do Brasil) a preservação da ordem pública, caberia também a ela atuar na segurança das universidades públicas por estas serem bens públicos. Por outro lado, há quem defenda que as universidades federais são autarquias com personalidade jurídica própria e, portanto, não caberia à Polícia Militar atuar em suas dependências. É fato que, na maioria dos *campi* de universidades federais, a atuação da Polícia Militar é assunto polêmico e que tais instituições costumam possuir órgãos próprios para a garantia da segurança em suas dependências.

No contexto da Universidade Federal do Rio Grande do Norte (UFRN), no Estado do Rio Grande do Norte, a responsabilidade da segurança universitária é da Divisão de Segurança Patrimonial (DSP), a qual agrega servidores públicos e funcionários terceirizados. É de responsabilidade dos agentes da DSP o atendimento de ocorrências diversas, desde ocorrências mais brandas como a detecção de veículos abertos ou acidentes a ocorrências emergenciais como assaltos e arrombamentos de veículos e prédios. Em um estudo realizado com a DSP, foram identificado alguns desafios que prejudicam os processos de registros de ocorrências de segurança na Universidade. Em particular, esses desafios podem ser caracterizados em três tipos: (i) desafios *comunicacionais*, que dizem respeito à comunicação tanto entre a comunidade universitária e a DSP quanto entre os supervisores da DSP e seus vigilantes; (ii) *desafios gerenciais*, que se relacionam aos processos de monitoramento e alocação de recursos, e; (iii) *desafios informacionais*, relacionados à maneira como os dados dos processos de gerenciamento são armazenados, analisados, visualizados e compartilhados.

Até o momento, tem-se conhecimento de poucas soluções endereçando do ponto de vista tecnológico a questão da segurança em *campi* universitários. Na direção de melhorar a segurança dentro de seus *campi*, a Universidade de São Paulo (USP) desenvolveu uma solução que consiste em um ponto de entrada para registrar ocorrências de segurança diretamente para o Departamento de Segurança da Universidade [Ferreira et al. 2017]. Outras propostas existentes são o *AppArmor*[2] e o *LiveSafe*[3], que contam respectivamente com a definição de perímetros virtuais com base na localização de dispositivos (*geofencing*) e chamadas de emergência através de um aplicativo móvel. Ainda assim, nenhum

[2]https://www.apparmor.com/
[3]https://www.livesafemobile.com/solutions/

dessas propostas aborda os desafios relacionados à segurança em *campi* anteriormente mencionados.

Nesse contexto, este artigo apresenta o SIGOc – Sistema Integrado de Gestão de Ocorrências, uma plataforma desenvolvida no contexto do Projeto *Smart Metropolis*[4] do Instituto Metrópole Digital (IMD) da UFRN com o intuito de apoiar as atividades de gerenciamento da segurança universitária realizadas pela DSP-UFRN, além de facilitar a comunicação entre ela e a comunidade universitária. A solução consiste no desenvolvimento e utilização dos seguintes sistemas:

(i) um aplicativo para dispositivos móveis para que as pessoas possuem algum vínculo com a universidade (discentes, docentes e servidores técnico-administrativos) possam registrar ocorrências de segurança;
(ii) um aplicativo móvel para a equipe de segurança da DSP, para que as ocorrências possam ser atribuídas de maneira mais confiável e rápida, e;
(iii) um sistema Web de gerenciamento das ocorrências, em que funcionários da DSP poderão despachar equipes de segurança para o atendimento de ocorrências, receber chamados, gerar relatórios e acompanhar o posicionamento da segurança universitária em tempo real.

O restante deste artigo está estruturado da seguinte forma. A Seção 2 apresenta uma visão geral dos desafios enfrentados pela DSP no gerenciamento da segurança pública na UFRN. O SIGOc, seus elementos e as funcionalidades de cada um são descritos na Seção 3. A Seção 4 analisa o problema comparativamente, mostrando como cada componente contribui para a otimização do modo atual de operação para o gerenciamento de ocorrências nos *campi* da UFRN. A Seção 5 trata acerca de alguns trabalhos a serem realizados futuramente com base na solução proposta. Por último, a Seção 6 traz algumas considerações finais.

2. Desafios de Segurança num Campus Universitário

No contexto da UFRN, foi possível observar que, ao longo dos anos, a obsolescência dos métodos de gerenciamento das atividades de segurança universitária acarretou na imprecisão dos dados coletados sobre as ocorrências de segurança em seus *campi*. O registro de ocorrências era feito pelos vigilantes manualmente em um caderno de ocorrências na DSP ao fim do expediente de cada vigilante. Dessa forma, não havia garantias de que todas as ocorrências atendidas pelos vigilantes ao longo do dia eram de fato registradas nem se estas eram feitas com o nível de detalhes que se esperaria. Esse fato inclusive dificultou a realização de uma avaliação situacional da segurança nos *campi* da UFRN ou da qualidade do serviço prestado pela DSP, uma vez que, com poucos registros oficiais, não era possível definir indicadores precisos sobre ocorrências registradas e atendidas ao longo dos anos.

Outra limitação observada no contexto da segurança da UFRN foi o fato de comunicação entre os supervisores da DSP e os vigilantes em ronda pelos *campi* ser toda realizada via rádio. Informações sobre a localização dos vigilantes, para saber qual deles estava mais próximo a uma determinada ocorrência, eram passadas entre todos os vigilantes em serviço e os supervisores. De forma similar, detalhes das ocorrências como

[4] http://smartmetropolis.imd.ufrn.br/

seu tipo, sua localização, características dos indivíduos e/ou ferramentas envolvidas e autor(a) do cadastro da ocorrência também eram passadas via rádio. O principal problema nesta forma de comunicação era o alcance do rádio, o qual nem sempre funcionava corretamente: o *campi* central da UFRN, localizado em Natal, é de grandes dimensões (123 hectares), enquanto nos *campi* localizados no interior do Estado do Rio Grande do Norte sequer havia a possibilidade de se usar o rádio. Havia ainda o problema da falta de registro das informações trocadas via rádio, novamente contribuindo para a perda de dados e dificultando a realização análises quanto às ocorrências registradas e atendidas e quanto à qualidade do atendimento das ocorrências. Adicionalmente, observaram-se atrasos no atendimento das ocorrências devido à natureza conversacional do processo de despacho de vigilantes e também por eventuais indisponibilidades do canal de comunicação.

Observou-se ainda que havia apenas um canal de comunicação através do qual a comunidade universitária podia reportar ocorrências de segurança observadas nos *campi* da UFRN, a saber, uma linha telefônica na central da DSP. Indivíduos da comunidade acadêmica precisavam telefonar para a DSP para reportar e descrever uma ocorrência de segurança e, por sua vez, os supervisores da DSP anotavam estas informações e as repassavam aos vigilantes para que estes fossem atender a ocorrência. O principal problema desse canal de comunicação era o seu desconhecimento por grande parte da comunidade universitária, além da falta de registro oficial das informações reportadas pela comunidade universitária aos supervisores da DSP. Ainda que existisse registro de algumas informações, a DSP não possuía ferramentas de análise de dados, ou seja, havia dificuldade em correlacionar informações sobre ocorrências anteriores e a visualização desses dados consistiam em formatos que empobrecem o entendimento deles, tais como simples planilhas e documentos de texto.

Analisando os desafios encontrados no contexto de gerenciamento de segurança na UFRN, foi feita uma caracterização dos desafios que soluções de segurança para *campus* inteligente precisam enfrentar. Tais desafios são descritos a seguir.

Desafios gerenciais. O gerenciamento eficiente de tarefas por um órgão de segurança de um *campus* inteligente é essencial para garantir velocidade no atendimento e evitar desorganização durante situações de emergência como assaltos ou arrombamentos. Na UFRN, a delegação de tarefas aos vigilantes ocorria por meio de ligação telefônica ou comunicação via rádio. A coordenação de esforços através desses métodos podia levar a atrasos devido à natureza conversacional do processo, o que podia fazer a diferença em situações de emergência. Além disso, o operador que delega as tarefas não possuía um nível profundo de informações sobre o posicionamento em tempo real da segurança universitária, ou seja, o planejamento tático também sofria com a utilização de métodos inadequados para atendimento. Portanto, faz-se necessário um método eficaz para o despacho de vigilantes às ocorrências, provendo-os com informações relevantes que os auxiliem na preparação para o atendimento e resolução das ocorrências.

Desafios comunicacionais. A deficiência comunicacional no gerenciamento de ocorrências de segurança afeta o atendimento e descrição de ocorrências a serem tratadas e isso pode comprometer a efetividade das ações de tratamento. Esse tipo de deficiência era clara no âmbito da UFRN. A comunidade universitária possuía uma linha telefônica para registro de ocorrências e, assim como nos desafios gerenciais, isso podia acarretar em atrasos de atendimento que podem ter influência significativa (e negativa) em

situações emergenciais. A inexistência de métodos de autenticação durante o registro de ocorrências deixava o sistema à mercê de usuários mal-intencionados, o que podia fazer com que recursos fossem desperdiçados em ações que eram consequência de informações não confiáveis.

Desafios informacionais. Os métodos tradicionais utilizados pelo DSP na UFRN não forneciam suporte para coleta, visualização ou análise de dados importantes sobre as ocorrências. Essa é uma abordagem que vai na direção contrária a como os dados devem ser manipulados em um contexto de uma cidade ou um *campus* inteligente. Mesmo quando coletados e armazenados, os dados ficavam contidos em planilhas ou arquivos de texto, o que impedia uma análise mais precisa das informações que eles podiam representar.

3. O Sistema Integrado de Gerenciamento de Ocorrências (SIGOc)

O Sistema Integrado de Gerenciamento de Ocorrências (SIGOc) é uma plataforma que foi desenvolvida com o objetivo de apoiar o gerenciamento de ocorrências de segurança e atividades de rondas de segurança em um *campus* universitário, além de prover à comunidade universitária um meio de comunicação direto com o órgão responsável pela segurança. No caso da UFRN, o SIGOc provê aos supervisores da DSP um sistema Web através do qual é possível ter uma visão geral das ocorrências de segurança reportada nos *campi* da universidade, bem como uma visão em tempo real da localização dos vigilantes em serviço e de quais ocorrências estão em atendimento e quais ainda não foram atendidas.

O SIGOc contempla ainda dois aplicativos para dispositivos móveis, um voltado para os vigilantes e outro para a comunidade universitária (docentes, discentes e servidores da universidade). O aplicativo *Vigilante UFRN* provê à Central de Supervisão da DSP a localização dos vigilantes em tempo real, bem como provê aos vigilantes informações sobre ocorrências a eles atribuídas através do sistema Web de gerenciamento das ocorrências. Por sua vez, o aplicativo *Campus Seguro* permite que a comunidade universitária registre ocorrências de segurança nas dependências dos *campi* da universidade, permitindo também que tais usuários acompanhem em tempo real se suas ocorrências já tiveram atendimento iniciado pela Central de Supervisão da DSP. A Figura 1 mostra como esses elementos estão interligados.

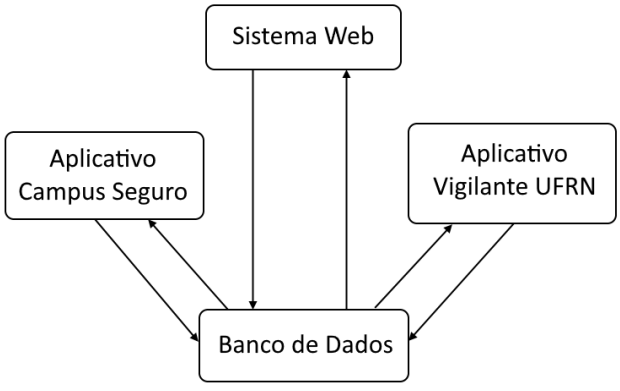


Figura 1. Arquitetura simplificada do SIGOc

Ao utilizar o SIGOc, eventuais ocorrências serão registradas através do aplicativo Campus Seguro. Feito isso, operadores do DSP, utilizando o sistema Web, alocarão um ou mais vigilantes para atendimento da ocorrência. A equipe alocada receberá então uma notificação no aplicativo Vigilante UFRN e se dirigirão até o local indicado. Ao concluir o atendimento, a equipe de vigilantes finalizará o atendimento e a ocorrência é encerrada e armazenada para possíveis consultas. Após todo esse processo, fica disponível ao usuário a possibilidade de registrar seu *feedback* referente à ocorrência em questão.

3.1. Aplicativos Campus Seguro e Vigilante UFRN

O Campus Seguro é um aplicativo para dispositivos móveis voltados para discentes e servidores da UFRN e funciona como meio de comunicação e cadastro de ocorrências de segurança. Através do aplicativo, a comunidade universitária poderá cadastrar ocorrências intuitivamente, provendo informações valiosas ao aplicativo como categoria da ocorrência (assalto, furto de veículos, consumo de drogas, etc.), localização precisa da ocorrência acompanhada de pontos de referência e número de criminosos (ver Figura 2-a/b). Mesmo que a ocorrência tenha acontecido dentro de uma estrutura física em que a localização não possa ser obtida com precisão, o usuário pode mover o marcador que indica o local do ocorrido, adicionar novos pontos de referência ou inserir informações relevantes no campo de descrição da ocorrência. O usuário acessa as funcionalidades do aplicativo apenas após se autenticar através do sistema de autenticação integrada da UFRN, isto é, somente integrantes da comunidade universitária poderão usufruir do aplicativo. O Campus Seguro também conta com acompanhamento do *status* da ocorrência em tempo real e cadastro de *feedback* após a conclusão do atendimento.

O Campus Seguro contribui para aproximar a DSP e a comunidade universitária, facilitando a interação entre os dois ao oferecer um meio direto de comunicação entre ambos sem interrupções, além de prover meios de comunicação confiáveis, diretos e sujeitos a menos interferências. Para serviços de segurança, isso se faz extremamente necessário já que as informações coletadas podem determinar o sucesso das ações empenhadas. Para o atendimento emergencial de ocorrências, a velocidade e a praticidade presentes durante o preenchimento, envio e encaminhamento dos dados é imprescindível.

O cadastro e acompanhamento de ocorrências também são funcionalidades do aplicativo Vigilante UFRN, porém outros recursos estão presentes nesse aplicativo, tais como a visualização da ocorrência através de um mapa, finalização do atendimento de uma ocorrência e notificação de chegada ao local da ocorrência (ver Figura 2-c/d). Outra funcionalidade é o *status* do vigilante, que mostra ao operador do sistema Web se o vigilante está disponível ou não para alocação a uma ocorrência. Quando o *status* do vigilante é “Disponível”, ele pode ser alocado, porém se o *status* for “Em Atendimento”, isso significa que ele já está alocado em outra tarefa. Outros *status* possíveis são “Indisponível” e “Em Pausa”, o que pode representar pausas para almoço, ida ao banheiro ou abastecimento de viatura. De maneira automática, ao se autenticar na aplicação, o vigilante passa a compartilhar sua localização com os operadores do sistema Web em intervalos de três segundos, o suficiente para limitar o uso de espaço e banda sem comprometer o acompanhamento realizado pelos supervisores. Em casos nos quais o sinal do GPS é ausente ou insuficiente, o sistema Web mantém a última posição registrada por aquele vigilante até que o aparelho do vigilante sinalize uma nova localização.

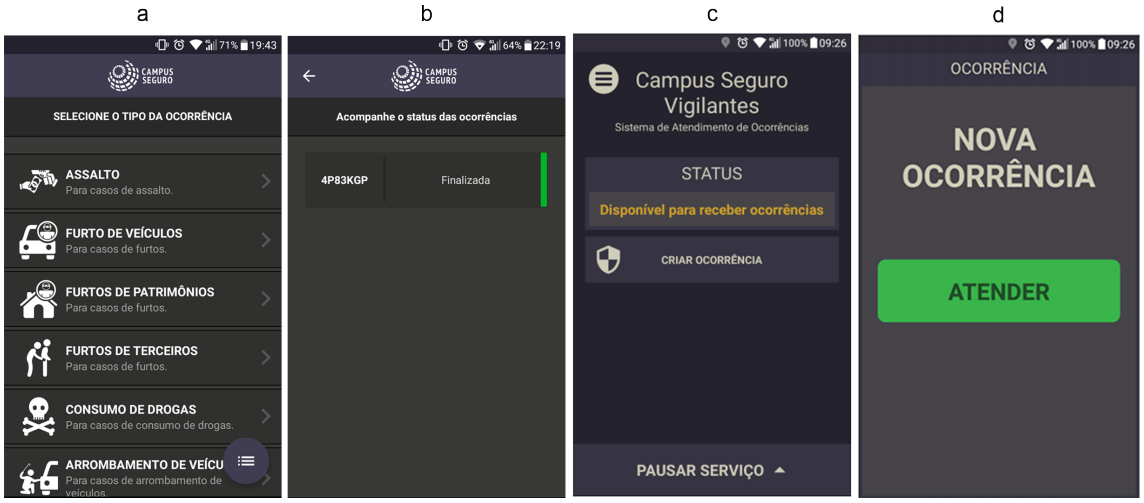


Figura 2. Aplicativos Campus Seguro (a/b) e Vigilante UFRN (c/d).

3.2. Sistema Web

O sistema Web da plataforma do Campus Seguro funciona como painel de controle para gerenciamento dos vigilantes e das ocorrências nos *campi* da UFRN (Figura 3). Esse sistema é utilizado por um operador para criar novas ocorrências e atribuir equipes de vigilantes a uma determinada ocorrência. Além disso, o operador pode adicionar, remover ou modificar vigilantes e ocorrências, bem como visualizar relatórios sobre os eventos no *campus*, como histórico de ocorrências.

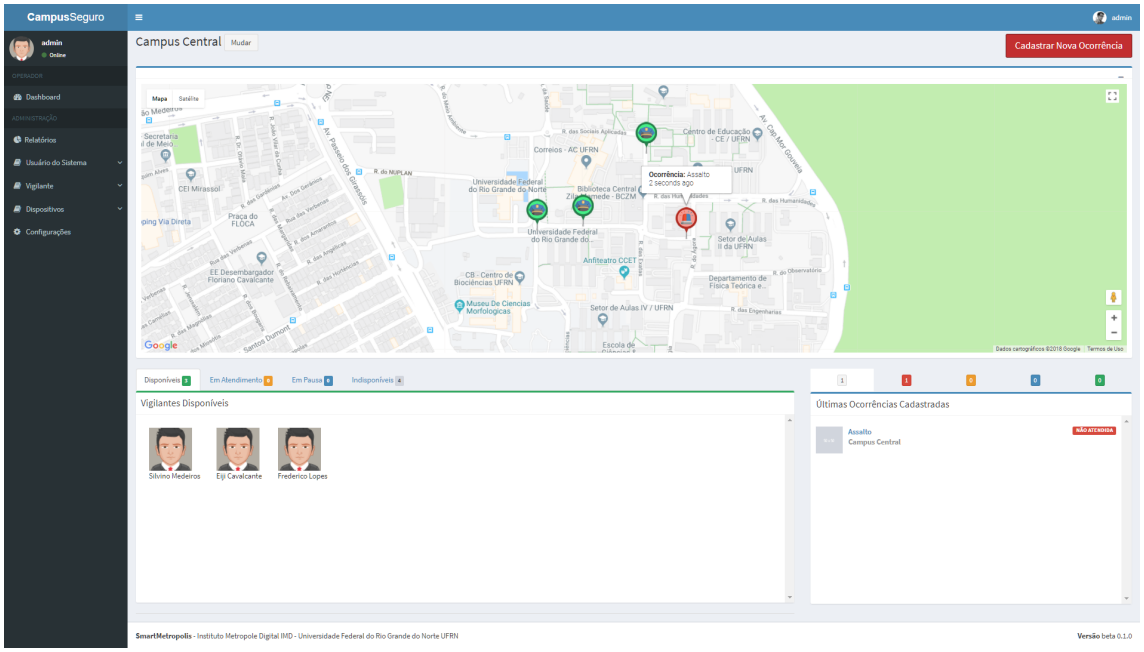


Figura 3. Tela inicial de acompanhamento do sistema Web

A adição do sistema Web à plataforma traz vários benefícios ao gerenciamento de segurança no *campus*. No momento do cadastro de uma ocorrência, o supervisor pode verificar dados importantes como usuário que a cadastrou, data e hora do cadas-

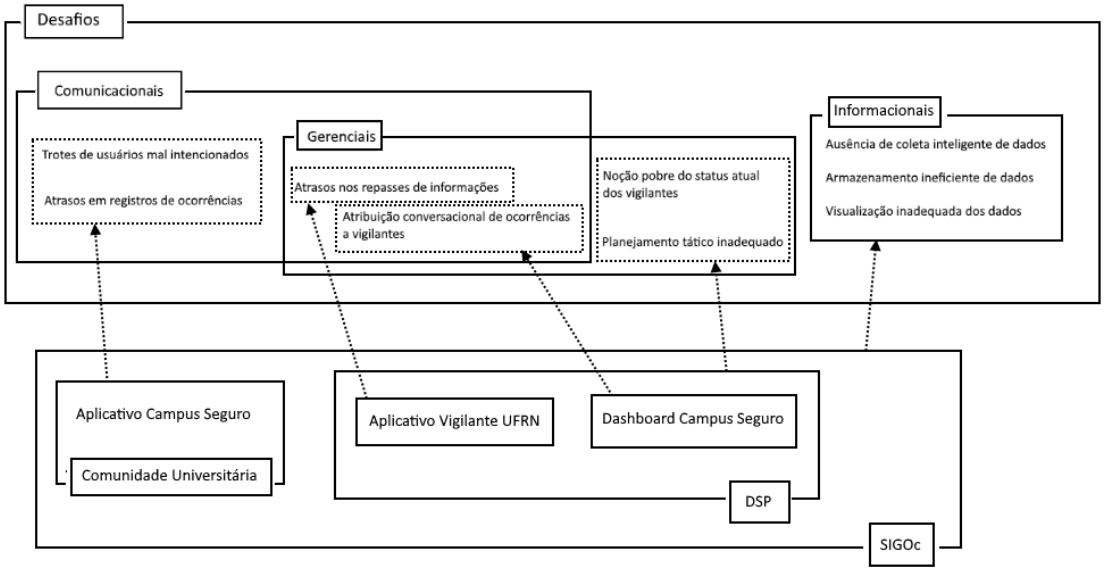


Figura 4. Relacionamento entre desafios encontrados e soluções propostas

tro, localização e tipo da ocorrência, e é o supervisor que fica responsável por atribuir qual vigilante deve atendê-la. Esses dados atualmente são repassados por telefone e o sistema Web padroniza essas informações, deixando clara ao operador a situação atual da ocorrência, reduzindo os riscos de falhas comunicacionais entre a comunidade universitária e a DSP.

Outro recurso importante provido pelo sistema Web é a visualização e gerenciamento de vigilantes em tempo real. Apesar de os vigilantes serem ainda despachados manualmente para o atendimento de ocorrências, essa atividade é agora feita com mais eficácia, já que o operador sabe o local exato da ocorrência e os vigilantes próximos a ela. De posse dessas informações, o operador seleciona a equipe de um ou mais vigilantes e os despacha para atender essa ocorrência. Esse alto nível de granularidade no processo tanto torna o gerenciamento de vigilantes e ocorrências mais robusto, com maior confiabilidade e precisão, quanto aumenta a qualidade das informações coletadas, agora mais precisas e confiáveis. As informações coletadas e armazenadas pelo sistema Web são posteriormente utilizadas para processamento e geração de relatórios.

4. Análise da Adoção do SIGOc no âmbito da UFRN

Na análise de como o SIGOc pode auxiliar a comunidade universitária e o DSP da UFRN nas questões relacionadas a ocorrências de segurança, foi possível observar como cada elemento da plataforma aborda um ou mais dos desafios identificados (ver Seção 2) e como a união dessas partes também colabora de um modo diferente para endereçá-los. O diagrama ilustrado na Figura 4 sintetiza essas relações entre os desafios e as soluções propostas.

O aplicativo Campus Seguro é direcionado à comunidade universitária e aborda principalmente o problema da comunicação desta com a DSP. O contexto comunicacional é melhorado substancialmente, visto que discentes e servidores agora possuem uma maneira rápida de reportar ocorrências e os operadores do sistema Web na DSP visualizam

essas ocorrências com informações pertinentes, como uma maior precisão da localização, que muito provavelmente seria perdida ou de menor qualidade na comunicação oral. Os operadores do sistema Web, com base nessas informações, podem gerenciar os vigilantes com mais precisão, ou seja, o contexto gerencial também é melhorado.

Ainda no contexto gerencial, é possível perceber que o aplicativo Vigilante UFRN, junto do sistema Web, traz recursos que aprimoram os processos de gerenciamento da DSP. O aplicativo repassa informações primordiais ao gerenciamento de vigilantes, como a localização atual do vigilante, *status* de atendimento de uma ocorrência e *status* atual do vigilante, informações essas que são mostradas no sistema Web e são essenciais para alocação dos vigilantes no atendimento de uma eventual ocorrência. Além disso, os recursos do sistema Web fornecem um controle muito mais refinado sobre o gerenciamento de vigilantes e ocorrências, o que seria mais difícil de ser alcançado apenas através de rádio e ligações telefônicas.

No contexto informacional, o SIGOc como um todo propicia maneiras sofisticadas de armazenamento e processamento utilizando os dados obtidos a partir dos elementos que fazem parte da plataforma. Por exemplo, as localizações dos vigilantes são armazenadas no banco de dados da plataforma e diversas análises podem ser realizadas sobre desses dados, revelando padrões de deslocamento e distribuição espacial dos vigilantes. O sistema Web também possui recursos de gerenciamento, permitindo cadastro, atualização e remoção de usuários e vigilantes.

5. Trabalhos em Andamento e Futuros

As deficiências comunicacionais, informacionais e gerenciais que se fazem presentes no contexto atual de gerenciamento de ocorrências nos *campi* da UFRN foram as principais motivações para o desenvolvimento do SIGOc e as soluções associadas. Apesar de já contar com recursos que otimizam o atendimento de ocorrências e o gerenciamento de vigilantes, o SIGOc ainda contém recursos importantes a serem implementados. Por exemplo, ocorrências podem ser atribuídas apenas a vigilantes com *status* “Disponível”, ou seja, vigilantes com *status* “Em Atendimento” não serão considerados para atribuição, impedindo realocação de pessoal em atendimento mediante à ocorrências concorrentes.

Atualmente, as aplicações que constituem o SIGOc estão em operação no *campus* central da UFRN, na cidade do Natal, e já são capazes de abordar os desafios anteriormente mencionados para contribuir com a melhoria da segurança por meio da gestão eficiente de ocorrências. Todavia, alguns recursos passaram a se fazer necessários para melhorar ainda mais esses fatores, os quais são discutidos a seguir.

Implementação das Vias Azuis. A dimensão geográfica da UFRN traz dificuldades com relação à cobertura da segurança. A vigilância não consegue monitorar toda a região e, por isso, alguns locais estão mais propensos a ocorrências, locais estes muitas vezes percorridos pela comunidade universitária. As vias azuis serão trajetórias distribuídas pela Universidade onde a vigilância é fortalecida através de câmeras e vigilantes. O aplicativo Campus Seguro proverá funcionalidade para calcular as Vias Azuis, ou seja, trajetórias dentro dos campi da UFRN com maior cobertura de câmeras de segurança e pontos de ronda dos vigilantes para que alunos e servidores se desloquem com maior sensação de segurança dentro da UFRN;

Botão de Pânico. O Botão de Pânico funcionará como ferramenta para reportar

rapidamente alguma ocorrência urgente dentro do *campus*, o que pode ser uma ferramenta útil principalmente em uma situação urgente ou de tensão do usuário, como no caso de um assalto. Atualmente, o aplicativo Campus Seguro requer vários passos a serem seguidos antes da ocorrência ser registrada, o que pode dificultar a sua utilização em casos emergenciais, nas quais o usuário precisa cadastrar a ocorrência rapidamente.

Relatórios. O sistema Web, em sua versão atual, é capaz de gerar relatórios para possibilitar consultas de dados sobre ocorrências e vigilantes. Contudo, o formato de apresentação desses dados é muito pobre e mostra apenas informações básicas sobre o tópico do relatório. No momento, não é possível correlacionar informações e obter um relatório de uma lista das ocorrências cadastradas em um intervalo de tempo. Portanto, esse refinamento é necessário para que o processamento dos dados armazenados pelo sistema resulte em informações de maior valor agregado que permitam análises mais significativas através do sistema.

6. Conclusão

Cidades inteligentes possuem necessidades variadas que vão desde o engajamento cívico até processamento e armazenamento de dados de forma adequada e que gere informações de valor agregado para apoiar processos de tomada de decisão por parte dos usuários. Alguns desafios decorrentes da deficiência em atender essas necessidades podem se manifestar tanto na própria cidade inteligente quanto em um *campus* universitário. Através da observação desses desafios no âmbito do gerenciamento de segurança na UFRN, este trabalho propôs a plataforma SIGOc como uma solução composta de dois aplicativos para dispositivos móveis e um sistema Web como um painel de controle no intuito de contribuir para melhoria da qualidade dos processos de gerenciamento de ocorrências pela DSP. Através de uma maior compreensão da problemática e o uso já em andamento da solução, trabalhos futuros podem ser realizados visando um aumento da segurança da comunidade universitária e melhora ainda mais sensível dos processos gerenciais da DSP.

Referências

Barrionuevo, J. M., Berrone, P., Ricart, J. E. (2012) “Smart cities, sustainable progress”, IESE Insight 14, pp. 50-57.

Ferreira, J. E., Visintin, J. A., Okamoto Jr., J., Pu, C. (2017) “Smart services: A case study on smarter public safety by a mobile app for University of São Paulo”, Proceedings of the 2017 IEEE Conference on Smart City Innovations. USA: IEEE.

Lacinák, M., Ristvej, J. (2017) “Smart city, safety and security”, Procedia Engineering 192, pp. 522-527.

Uma metodologia de localização Indoor para smartphones em ambientes de Cidades Inteligentes.

Hilário José Silveira Castro[1], **Ivanovitch Medeiros Dantas da Silva**[2], **Silvio Costa Sampaio** [2]

[1]Universidade Federal do Rio Grande do Norte (UFRN)
Programa de Pós-Graduação em Engenharia Elétrica & Computação
Caixa Postal 1524 – 59.078 − 970 – Natal – RN – Brasil

[2]Universidade Federal do Rio Grande do Norte (UFRN)
Instituto Metrópole Digital

hilariojscastro@gmail.com, ivan@imd.ufrn.br, silviocs@imd.ufrn.br

Abstract. *Indoor localization systems have attracted interests from a wide range of areas, allowing for the optimization and emergence of new services. With the expansion of instrumented scenarios with IoT devices, location systems gain even greater importance, providing references to a location based on the transmitters of the environment. In this work, we present a methodology for an indoor location system for smartphones, capable of tracking the movement of a user and defining their location more precisely, by merging data from different sources and Kalman filters, such as inertial sensors, Wi-Fi and BLE wireless networks installed in the environment.*

Resumo. *Sistemas de localização em ambientes internos (indoor) têm atraído interesses das mais variadas áreas, permitindo a otimização e o surgimento de novos serviços. Com a expansão de cenários instrumentados com dispositivos IoT (do inglês Internet of Things), os sistemas de localização ganham uma importância ainda maior, proporcionando referências para uma localização baseada nos transmissores do ambiente. Neste trabalho, apresentamos uma metodologia para um sistema de localização indoor para smartphones, capaz de rastrear a movimentação de um usuário e definir a sua localização de forma mais precisa, através da fusão de dados de diferentes fontes e Filtro de Kalman, tais como sensores inerciais, redes sem fio Wi-Fi e BLE instalados no ambiente.*

1. Introdução

A evolução tecnológica dos últimos anos permitiu o surgimento de diversos novos conceitos, entre os quais merecem destaque a Internet das Coisas (do inglês *Internet of Things*) e as Cidades Inteligentes que, juntos, prometem revolucionar a utilização da tecnologia para o benefício do cidadão. De fato, uma cidade inteligente pressupõe a utilização de uma grande quantidade de dispositivos de IoT, proporcionando a comunicação e interação dos diferentes objetos e atores da cidade, através da Internet [Al-Fuqaha et al. 2015]. Neste cenário, os sistemas de localização *indoor* (*Indoor Location System* - ILS) desempenham um importante papel, ao permitir que as informações dos diferentes elementos monitorados possa ser espacialmente referenciada. Por exemplo, ao ativar um serviço inteligente, como algum serviço de emergência, através da aplicação

de métodos de ILS, é possível determinar a posição de um usuário em uma planta por meio do seu smartphone com uma certa precisão. Este trabalho foi desenvolvido considerando esta situação e objetivando aumentar a precisão das coordenadas geradas. Soluções para este problema são essenciais às Cidades Inteligentes, visto que grande parte dos elementos a serem monitorados certamente estarão em ambientes internos e não em áreas abertas, onde o uso do GPS (do inglês *Global Position System*) seria uma solução óbvia.

Com uso de sistemas microeletromecânicos nativos de um smartphone, uma ILS pode rastrear a movimentação do usuário e gerar sua posição e coordenadas na planta por meio de métodos como o *Pedestrian Dead Reckoning* (PDR). Entretanto, imprecisões dos sensores inerciais deturpam as estimações do sistema, acarretando em erros cumulativos no posicionamento do usuário. Uma forma de mitigar deturpações é a hibridização do sistema, fornecendo novas referências para o rastreio [Correa et al. 2017].

A hibridização de uma ILS ocorre com a mescla de sistemas ou técnicas, como adição de redes de transmissores sem fio (*wireless*) para fornecer novos referenciais ao sistema inercial. Com uso das indicações da potência dos sinais recebidos (*Received Signal Strength Indicator* - RSSI) de estações sinalizadoras, uma nova coordenada pode ser estimada, servindo de referência extra para o sistema.

O uso de parâmetros de transmissores sem fios, relacionado ao espalhamento de dispositivos sem fios, como referências (âncoras) devem ser tidos em conta já na instalação e disposição destes dispositivos na planta. Dois padrões são popularmente empregados: a família de padrões IEE802.11 (mais conhecida como *Wi-Fi*) e *Bluetooth Low Energy (BLE)*. Este dois padrões de comunicação são normalmente encontrados em smartphones [Correa et al. 2017].

Neste contexto, este artigo apresenta uma proposta de metodologia de ISL para smartphones. Brevemente, é proposto uma combinação de dados gerados por sistemas inerciais na metodologia PDR com dados derivados da RSSI no sistema de trilateração, formado com base em redes *Wi-Fi* e *BLE*, corrigidas por um método de correção de parâmetro. Ao final, os dados serão interpolados por meio de um Filtro de Kalman.

O restante do artigo é dividido em: Seção 2 com trabalhos relacionados; Seção 3 com a descrição da metodologia proposta e detalhes quanto: métodos de rastreio do alvo com uso dos sensores inerciais; propagação de sinais e a distância entre equipamentos; Correção entre distância entre equipamentos e localização por trilateração; e fusão de dados por Filtro de Kalman; Na seção 4 são apresentados alguns testes e análise de resultados; Por fim, a Seção 5 Com a conclusão do artigo e trabalhos futuros.

2. Trabalhos Relacionados

Nesta seção, são abordados alguns trabalhos encontrados na literatura relacionados com a construção de sistemas híbridos de localização para smartphones. Em geral, a propostas apresentadas utilizam do multi-sensoriamento do dispositivo, por meio do uso dos sensores para medição inercial, associando a capacidade de realizar comunicação com redes *Wi-Fi* ou *BLE*, entre outros [Correa et al. 2017].

Quanto ao uso de parâmetros de transmissão, uma metodologia comumente utilizada é a *fingerprint*. Neste método, são registradas as RSSI de todas as estações em cada posição da planta, criando um banco de dados de RSSI. Na resolução, um método de

associação do banco de dados utiliza as RSSI recebidas posteriormente para determinar a posição. Outras propostas utilizam práticas para aprimorar ou acelerar os resultados, como em [Liu et al. 2012] que utilizou *Wi-Fi fingerprints* fusionados com dados inerciais por meio de Modelo Oculto de Markov para gerar coordenadas. De forma similar, os autores [Radu and Marina 2013] combinaram dados utilizando de Filtro de Partículas e [Chen et al. 2015] por filtro de Kalman. Outra abordagem está quanto a aproximação entre dispositivos, como nos autores [Chen et al. 2016] que utilizaram de informações de redes *Wi-Fi* e *BLE* para corrigir sistemas inerciais.

Já o uso de *fingerprint* está associado a precisão em relação a outros métodos baseados em parâmetros de sinais. Entretanto, o método possui como contrapartida um grande esforço na geração e recalibração do mapa de RSSI [Khalajmehrabadi et al. 2017]. Uma formulação com menores necessidades computacionais, menores esforços prévios e sem necessidade de recalibração de mapa é a trilateração. Mesmo com nível de precisão inferior ao *fingerprint*, a trilateração pode ser utilizada de forma semelhante ao outro método para fusão de dados de redes com sistemas inerciais.

Ao tratar de cenários *IoT*, um smartphone pode utilizar as estações *wireless* do ambiente como âncoras ou referências em seus métodos. Duas tecnologias de comunicação nativas no dispositivo e empregadas em dispositivos *IoT* são o *Wi-Fi* e *BLE* [Al-Fuqaha et al. 2015]. Como pode ser observado nas citações, sistemas híbridos trazem uma gama de possibilidades e uma abordagem pouco trabalhada está quanto ao o uso de padrões mistos de transmissão.

Finalmente, neste trabalho é apresentada uma proposta de metodologia para aplicação de uma ILS para smartphones que utiliza uma maior quantidade de dados de diferentes fontes a fim melhorar a precisão das coordenadas geradas, em relação aos métodos descritos listados acima. Na abordagem aqui proposta, as coordenadas serão inicialmente determinadas com base da combinação do método PDR e dados de trilateração. Em adição, será utilizado um sistema misto de âncoras *Wi-Fi* e *BLE*, juntamente com um método correção de parâmetro de distância.

3. Metodologia

Com base em um cenário *IoT* instrumentado com transmissores *wireless*, um usuário, percebido e representado por seu smartphone partirá de um ponto de localização conhecida e terá seu trajeto na planta rastreado pelo sistema. O posicionamento do alvo é definido como uma coordenada em um eixo cartesiano, onde as âncoras *Wi-Fi* e *BLE* possuem suas coordenadas conhecidas.

No método, sensores inerciais identificam a movimentação do usuário com uso do sensor acelerômetro, registrando a ocorrência do deslocamento do portador do smartphone. O direcionamento do usuário é rastreado por meio do sensor orientação, que atua como um compasso digital. Ao final, os dados são aplicados ao método PDR, gerando as primeiras coordenadas.

Em paralelo, o sistema coletará as RSSI das estações por 5 segundos, utilizando a média das amostras para cada estação detectada. Com uso de parâmetros associados a cada estação, é estimada a distância entre estações, concluindo as novas coordenadas com uso da Trilateração. Com os dados formulados, o método finaliza por interpolar

os resultados através do Filtro de Kalman, fornecendo um novo dado relativo entre as formulações.

O fluxograma do método segue como na orientação da Figura 1.

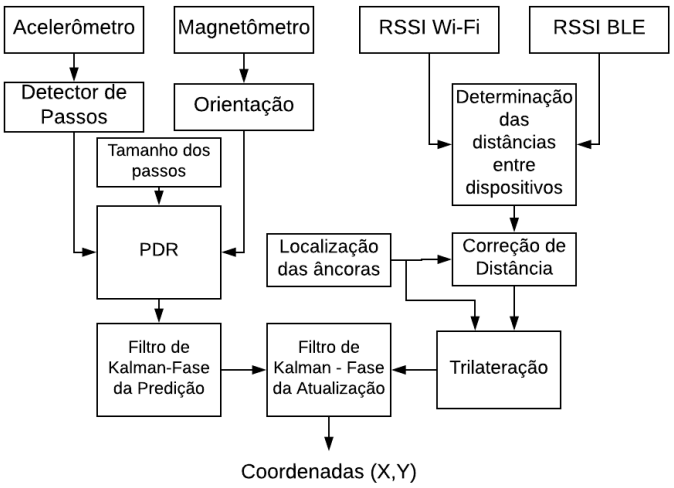


Figure 1. Diagrama de fluxo da metodologia.

3.1. Sensores inerciais e PDR

O rastreio da movimentação do usuário é realizado pelo método *Pedestrian Dead Reckoning* (PDR), descrevendo a posição atual utilizando como base uma posição anterior e o deslocamento realizado [Ojeda and Borenstein 2007]. O método estima as coordenadas atuais (K) de um usuário com base no conhecimento das coordenadas (X, Y) anteriores ($K-1$), acrescidas do produto da direção tomada em cada eixo (θ_K) e na quantidade de movimento (d_K) realizado pelo usuário. Como demonstradas pela Equação (1):

$$\begin{bmatrix} X_K \\ Y_K \end{bmatrix} = \begin{bmatrix} X_{K-1} \\ Y_{K-1} \end{bmatrix} + \begin{bmatrix} cos(\theta_K) \\ sen(\theta_K) \end{bmatrix} d_K \tag{1}$$

A quantidade de movimento e orientação angular são rastreadas por unidades de medição inerciais, em três eixos, contidas no smartphone. Os sensores da unidade requisitados são: Acelerômetro e magnetômetro. Magnetômetro é um sensor utilizado para medição de densidade unidade, indicando a direção e sentido de campos magnéticos em sua proximidade, semelhante a uma bussola, rastreando nos eixos a densidade do fluxo magnético (medido em Tesla ou As/m^2) [Milette and Stroud 2012]. Enquanto o acelerômetro atua na para medição da aceleração do dispositivo em m/s^2.

Com o porte do dispositivo na mão do usuário, a detecção de um passo ocorre quando movimentação do usuário implicar na oscilação da aceleração linear em sua mão, durante ato de erguer e repousar o corpo ao dar um passo.

O ato de erguer o corpo indicará um aumento na aceleração vertical, e o movimento sequencial de repousar indicará uma aceleração em sentido contrário, com os atos rastreados com base nos valores de referências positivos e negativos. Em adição, a identificação de ultrapassagem das referências deve ocorrer em um intervalo de

tempo semelhante ao tempo de um passo humano, indicado entre $150\ ms$ e $400\ ms$ [Jin et al. 2011], a exemplo da contagem de passos da Figura 2. Com a detecção de um passo, a quantidade de deslocamento (d_k) será de $70\ cm$, que está relativo a um valor médio de um passo humano [Kent 2017].

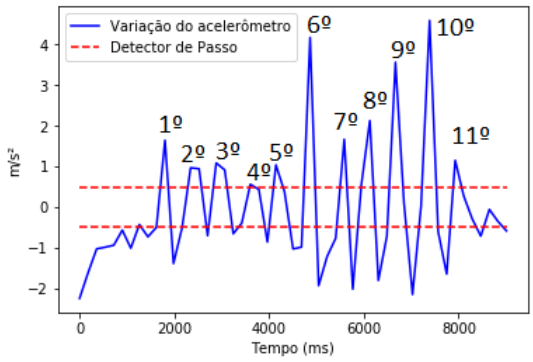


Figure 2. Em uma caminhada de 10 passos o sistema identificou 11 passos.

A mudança da orientação (θ_k) do usuário será realizada pelo sensor orientação. Este sensor pode monitorar a direção do dispositivo em relação ao norte magnético da terra com uso do acelerômetro e magnetômetro. Entre a formação de 3 eixos no sensor, será utilizado o eixo azimute [Chen et al. 2015]. A leitura do sensor pode ser efetivada por meio de atividades dos eventos dos sensores do dispositivo, como no sistema operacional Android [Milette and Stroud 2012]. Entretanto, os ruídos atrelados aos sensores ocasionam em erros acumulativos para estimação das coordenadas do método PDR, resultando na deturpação da orientação e na quantidade de movimento do usuário, como no exemplo da Figura 2.

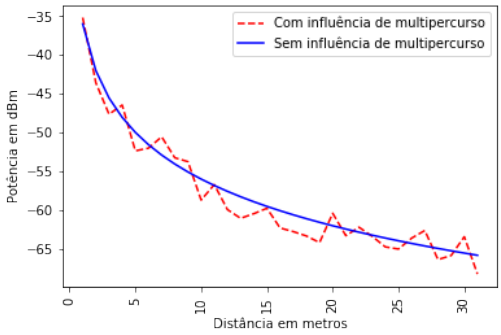
3.2. Estimação da distância entre estações e a Trilateração.

No dado ambiente, as estações com transmissões *wireless* enviam sinais a serem captados pelo smartphone. Com uso das RSSI das estações, o dispositivo estimará a distância até o transmissor, por meio da equação (2), baseada no modelo de propagação de sinais Log-distância [Gomes et al. 2013].

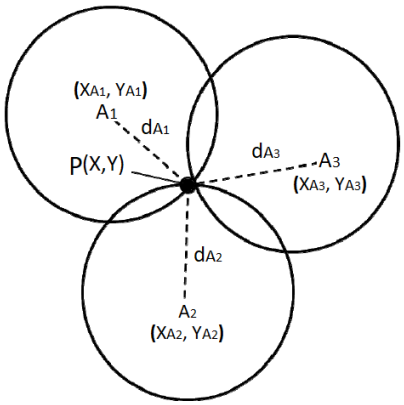
$$d = d_0.10^{(Pr(d_0)-Pr(d))/10.\eta} \tag{2}$$

Na Equação do modelo tem-se: $Pr(d)$ como a intensidade de potência do sinal recebido (em dBm) a uma dada distância; $P(d_0)$ demonstra a potência do sinal em uma distância de referência (d_0), normalmente a um metro do transmissor; η representa a intensidade de desvanecimento do sinal no ambiente, utilizado com intensidade 2 [Rappaport 2001].

Entretanto, os sinais ao propagarem pelo ambiente poderão interagir com barreiras, como estruturas da instalação e o corpo do usuário [Junsheng 2017]. A interação introduz mudanças na intensidade do sinal, seguindo o comportamento de uma variável aleatória gaussiana de média zero com desvio padrão σ [Gomes et al. 2013], como exemplificado na Figura 3(a). O efeito descrito, chamado de múltiplos percursos, junto a variação do η introduzem os erro na estimação da distância entre estações.



(a) Comparativo entre desvanecimento esperado do sinal e suas mudanças da intensidade da RSSI ocasionadas pelos múltiplos percursos.



(b) Ponto desconhecido P é o encontro entre três raios formados ao redor de 3 coordenadas conhecidas.

Figure 3.

Utilizando a distância estimada entre dispositivos, aplicasse a Trilateração para determinar as coordenadas do alvo. No principio, é computada a coordenada desconhecida como um ponto de distância comum entre pelo menos três outros pontos (âncoras) de coordenadas conhecidas [Makki et al. 2015], a exemplo da Figura 3(b).

A separação física entre estação receptora e transmissora está relacionada com a distância euclidiana entre os equipamentos. De base no uso necessário de pelo menos 3 pontos, será formado um sistema de Equações (3):

$$\begin{aligned} d_{A_1}^2 &= (X - X_{A_1})^2 + (Y - Y_{A_1})^2 \\ d_{A_2}^2 &= (X - X_{A_2})^2 + (Y - Y_{A_2})^2 \\ d_{A_3}^2 &= (X - X_{A_3})^2 + (Y - Y_{A_3})^2 \end{aligned} \tag{3}$$

Na Equação (3): X e Y correspondem as coordenadas desconhecidas do ponto móvel; d_{A_n} (com n correspondendo aos pontos 1, 2 e 3) como a distância entre dispositivo móvel e ponto fixo n; X_{A_n} e Y_{A_n} são as coordenadas das estações fixas.

Não obstante, em consequência aos erros na estimação da distância entre estações, as coordenas (X,Y) do alvo terão suas resoluções deturpadas. Para mitigação dos ruídos, será atribuído uma metodologia de correção do parâmetro de distância entre estações, utilizando um sistema de votos. Uma âncora e sua distância para o dispositivo móvel receberá um voto de inadequado seguindo o sistema de Equação (4):

$$V_n = \begin{cases} V_n, & \text{se} \quad d_{A_n A_I} - d_{A_n} \leq 1,25 d_{A_I} \\ V_n + 1, & \text{se} \quad d_{A_n A_I} - d_{A_n} > 1,25 d_{A_I} \end{cases} \tag{4}$$

No sistema: V_n corresponde a variável de votos de deturpação para uma âncora n, com contagem inicialmente nula; $d_{A_n A_I}$ é a distância euclidiana entre uma âncora avaliada (n) para uma âncora avaliadora (I); d_{A_n} é a distância entre âncora avaliada e smartphone; d_{A_I} distância entre âncora avaliadora e smartphone.

A âncora com menor número de votos de deturpação (V_n) será considerada a referência do grupo. Com base na âncora de referência, a distância das demais âncoras para o smartphone serão corrigidas seguindo a equação (5), assumindo : $d_{A_{V_n}}$ como a distância da âncora de referência para o smartphone e $d_{A_{V_n}A_I}$ a distância euclidiana de uma âncora (I) para a âncora de referência. Em adição, devem ser eliminadas âncoras com distâncias para o dispositivo móvel maiores que o alcance de suas transmissões no padrão, considerado 10 metros no projeto.

$$d_{A_I} = d_{A_{V_N}A_I} - d_{A_{V_N}} \tag{5}$$

Com o ajuste nas distâncias finalizado, a resolução do sistema (3) segue utilizando o Método dos Mínimos Quadrados (MMQ) [Anton and Busby 2011].

Com a possibilidade de resultar em diversas âncoras, será aplicada a trilateração em grupos formado por todas as estações fixas, formando M grupos de 3 âncoras, resultando em várias coordenadas (X, Y) diferentes. A coordenada desconhecida será então atrelada a média da soma das coordenadas dos grupos de 3 estações.

3.3. Filtro de Kalman e a Fusão de técnicas

Para a interpolação dos dados das metodologias será utilizado o Filtro de Kalman (FK). O FK é um algoritmo de filtro Bayesiano recursivo, capaz de utilizar medições ruidosas ao longo do tempo para gerar resultados que tendem a se aproximar dos valores reais das grandezas medidas [Koo et al. 2014]. O FK é regido por dois grupos de equações: Grupo de Predição e Grupo de Atualização.

Na fase de predição, as Equações utilizam do modelo de dinamismo do sistema associado ao método PDR, como na Equação (6):

$$X_K = X_{K-1} + u_k \tag{6}$$

Na Equação, tem-se: X_k como as coordenadas (X_k, Y_k) atuais do alvo e X_{k-1} corresponde as coordenadas anteriores; u_k descreve a variação das coordenadas (X, Y) entre os estados anterior e atual, fornecida pela transição de coordenadas do PDR. Em sequência, é associado a equação de propagação da covariância do estado ao ruído do modelo, finalizando a etapa de equações de predição, como na Equação (7):

$$P_K^- = P_{K-1} + Q \tag{7}$$

Compõe a Equação (7): P_k^- é Propagação da covariância do estado a priori; Q representa uma variável aleatória com comportamento de ruído gaussiano de média zero no PDR. Seguindo o fluxo das equações, com adição das predições ao o grupo de Atualização. Neste, ocorre a fusão das coordenadas do PDR e da trilateração, expressa em P_k, como mostram as Equações (8), (9) e (10):

$$K_k = P_k^-(P_K^- + R)^{-1} \tag{8}$$

$$X_k = X_k^- + K_k(Z_k - X_k^-) \tag{9}$$

$$P_k = P_K^- - K_k P_k^- \tag{10}$$

São indicados na Equação: K_k como o ganho do filtro de Kalman; Z_k indica as coordenadas da trilateração (tr) (X_{tr},Y_{tr}); R compõe uma variável aleatória com comportamento de o ruído gaussiano de média zero na trilateração; P_k fornece as coordenadas fusionadas (X, Y) computadas pela conclusão do filtro. Por meio de K_k, o FK determinará a contribuição de cada método, adicionando um fator de qualidade as coordenadas de cada método, proporcionados por meio dos ruídos R e Q.

Quanto maior a intensidade de R em relação a Q, indicará uma confiança maior do método da trilateração em relação ao PDR, o mesmo vale para uma análise inversa, favorecendo o PDR. Outro ponto está relativo ao uso da intensidade dos ruídos diferentes para atrelar resultados melhores em situações diversas. Um exemplo de modificação das intensidades para favorecer os resultados está quanto ao favorecimento para o PDR para casos que existam menos que 3 âncoras para aplicar a trilateração, permitindo o resultado refinado para casos específicos.

4. Testes e análise de resultados.

Nesta seção, serão demonstrados alguns testes da proposta. Os dados dos sensores inerciais e das redes foram coletados e armazenados por meio de um aplicativo desenvolvido pelo grupo de pesquisa e aplicada na plataforma Smartphone Samsung Galaxy S4. Os dados armazenados foram aferidos por meio da linguagem de programação *Python*.

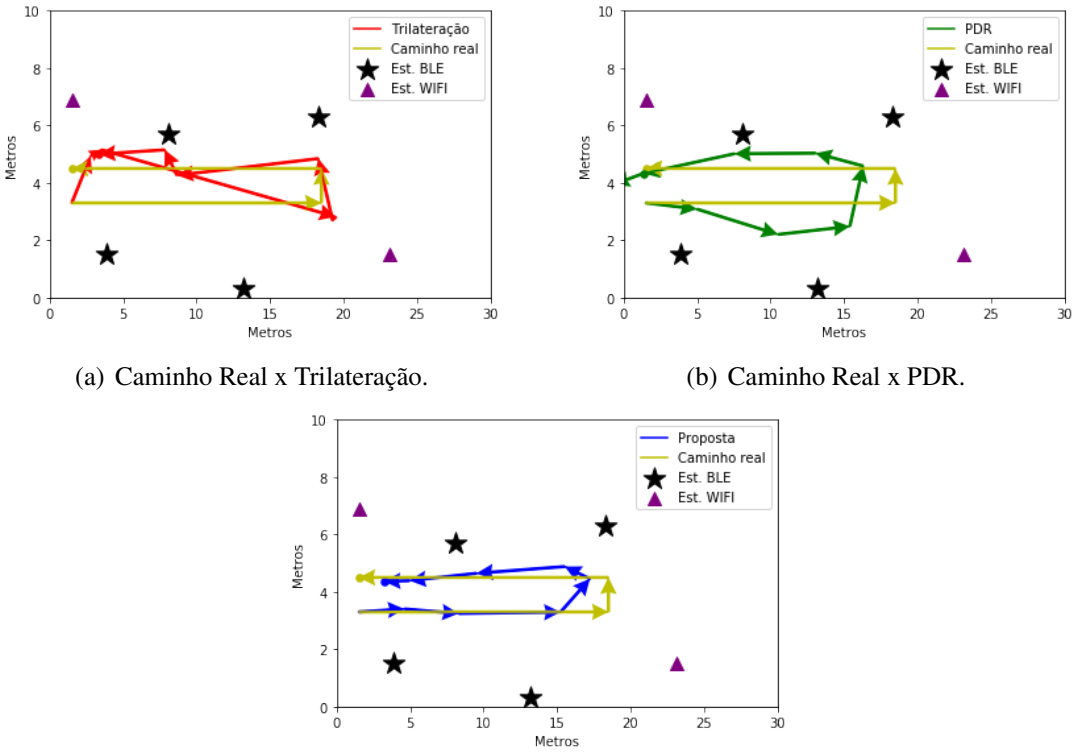
Foram utilizadas como âncoras: 2 estações *Wi-Fi* Cisco Aironet 1040; e 4 Sensor Tag CC1350 como estações *BLE*. Os testes ocorreram em uma área aberta com 300 m^2, com existência de cadeiras, mesas e passagem de pessoas.

A avaliação do método segue como um comparativo de desempenho entre a proposta e os outros métodos isolados. Nesta avaliação, um usuário portando a plataforma em sua mão segue por um dado caminho. Com os dados coletados pelo dispositivo, cada método irá gerar uma coordenada á cada 5 segundos, formando um trajeto que segue exemplificado nas pelas Figuras 4(a), 4(b) e 4(c).

Em uma análise inicial, é observado o afastamento do método PDR em relação ao trajeto real do usuário ao longo do trajeto estimado. Entretanto, ao associar os dados da Trilateração, mesmo com menor precisão, recondiciona a direção do rastreio do trajeto.

Para uma análise mais complementar, é analisado o desempenho dos métodos pela precisão, comparando a distância euclidiana entre a coordenada real e a coordenada estimada de cada formulação. Na avaliação, foram levantados dados de 20 trajetos idênticos as das Figuras 4(a), 4(b) e 4(c), utilizando como parâmetro avaliativo a posição final do usuário no trajeto. São levantados os erros máximo, mínimo e médio de precisão (em metros) de cada método, registrados na Tabela 1.

Como pode ser observado, o método proposto demonstra uma melhor aproximação das coordenadas geradas em comparação aos outros métodos. O método PDR possui um menor erro a curto prazo, entretanto os erros acumulativos deturpam o posicionamento real do alvo. Embora apresente menor precisão, a trilateração fornece



(a) Caminho Real x Trilateração.

(b) Caminho Real x PDR.

(c) Caminho Real x Método proposto.

Figure 4.

novos referenciais para utilização no método proposto. A combinação dos dados resultam em estimações das coordenadas do alvo mais próximas do caminho real.

Uma vantagem a ser destacada do método em relação a outras metodologias descritas na seção 2, está quanto aos esforços de recalibração, graças a não necessidade de projetar um mapa de RSSI do ambiente, principalmente em grandes granularidades. No método é ocasionado apenas a necessidade de atualização das coordenadas das âncoras e seus parâmetros em situações de mudanças do ambiente.

Table 1. Erros de estimação (em metros) de cada metodologia.

Método	Máximo	Minimo	Média
Trilateração	5,872 m	1,258 m	2,767 m
PDR	4,216 m	0,801 m	2,384 m
Proposta	3,188 m	0,417 m	1,224 m

5. Conclusão e Trabalhos Futuros.

Este trabalho discutiu e apresentou uma proposta de metodologia de localização *indoor* para smartphones. Resultados preliminares indicaram um bom desempenho da metodologia, a partir da reorientação do método PDR por novas referências. Entretanto, em alguns casos, o método proposto apresentou desempenho inferior em algumas a uma das outras resoluções, indicando uma necessidade de refinamento do projeto. Assim, parte dos futuros esforços do projeto estarão voltados a uma melhoria na sintonia do Filtro de Kalman.

O aprimoramento do filtro seguirá por meio do controle dos parâmetros R e Q, sendo adicionado o uso de outras versões do Filtro de Kalman.

References

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376.

Anton, H. and Busby, R. (2011). *Álgebra Linear Contemporânea*. Bookman.

Chen, Z., Zhu, Q., and Soh, Y. C. (2016). Smartphone inertial sensor-based indoor localization and tracking with ibeacon corrections. *IEEE Transactions on Industrial Informatics*, 12(4):1540–1549.

Chen, Z., Zou, H., Jiang, H., Zhu, Q., Soh, Y. C., and Xie, L. (2015). Fusion of wifi, smartphone sensors and landmarks using the kalman filter for indoor localization. *Sensors*, 15(1):715–732.

Correa, A., Barcelo, M., Morell, A., and Vicario, J. L. (2017). A review of pedestrian indoor positioning systems for mass market applications. *Sensors*.

Gomes, R., Alencar, M., Fonseca, I., and Lima Filho, A. (2013). Desafios de redes de sensores sem fio industriais. 4:16–27.

Jin, Y., Toh, H.-S., Soh, W. S., and Wong, W.-C. (2011). A robust dead-reckoning pedestrian tracking system with low cost sensors. pages 222–230.

Junsheng, H. (2017). Wireless industrial indoor localization and its application. Master's thesis, The Artitic University of Norway.

Kent, L. T. (2017). Distância média do passo de corrida. `http://www.ehow.com.br/distancia-media-passo-corrida-info_8070/`.

Khalajmehrabadi, A., Gatsis, N., and Akopian, D. (2017). Modern wlan fingerprinting indoor positioning methods and deployment challenges. *IEEE comumunications Surveys & Tutorials*, 19º:1974–2002.

Koo, B., Lee, S., Kim, S., and Sin, C. (2014). Integrated pdr/fingerprinting indoor location tracking with outdated radio map. pages 1–5.

Liu, J., Chen, R., Pei, L., Guinness, R., and Kuusniemi, H. (2012). A hybrid smartphone indoor positioning solution for mobile lbs. *Sensors*, 12(12):17208–17233.

Makki, A., Siddig, A., Saad, M., and Bleakley, C. (2015). Survey of wifi positioning using time-based techniques. *Computer Networks*, 88:218 – 233.

Milette, G. and Stroud, A. (2012). *Professional Android Sensor Programmiing*. Wrox.

Ojeda, L. and Borenstein, J. (2007). Personal dead-reckoning system for gps-denied environments. pages 1–6.

Radu, V. and Marina, M. K. (2013). Himloc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced wifi fingerprinting. pages 1–10.

Rappaport, T. S. (2001). *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA, Prentice Hall PTR, 2nd edition.

Criação de Modelo para Simulação de Movimentação de Ônibus a Partir de Dados Reais*

Melissa Wen[1], Thatiane de O. Rosa[1,3], Mariana C. Souza[2], Robson P. Aleixo[1], Camilla Alves[1], Lucas Sá[1], Eduardo Felipe Zambom Santana[1], Fabio Kon[1]

[1]Universidade de São Paulo (USP)

[2]Universidade Federal de Mato Grosso do Sul (UFMS)

[3]Instituto Federal de Educação, Ciência e Tecnologia do Tocantins (IFTO)

{wen,thatiane,lucassa,efzambom,fabio.kon}@ime.usp.br

mariana.caravanti@aluno.ufms.br, robson.aleixo@optimumsolucoes.com,

camilla.almeida.silva@usp.br

Resumo. *A dinâmica socioespacial de uma cidade sofre constantes mudanças ao longo do tempo. Por consequência, a malha viária e o sistema de transporte público precisam de otimizações contínuas para atender às demandas dos cidadãos. Nesse contexto, uma boa alternativa para reduzir custos e impactos na avaliação de soluções é o emprego de simuladores que utilizem modelos consistentes com a realidade. Em vista disso, processamos dados de deslocamento e de planejamento do sistema de ônibus de São Paulo para melhorar o modelo de movimentação de ônibus usado pelo InterSCSimulator, um simulador altamente escalável para cidades inteligentes. Apresentamos um modelo de movimentação baseado em dados reais do serviço de ônibus de São Paulo a fim de tornar o simulador mais eficaz ao recriar cenários de mobilidade urbana.*

Abstract. *The socio-spatial dynamics of a city undergoes constant changes over time. Consequently, the road network and the public transport system need continuous optimization to meet citizen demands. An alternative to reduce costs and impacts on evaluation of solutions is the use of simulators and models consistent with reality. Considering that, we processed vehicle tracking data and bus system planning information of São Paulo to improve the bus movement model used by InterSCSimulator, a highly scalable simulator for smart cities. In this paper, we present a mobility model based on real data from the São Paulo bus service to make the simulator more effective when recreating urban mobility scenarios.*

1. Introdução

Áreas metropolitanas vêm crescendo em todo o mundo e, consequentemente, expandindo em extensão e população [1, 2]. Tal crescimento influencia diretamente as redes de transporte, gerando problemas como congestionamentos, longos tempos de espera para deslocamento e poluição [1, 3]. Para atacar tais problemas é importante investir em soluções

*Este trabalho foi desenvolvido no contexto do INCT da Internet do Futuro para Cidades Inteligentes, apoiado pela FAPESP proc. 2014/50937-1 e CNPq proc. 465446/2014-0.

inteligentes de mobilidade urbana, no planejamento e melhoria do transporte público [4]. Planejar e testar soluções de mobilidade urbana em ambientes reais são atividades complexas, que envolvem altos custos e causam grandes impactos [1, 5]. Nesse contexto, os simuladores apresentam-se como uma boa ferramenta aos gestores de redes de transportes, pois torna possível testar cenários, sem a necessidade da construção de uma infraestrutura física [4]. Entretanto, para que os simuladores atinjam os seus objetivos é fundamental o desenvolvimento de modelos coerentes com a realidade do ambiente estudado [6].

Para preencher as lacunas de escalabilidade e usabilidade dos simuladores de cidades inteligentes, foi criado o InterSCSimulator - um simulador fácil de utilizar e capaz de reproduzir áreas metropolitanas completas a partir de um mapa contendo dezenas de milhares de ruas e milhões de veículos em movimento [4]. Porém, os modelos utilizados por esse simulador ainda não levam em consideração as estocasticidades da realidade do sistema de transporte coletivo. Diante disso, apresentamos neste trabalho um modelo que visa incorporar alguns aspectos estocásticos às simulações de movimentação dos ônibus. Para desenvolvê-lo, utilizamos dados AVL (*Automatic Vehicle Location*) e de planejamento do sistema de transporte público da cidade de São Paulo. Por fim, validamos nosso modelo por meio de análises comparativas entre o comportamento observado a partir de dados reais de movimentação e o comportamento observado a partir de eventos simulados.

Este artigo está organizado da seguinte forma: na Seção 2 apresentamos conceitos de cidades inteligentes e simulação de mobilidade urbana. Na Seção 3 são apresentados alguns trabalhos relacionados. Na Seção 4 a metodologia do estudo é relatada. Na Seção 5 o modelo de movimentação de ônibus é descrita. A Seção 6 apresenta discussões sobre a validação do modelo. Por fim, a Seção 7 aponta as conclusões e trabalhos futuros.

2. Cidades Inteligentes e Simulação de Mobilidade Urbana

O conceito Cidade Inteligente é bastante amplo e possui um conjunto diversificado de termos considerados sinônimos [7, 8, 9]. Diante disso, adotaremos uma definição mais coerente com este estudo, apresentada por Marsal-Llacuna et al. [10]. Segundo os autores, cidades inteligentes são aquelas construídas com o melhor uso de dados, informações e tecnologias de informação (TI) para monitorar e otimizar a infraestrutura existente, fornecer serviços mais eficientes aos cidadãos, aumentar a colaboração entre diferentes atores econômicos e incentivar modelos de negócios inovadores nos setores público e privado. De forma genérica, tal conceito envolve encontrar soluções para lidar com desafios das cidades a partir da utilização de tecnologias da informação e comunicação [2].

Dentre os desafios enfrentados pelas cidades, um dos mais relevantes está relacionado ao domínio de mobilidade [3, 8]. Nesse domínio, problemas como poluição, fluxo de veículos, congestionamento e longos tempos de espera, impactam diretamente a economia da cidade, meio ambiente e a qualidade de vida dos seus cidadãos [3]. Com isso, Neirotti et al. identificam que os estudos sobre mobilidade inteligente possuem três principais objetivos: otimizar a logística e o transporte em áreas urbanas considerando condições de tráfego e consumo de energia, fornecer aos usuários informações dinâmicas e multimodais para obter um fluxo de veículos mais eficiente e, garantir transporte público sustentável por meio de combustíveis ecológicos e sistemas de propulsão inovadores [11].

Diante disso, planejar e testar soluções de mobilidade urbana mostram-se atividades complexas e desafiadoras, devido aos custos (financeiros, de energia e de tempo) e

impactos (sociais, políticos, infraestrutura e meio ambiente) gerados [1, 5]. Como uma estratégia mais viável aos gestores públicos e privados de redes de transportes, é possível simular cenários para testar diferentes soluções envolvendo vários domínios como fluxo de veículos, transporte público e utilização de recursos [4]. No entanto, Ros et al. afirmam que os resultados dessas simulações são de fato significativos apenas quando modelos coerentes com a realidade são processados [6]. De Dios Ortúzar et al. explicam ainda que para projetar modelos que representem contextos futuros é necessário utilizar as variáveis de estudo do presente [5]. Logo, os modelos devem receber, como entrada, dados do cenário atual (ano-base) e possíveis dados do cenário idealizado (dados projetados).

Neste estudo, adotamos o InterSCSimulator, um simulador escalável e de código aberto desenvolvido para o contexto de Cidades Inteligentes. Ele é capaz de simular cenários de trânsito com milhões de agentes, usando um mapa real de uma grande cidade [4] e possui quatro componentes principais, denominados: *Definição do Cenário*, responsável por ler os arquivos de entrada e criar o grafo da cidade e os veículos que serão simulados; *Motor de Simulação* onde são executados os algoritmos e os modelos de simulação; *Visualização do Mapa* que utiliza a saída do Motor de Simulação e o mapa da cidade simulada para criar uma animação visual da movimentação dos veículos no grafo da cidade; e *Visualização de Gráficos* que também utiliza a saída do Motor de Simulação e gera diversos gráficos com análises sobre o trânsito da cidade (Figura 1). Partindo desses quatro componentes, temos que qualquer modelo a ser avaliado com o simulador precisa descrever o mapa da cidade e as viagens a serem simuladas para definição de cenários. A partir disso, o resultado do modelo pode ser visto por meio de animação visual e gráficos.

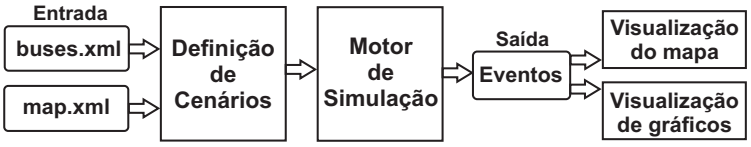


Figura 1. Componentes do InterSCSimulator [4].

Após definir os principais conceitos que sustentam esta pesquisa, a próxima seção explora alguns estudos científicos que analisam aspectos de mobilidade urbana e simulação que vão ao encontro do objetivo aqui proposto.

3. Trabalhos Relacionados

Uma série de estudos analisam diferentes aspectos da mobilidade urbana em cidades inteligentes. Ao considerar o objetivo deste artigo, duas frentes de pesquisa são importantes:

Mobilidade de ônibus nas cidades: estudos que analisam características das viagens dos ônibus de uma cidade com base em dados capturados do GPS (Sistema de Posicionamento Global) dos veículos (denominados AVL) e de outros sensores como smartphones dos usuários. Zhou et al. realizaram um estudo com o objetivo de prever o número de passageiros e o tempo de chegada de ônibus [12]. Para isso, construíram uma base de aprendizagem a partir de um conjunto de modelos de predição, que processava os dados de GPS dos ônibus e de um aplicativo desenvolvido para os usuários do serviço.

Rahman et al. buscaram prever, em tempo real, o horário de chegada dos ônibus [13] a partir da análise das distâncias entre pontos de medição do GPS, denominados "pseudo-horizontes". Para avaliar as variações nos tempos das viagens, consideraram

o comportamento de acordo com o desvio-padrão, coeficiente de variação e obliquidade. Yu et al. visaram prever a ocorrência de situações em que dois ônibus de uma mesma linha chegam a uma mesma parada quase que simultaneamente, fenômeno conhecido como *"bus bunching"* [14]. Para detectar esse fenômeno utilizaram a regressão LS-SVM (*Least Squares Support Vector Machine*), com dados temporais e espaciais dos passageiros.

Simuladores de mobilidade urbana: uma série de simuladores de trânsito consideram diferentes agentes (veículos, pessoas, sensores) e utilizam dados reais para simular cenários de mobilidade urbana. Como exemplo, temos o MATSim e o DTALite. Entretanto nenhum desses simuladores é capaz de reproduzir uma área metropolitana completa contendo um mapa com milhares de ruas e milhões de veículos em movimento [4]. Para preencher essa lacuna, utilizamos neste trabalho o InterSCSimulator - um simulador escalável, fácil de utilizar, paralelizável e distribuido - capaz de reproduzir cenários complexos de Cidades Inteligentes.

4. Metodologia

O InterSCSimulator tem o potencial de agregar múltiplas contribuições aos métodos tradicionais de modelagem em planejamento de transportes. Visando dotar o simulador de maior precisão na descrição da oferta de ônibus, desenvolvemos um modelo de movimentações de ônibus com base em dados GTFS (*General Transit Feed Specification*) de planejamento e dados de AVL do sistema de transporte público da cidade de São Paulo. Nossos estudos foram guiados pelas seguintes questões de pesquisa:

RQ1. *"Quais dados de AVL e de planejamento de transporte público são necessários para desenvolver um modelo que descreva de maneira mais realista a movimentação de ônibus em uma cidade?"*

RQ2. *"Como esses dados devem ser sintetizados para simular adequadamente a movimentação das linhas de ônibus de uma cidade?"*

4.1. Estudo de Caso

São Paulo é uma cidade com grandes dimensões populacionais, espaciais e, consequentemente, grandes desafios de mobilidade urbana. Atualmente, sua frota de ônibus possui cerca de 14,4 mil veículos deslocando 6 milhões de passageiros por dia útil. Em 2017, as suas mais de 2 mil linhas de ônibus atenderam um total de 1.630.604.027 passageiros, segundo a SPTrans, empresa responsável por gerir o sistema de ônibus da cidade.

Para aprimorar as simulações, incorporamos ao modelo utilizado pelo InterSCSimulator informações reais de circulação dos ônibus. Para isso, foram utilizadas duas fontes de dados: GTFS e AVL. Os dados de GTFS refletem o planejamento do serviço, do qual extraímos dados que representam informações estáticas do sistema, tais como: linhas de ônibus, itinerário, localização das paradas de ônibus e o trajeto realizado de uma parada para a outra. Esses dados são abertos e podem ser acessados por meio da página web da SPTrans [1].

Os dados de AVL representam o comportamento real dos ônibus, ou seja, os horários reais de início de circulação, frequências das saídas, assim como velocidade média de deslocamento de cada veículo. Essas informações são obtidas via tecnologia

[1] http://www.sptrans.com.br/desenvolvedores/GTFS.aspx

GPS, e foram fornecidas para este trabalho pela Scipopulis [2], uma *startup* que presta serviços e desenvolve produtos a partir de dados coletados por meio da API do sistema de monitoramento de transporte "Olho Vivo"[3]. Devido à experiência da *startup* em mobilidade urbana, tais dados estavam pré-processados, limpos e relacionados de forma a facilitar o entendimento e extração das informações mais relevantes para construção do modelo. Além disso, alguns erros de geolocalização estavam mitigados, o que favoreceu a precisão do modelo desenvolvido.

Para este trabalho, a *startup* Scipopulis disponibilizou registros diários de monitoramento de uma semana típica (22 a 28 de outubro de 2017), ou seja, 7 dias corridos onde não houveram registros de eventos que impactassem o funcionamento do transporte coletivo. Além desses, foram disponibilizados dados de outros 3 dias considerados atípicos: véspera de feriado prolongado (13 de abril de 2017), clássico de futebol com lotação máxima no estádio Morumbi (24 de setembro de 2017) e intensos protestos pela capital (15 de março de 2017). A combinação dos dados AVL e GTFS trouxe informações da dinâmica de movimentação de 14.139 ônibus e 2.183 linhas no ano de 2017.

4.2. Análise de Dados e Validação do Modelo

Tratando-se da primeira iniciativa de modelagem de dados de AVL e GTFS para utilização no InterSCSimulator, optamos por dotar o modelo de três informações principais: (a) *itinerário de todas as linhas*, representado como arestas entre paradas consecutivas; (b) *velocidades médias das arestas por intervalo de hora*; (c) *o intervalo médio de saída dos ônibus dos terminais por intervalo de hora* para todas as linhas e seu início de circulação;

Assumimos como hipótese que a informação da velocidade média em cada aresta permite que o modelo seja capaz de representar a movimentação de ônibus, incorporando os efeitos da imprevisibilidade e não homogeneidade do mundo real. Tais efeitos implicam em variações nos tempos totais de viagem de uma linha. Também adicionamos as linhas de ônibus representadas no modelo informações referentes à frequência de saída de seus veículos dos pontos terminais. Com isso, busca-se que o modelo reflita o número de veículos da frota em circulação ao longo do dia. Como análise exploratória das fontes de dados, averiguamos a qualidade das informações de circulação com base no registro diário do início de operação dessas linhas, registro de saída do terminal e registro completo do deslocamento dos seus veículos. Também verificamos a compatibilidade entre as frequências planejadas (dados de GTFS) e as observadas (dados de AVL).

A partir da qualidade dos registros, elegemos duas linhas que fazem parte do deslocamento diário da população de baixa renda (residem na periferia e trabalham no centro) para comparar o comportamento real e simulado. O processo de escolha também levou em consideração o estudo da variação dos tempos de viagem como critério de validação do modelo. Dado que, linhas que possuem viagens longas são mais sucetíveis a variações, logo mais adequadas para testar a fidelidade do modelo à realidade, as seguintes rotas foram eleitas: 856R-10-*Socorro/Lapa* e 4311-10-*Term. S. Mateus/Term. Prq. D. Pedro*.

Consideramos a qualidade das informações levantadas acima para validação simplificada do modelo. Assim, para realizar as simulações, elegemos o dia 26 de outubro de 2017, uma quinta-feira na qual verificou-se a ausência de eventos atípicos na cidade que

[2]https://www.scipopulis.com/
[3]http://olhovivo.sptrans.com.br/

pudessem gerar impactos significativos na circulação dos ônibus. Realizamos primeiramente o processamento dos dados AVL para obter as durações reais de todas as viagens com registros completos para cada linha em estudo. Posteriormente, para essas mesmas linhas, verificamos no simulador as durações de todas as viagens iniciadas ao longo das 24 horas do dia. Por fim, comparamos os valores obtidos nesses dois processos anteriores para verificar a similaridade entre os valores de AVL e os resultantes no simulador, aplicando o *Teste t de Student* para validar a eficácia do modelo na representação da realidade.

5. Resultados

A partir dos dados de AVL e GTFS do estudo de caso, construímos um modelo que simula a movimentação diária de 14.139 veículos, operando em 2.183 linhas de ônibus, a fim de representar 8 cenários: 7 dias da semana e 1 dia atípico (véspera do feriado de páscoa). Tais modelos foram desenvolvidos a partir de um conjunto de modificações nos atributos dos arquivos originais de entrada do InterSCSimulator e estão disponíveis na página do projeto InterSCity [4].

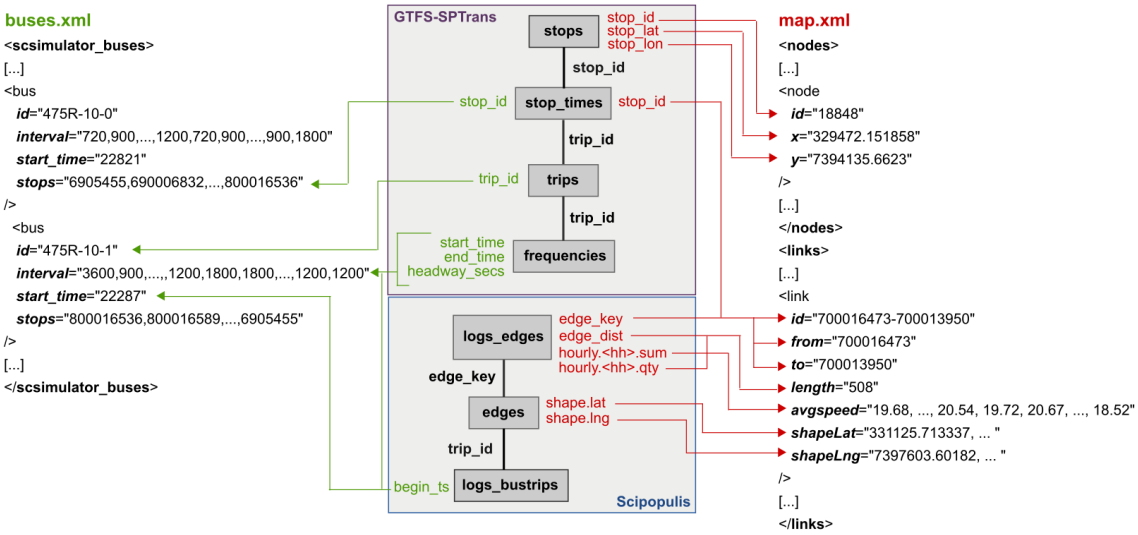


Figura 2. Relacionamento das bases de dados para construção do modelo.

Conforme descrito na Figura 1, o modelo de movimentação dos ônibus é representado por dois arquivos no formato XML (*eXtensible Markup Language*), denominados *map* e *buses*. O arquivo *buses*, descreve as linhas de ônibus do sistema de transporte coletivo *bus* e caracteriza todas as viagens de ônibus que serão realizadas durante a simulação. Já o arquivo *map*, define a malha viária do sistema de ônibus, onde cada parada de ônibus é representada pela marcação *node* e as arestas que ligam duas paradas são representadas por *links*. A Figura 2 apresenta a estrutura desses dois arquivos de entrada e a maneira como seus atributos se relacionam com as informações das fontes de dados.

Considerando o modelo original, modificamos a sua arquitetura para permitir ao simulador representar o comportamento não-homogêneo do serviço de ônibus da cidade ao longo do dia, de forma que seus atributos passam a ter valores variáveis por intervalo de hora. De maneira resumida, os atributos de frequência de saída dos ônibus deixaram de variar apenas em termos de períodos de pico e não-pico, e passaram a variar por intervalo

[4]http://interscity.org/software/interscsimulator/

buses.xml			
buses			
id	Identifica unicamente uma linha		
interval	Lista de 24 posições que armazena a frequência de saída do terminal, em segundos, dos seus ônibus por faixa de hora		
start_time	Horário em que a linha começou a funcionar		
stops	Lista de ID das paradas de ônibus que fazem parte do seu itinerário de viagem		
map.xml			
nodes		**links**	
id	Identificador único da parada de ônibus	id	Identificador único formado pela união do id do *node* origem e *node* destino.
lat	Localização geográfica latitudinal	to	Id da parada de ônibus de destino
lng	Localização geográfica longitudinal	from	Id da parada de ônibus de origem
		length	Distância em metros entre as paradas de origem e destino
		avgspeed	Lista de 24 posições, representando as 24 horas do dia, onde são armazenadas a velocidade média dos ônibus que transitaram no *link*
		shapeLat shapeLng	Coordenadas latitudinais e longitudinais de pontos de referência que estão contidos dentro do espaço que compõe o *link* e determinam a forma do trajeto

Tabela 1. Descrição dos atributos que compõem o modelo.

de hora. Ainda com o processamento dos dados, o deslocamento dos veículos de uma parada de ônibus a outra deixou de simular o melhor caso (velocidade máxima da via e melhor trajeto) e passou a retratar, por faixa de hora, a velocidade média real do percurso real entre duas paradas. A descrição detalhada dos atributos é apresentada na Tabela 1.

Algoritmo 1: Model_generator

Entrada: Dados AVL (edges, logs_bustrips, logs_edges_speed) e dados GTFS(frequencies, stop_times, trips)
Saída: Arquivos buses.xml e maps.xml

```
#Definição do Modelo
modelo : { bus : { id, start_time ← 0
                   interval ← lista[24]
                   stops ← lista  }
           node : { id, lat, lng ← 0 }
           link :  { id, from, to, length ← 0
                     avgspeed ← lista[24]
                     shapeLat, shapeLng ← lista }
         }

Função Principal( )
    dataRef ← <data do modelo_gerado>
    #Gera arquivo buses.xml
    para cada elemento viagem em GTFS.trips faça
        onibus ← novo modelo.bus
        onibus.id ←  viagem.trip_id
        #Obtém horário inicial de saída do ônibus
        primeiraViagem ← encontrePrimeiraViagem(AVL.logs_bustrips, onibus.id, dataRef)
        onibus.start_time ← primeiraViagem.begin_ts
        onibus.interval  ← calculaIntervaloSaidaPorHora(AVL.logs_bustrips, onibus.id, dataRef)
        onibus.stops  ← crieItinerario(GTFS.stops, onibus.id)
        escreva objeto onibus no arquivo buses.xml

    #Gera arquivo Maps.xml
    paradasDeOnibus ← coleteInformacoesParadas(GTFS.stops)
    escreva paradasDeOnibus no arquivo maps.xml
    to, from ← string
    para cada elemento parada em GTFS.stop_times faça
        se parada é o primeiro ponto do itinerário de uma viagem faça
            from ← parada.id
        senão faça
            to ← parada.id
            aresta ← concatene(from, to)
            from ← parada.id
            arestaId ← aresta
            #Obtém velocidade média em cada edge por hora
            arestaVel ← ColeteVelocMediaArestasPorHora(AVL.logs_edges_espeed, arestaId, dataRef)
            #Obtém estrutura das arestas
            arestaEstr ← coleteInfoAresta(AVL.edges, GTFS.stop_times, arestaId, arestaVel, dataRef)
            escreva  arestasEstr no arquivo maps.xml

Função calculaIntervaloSaidaPorHora(AVL.logs_bustrips, onibus.id, dataRef)
    contaOnibus, intervalo ← lista[24] de inteiros
    encontre em AVL.logs_bustrips todos os elementos tal que \
     date = dataRef e trip_id = onibus.id
        #Ponto inicial da viagem
        para todos os elementos viagem em AVL.logs_bustrips onde \
         begin_stop_seq = 0 faça
            horaReferencia ← viagem.begin_ts.hour
            incremente valor de contaOnibus[horaReferencia]
    #Cálculo das frequências de saída para cada hora do dia
    intervalo ← calcFrequenciasPorSegundo(contaOnibus)
    retorne intervalo

Função ColeteVelocMediaArestaPorHora(AVL.logs_edges_espeed, arestaId, dataRef)
    velocMedia  ← lista[24]
    encontre o elemento aresta em AVL.logs_edges_speed tal que \
     date = dataRef e edge_key = idAresta
        idAresta ← aresta.edge_key
        para cada elemento hora em aresta.hourly faça
            #Cálculo de velocidade média por hora
            velocMedia[hora] ← (aresta.edge_dist/(listaVelocPorHora[hora].sum /
             listaVelocPorHora[hora].qty))
    retorne velocMedia
```

Figura 3. Algoritmo utilizado na construção do modelo de simulação.

A Figura 3 apresenta o algoritmo utilizado no processamento das informações das bases de dados para construção do modelo. Além disso, um vídeo ilustrando a simulação (no InterSCSimulator) das linhas adotadas para validação do modelo foi gravado e disponibilizado no YouTube [5].

6. Discussão

De acordo com a estratégia definida na Seção 4.2, foram eleitas duas linhas de ônibus para validação simplificada do modelo. A comparação dos tempos de viagem é ilustrada

[5] www.youtube.com/watch?v=UO4kFBNqhXY&feature=youtu.be

na Figura 4 e refere-se aos horários entre 6:00 e 23:59 do dia 26 de outubro de 2017.

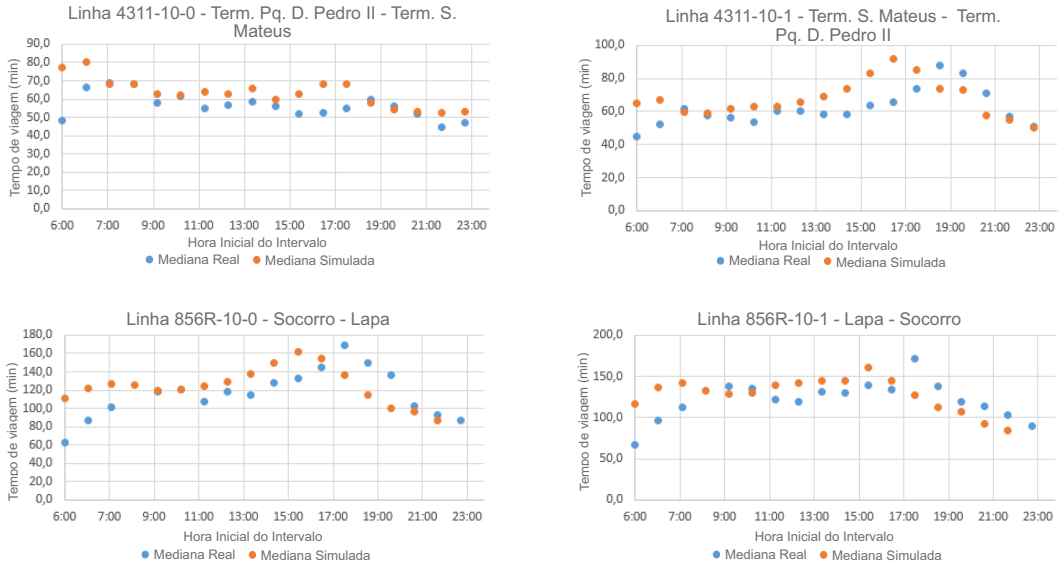


Figura 4. Medianas das durações das viagens reais e simuladas.

Ainda ao analisar os gráficos da Figura 4, podemos verificar que as medianas dos tempos de viagens reais possuem valores maiores em horários de pico, quando comparados aos horários de não-pico. Esse padrão de comportamento também pode ser observado nas medianas dos tempos de viagens simuladas. Sabendo-se que para longas distâncias as velocidades dos ônibus possuem uma distribuição normal [13], também aplicamos o *Teste t de Student* para verificar a similaridade entre as durações das viagens reais e simuladas.

Teste t de Student 856R-10	Real (min)	Simulado (min)
Média	120	127
Número de Observações	57	57
Hipótese nula- Diferença de Médias igual a zero	0	
Valor de t	112	
Graus de Liberdade	-1,53	
P(T<=t) duas-caudas	0,13	
t crítico duas-caudas	1,98	

Teste Z 4311-10	Real (min)	Simulado (min)
Média	60	64
Número de Observações	70	70
Hipótese nula- Diferença de Médias igual a zero	0	
Valor de Z	-1,87	
P(T<=t) duas-caudas	0,06	
t crítico duas-caudas	1,96	

Figura 5. Aplicação de teste de hipótese para comparação dos tempos de viagens das linhas 865R-10 e 4311-10

A Figura 5 apresenta os resultados dos testes de hipótese realizados para as duas linhas em questão. Na primeira linha, 856-10, foram observadas ao todo 57 viagens. Nos dados de AVL, tais viagens obtiveram tempo médio de duração de 120 minutos, contra 127 minutos obtidos nos dados simulados. Uma vez que ambas as amostras apresentaram distríbuições normais com variâncias não diferentes, aplicamos o teste de hipótese *t de student* a fim de validarmos a efetividade do modelo. Podemos observar que o valor de t igual a -1,53 encontra-se dentro do intervalo de confiança, nos garantindo que as médias

de tempo não são diferentes. Já para a segunda linha, 4311-10, foram observadas 70 viagens onde o tempo médio de duração real foi de 60 minutos, contra 64 minutos obtidos pelos dados simulados, como as amostras não apresentaram uma distribuíção normal, para a validação do modelo, aplicamos o teste Z. Nesse caso, notamos que o valor de z, -1,87, encontra-se dentro do intervalo de confiança, nos permitindo concluir que as médias de tempo também não são diferentes. Com esses resultados, garantimos a qualidade de representação do modelo e trazemos maior confiabilidade para as análises.

No estágio atual, o modelo emprega a variável estocástica de velocidade média nas arestas por horário. Apesar de se tratar de uma validação preliminar, podemos perceber que o modelo apresentado permite ao simulador reproduzir as tendências de duração das viagens por faixa de horário, em função das velocidades médias de deslocamento entre paradas consecutivas. Entendemos que melhores resultados podem ser alcançados com a inserção de outros fatores que podem afetar os tempos de viagem, como semaforização e embarque/desembarque de usuários. Por fim, também necessitamos expandir esta análise para as demais linhas do sistema de ônibus simulado.

7. Conclusão

Diversas ferramentas são desenvolvidas para apoiar iniciativas de melhoria de mobilidade urbana. Dentre elas está o InterSCSimulator, um simulador escalável que permite execuções paralelas e distribuídas para representar cenários complexos de cidades inteligentes. Para dotar os cenários gerados pelo InterSCSimulator de aspectos da realidade, desenvolvemos, neste trabalho, um modelo de movimentação de ônibus, utilizando dados reais do sistema de transporte de ônibus da cidade de São Paulo.

A construção da arquitetura deste modelo foi possível a partir da resolução de duas perguntas. Para a primeira, *"Quais dados AVL e de planejamento de transporte público são necessários para desenvolver um modelo que descreva de maneira mais realista a oferta de ônibus de uma cidade?*, utilizamos como estudo de caso a cidade de São Paulo. Avaliamos a integridade dos registros AVL dos seus ônibus quando combinados com os dados GTFS do planejamento de transporte. E assim, definimos quais informações podem ser extraídas das fontes de dados, de modo a incorporar à simulação aspectos reais do sistema de transporte coletivo.

Para responder a segunda pergunta, *"Como esses dados devem ser sintetizados para simular adequadamente a movimentação das linhas de ônibus de uma cidade?"*, apresentamos uma arquitetura para o modelo, onde os seus atributos retratam o comportamento não-homogêneo da circulação dos ônibus, por faixa de horário. Nessa arquitetura, os aspectos estocásticos foram incorporados utilizando a média das velocidades médias dos veículos, por seguimento entre paradas de ônibus. Por fim, também acrescentamos a cada seguimento a descrição real do seu formato.

Novas pesquisas devem ser encaminhadas para incorporar outros aspectos estocásticos de mobilidade urbana ao InterSCSimulator e assim, melhorar a sua acurácia ao prever eventos e retratar a realidade. Como trabalho futuro, em um primeiro momento, é importante realizar uma validação mais consistente do modelo gerado. Além disso, entendemos que é relevante incorporar na modelagem mais elementos que incidam sobre os tempos de viagem de cada linha por intervalo de hora, representando os impactos de outras variáveis além da velocidade média. Também consideramos importante a construção

de modelos que representem o comportamento típico dos ônibus da cidade por dia da semana, compilando dados de diferentes datas de um mesmo dia.

Referências

[1] D. Hall, "Integration opportunities at transit jurisdictional borders," Master's thesis, University of Waterloo, 2013.

[2] A. Caragliu, C. D. Bo, and P. Nijkamp, "Smart cities in europe," *Journal of Urban Technology*, vol. 18, no. 2, pp. 65–82, 2011.

[3] C. Benevolo, R. P. Dameri, and B. D'Auria, "Smart mobility in smart city," in *Empowering Organizations* (T. Torre, A. M. Braccini, and R. Spinelli, eds.), (Cham), pp. 13–28, Springer International Publishing, 2016.

[4] E. F. Z. Santana, N. Lago, F. Kon, and D. S. Milojicic, "Interscsimulator: Large-scale traffic simulation in smart cities using erlang," in *The 18th Workshop on Multi-agent-based Simulation - MABS 2017*, 2017.

[5] J. de Dios Ortúzar and L. G. Willumsen, *Modelling Transport*. John Wiley & Sons, 2011.

[6] F. J. Ros, J. A. Martinez, and P. M. Ruiz, "A survey on modeling and simulation of vehicular networks: Communications, mobility, and tools," *Computer Communications*, vol. 43, pp. 1 – 15, 2014.

[7] T. Nam and T. A. Pardo, "Conceptualizing smart city with dimensions of technology, people, and institutions," in *Proceedings of the 12th Annual International Digital Government Research Conference: Digital Government Innovation in Challenging Times*, dg.o '11, (New York, NY, USA), pp. 282–291, ACM, 2011.

[8] R. Giffinger, C. Fertner, H. Kramar, R. Kalasek, N. Pichler-Milanović, and E. Meijers, "Ranking of european medium-sized cities," tech. rep., Vienna University of Technology, 10 2007.

[9] E. F. Z. Santana, A. P. Chaves, M. A. Gerosa, F. Kon, and D. S. Milojicic, "Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture," *ACM Comput. Surv.*, vol. 50, pp. 78:1–78:37, Nov. 2017.

[10] M.-L. Marsal-Llacuna, J. Colomer-Llinàs, and J. Meléndez-Frigola, "Lessons in urban monitoring taken from sustainable and livable cities to better address the smart cities initiative," *Technological Forecasting and Social Change*, vol. 90, pp. 611 – 622, 2015.

[11] P. Neirotti, A. D. Marco, A. C. Cagliano, G. Mangano, and F. Scorrano, "Current trends in smart city initiatives: Some stylised facts," *Cities*, vol. 38, pp. 25 – 36, 2014.

[12] C. Zhou, P. Dai, F. Wang, and Z. Zhang, "Predicting the passenger demand on bus services for mobile users," *Pervasive and Mobile Computing*, vol. 25, pp. 48 – 66, 2016.

[13] M. M. Rahman, S. Wirasinghe, and L. Kattan, "Analysis of bus travel time distributions for varying horizons and real-time applications," *Transportation Research Part C: Emerging Technologies*, vol. 86, pp. 453 – 466, 2018.

[14] H. Yu, D. Chen, Z. Wu, X. Ma, and Y. Wang, "Headway-based bus bunching prediction using transit smart card data," *Transportation Research Part C: Emerging Technologies*, vol. 72, pp. 45 – 59, 2016.

Model-Driven Mobile CrowdSensing for Smart Cities

Paulo César F. Melo, Fábio M. Costa

Instituto de Informática – Universidade Federal de Goiás (UFG)
Caixa Postal 131 – 74.690-900 – Goiânia – GO – Brazil

{paulomelo,fmc}@inf.ufg.br

Abstract. *Making cities smarter can help improve city services, optimize resource and infrastructure utilization and increase quality of life. Smart Cities connect citizens in novel ways by leveraging the latest advances in information and communication technologies (ICT). The integration of rich sensing capabilities in today's mobile devices allows their users to actively participate in sensing the environment. In Mobile CrowdSensing (MCS) citizens of a Smart City collect, share and jointly use services based on sensed data. The main challenges for smart cities regarding MCS is the heterogeneity of devices and the dynamism of the environment. To overcome these challenges, this paper presents an architecture based on models at runtime (M@rt) to support dynamic MCS queries in Smart Cities. The architecture is proposed as an extension of the InterSCity platform, leveraging on its existing services and on its capability to integrate city infrastructure resources.*

1. Introduction

Modern-time cities face challenges to achieve goals related to socio-economic development and quality of life, notably due to the concentration of the population and the pressures that arise from it. The concept of Smart City was proposed in response to these challenges (Celino et al., 2013). One of its main themes is the integration of the physical and virtual worlds (Borgia et al., 2014). This integration is achieved with the introduction of capabilities for environmental sensing and actuation, allowing capture, analysis and processing of real-world data, transforming the data into useful information and allowing autonomic interventions in the urban space. Thus, smart city resources, also called things in an Internet of Things perspective, are equipped with sensing and/or actuation capabilities, along with communication capabilities to share information.

In this context, smart cities need to take advantage not only from information collected from sensors that belong to its infrastructure, but also from the mobile devices owned by its citizens, which increasingly have advances sensing capabilities (e.g. cameras, microphone, accelerometer, GPS). In Mobile CrowdSensing (MCS) citizens of a smart city collect, share and jointly use services based on community-sensed data (Stojanovic et al., 2016).

Smart cities have a wide range of domains, such as MCS, and these domains can be integrated into a complete and consistent solution as part of a software platform, which includes foundation services for the development, integration, deployment, and management smart city applications. In the MCS domain, the development of smart city platforms to support applications poses a number of challenges, such as interoperability among different resources, recruiting of appropriate data sources, collection and processing of data from those sources, and runtime adaptation of the applications in dynamic environments (Alvear et al., 2018).

To help overcome these challenges, this paper proposes an architecture for processing MCS queries in smart cities using an approach based on models at runtime that is integrated as part of an existing smart city platform called InterSCity (Del Esposte et al., 2017). The proposed approach fulfills the following goals: (a) processing

user-defined MCS queries; (b) providing a scalable MCS service; (c) dynamically adapting query processing to changes in the environment, by adjusting the set of selected devices and sensors and by allowing users to alter queries on-the-fly (mainly in the case of long-running queries); and (d) providing composite resources (based on the dynamic combination of crowd-based resources) that applications can use in a transparent way.

In order to demonstrate the feasibility of this approach, we present a scenario for monitoring the noise levels of a city in order to identify critical areas with high levels of noise. Data gathered in this way can be used by environmental control applications (Zappatore et al., 2016).

The rest of the paper is organized as follows. Section 2 presents the state of the art on MCS, a model-driven approach for MCS and its integration with Smart Cities. Section 3 discusses smart city platforms in general and the InterSCity platform in particular. Section 4 describes the proposed architecture, while its implementation is presented in Section 5. Section 6 presents a scenario to demonstrate the functionality of the platform, and Section 7 reviews the main contributions and discusses future work.

2. Mobile CrowdSensing (MCS)

MCS refers to the opportunistic or participatory use of a large set of sensors embedded in current general purpose mobile devices for the purpose of measuring and mapping interesting phenomena by means of the collaborative sharing of sensors (Ganti et al., 2011). MCS environments encompass a variety of applications that need to communicate and exchange data. The major challenges are related to the amount and diversity of devices, the dynamism of the scenarios, and the proper selection of devices to fulfill a given request.

Existing platforms for MCS address challenges such as facilitating application development, supporting efficient and scalable dissemination of sensor data, enabling mobility management of the applications and providing incentives for participatory sensing. However, the programming models in most of these platforms makes it difficult to develop dynamic applications that need to change quickly in the face of changes in the application or its environment. The next section addresses a robust alternative to address this issue.

2.1. Model-Driven Approach for MCS

A model-driven approach to MCS is motivated by the need to overcome the adaptability challenges due to the variety of applications and the mobility of devices. The use of models@runtime in MCS allows the description of the crowdsensing behavior of an application in a dynamic way, thus enabling runtime adaptation of such behavior. In general, the use of a model-based approach enables the shortening of the semantic gap between the problem to be solved and the platform being used, promoting the use of abstractions that are closer to the problem domain.

In this context, CSVM (CrowdSensing Virtual Machine) (Melo, 2014) is a platform driven by models@runtime (Blair et al, 2009) that enables the creation and execution of MCS queries by specifying and interpreting models described in a domain-specific modeling language called CSML (CrowdSensing Modeling Language).

CSVM was implemented as a distributed architecture containing 5 layers and comprised by a central component (CSVMProvider) and a distributed component (CSVM4Dev) which is instantiated on each participating mobile device. In addition to its reliance on modeling techniques, CSVM demonstrates the flexibility in creating queries from high-level models that can be modified at runtime.

CSML is the domain-specific modeling language (DSML) interpreted by CSVM. It allows the creation and manipulation of models that describe queries and their execution. Its constructs are used to model the two major functionalities required by MCS applications, namely device registration, which integrates the device as part of the crowdsensing environment, and query specification, which allows the user to create queries that involve sensor data gathered from multiple devices. These two functionalities are specified in the form of two kinds of submodels, also called schemas: Control Schema (CS) and Data Schema (DS). CS are models that represent logical CrowdSensing configurations and are further subdivided into Environment Control Schemas (ECS) and Query Control Schemas (QCS). The constructs used to specify schemas are defined in the CSML metamodel, which in turn is defined according to OMG's metamodeling architecture, the Meta-Object Facility (MOF) (OMG, 2008).

An ECS describes the crowdsensing environment and serves as a representation of the devices (and sensors) that are available. A QCS is a model at runtime that specifies one or more queries in terms of the desired types, quantities, and location of sensors, as well as the operation to be executed on sensor data (e.g., average, sum, etc.) and the type of notification of sensor data to clients (e.g., following an event-driven approach). As an example, a QCS can be used to describe a query to monitor the noise level in a specific place or region. Finally, a DS is a model that represents an empty form, which specifies the type of sensor information required in a query.

CSML can be used to specify MCS functions in different domains, including Smart Cities. Its constructs are based solely on elements of the MCS technical domain, making it independent of any specific application domain. In this work we are interested in investigating the benefits of CSML's model-driven approach to support MCS applications in the domain of Smart Cities.

2.2. MCS for Smart Cities

MCS fits naturally in smart cities since every citizen, with their mobile phones equipped with a variety of sensors, can be considered a data source in the city. Cooperation between citizens that are part of a crowd enables large-scale sensing tasks. In this context, models architectures are proposed, aiming at a horizontal approach (Petkovics et al., 2015) or even a reference architecture with shelf components also called off-the-shelves (Diniz et al., 2015).

Various platforms for Smart Cities try to incorporate MCS in order to provide a more complete platform solution that involves community (human) and collaborative sensors. Examples are CrowdOut (Aubry et al., 2014), Borja e Gama (Borja et al., 2014) and SOFIA (Filipponi et al., 2010). CSVM in turn is a complete platform to model and process MCS queries, handling the major requirements of the MCS domain. Its strength lies in the use of a model-driven approach to collect, process and store the data from devices in addition to supporting the construction of MCS applications through the use of dynamic models. These features naturally fit in the kind of dynamic environment that is characteristic of smart cities. However, CSVM's architecture does not consider its integration with other services that are required to handle the requirements of smart cities. Examples are the integration of infrastructure sensors and social networks as data sources, as well as the processing of big data that arises from the collection of data from a large number of sources.

3. Platform for Smart Cities

A smart city platform must integrate multiple domains into a complete and consistent middleware solution, providing facilities for the development, integration, deployment, and management of applications (Santana et al., 2017). Building such platforms involves challenges: enabling interoperability between a city's multiples

systems, guaranteeing citizens' privacy, managing large amounts of data, supporting scalability, supporting adaptability of dynamic environment, and dealing with a large variety of sensors. In order to overcome these challenges, several smart city platforms have been developed, such as OpenIoT (Solatos et al., 2015), SMARTY (Anastasi et al., 2013), U-City (Piro et al., 2014), and InterSCity (Batista et al., 2016), which was chosen by this work to present a microservice architecture that allows easy adaptation of new services like MCS and it is discussed next.

3.1 InterSCity

The InterSCity platform has a microservice-based architecture designed as a unified reference architecture for smart cities (Santana et al., 2017). The architecture is shown in Figure 1 as a set of high-level cloud-based (RESTful) services. To provide easy and decentralized communication, each InterSCity microservice has well-defined boundaries to communicate with both IoT devices and smart city applications. Currently, the platform is composed of six microservices that provide: integration with different IoT devices (Resource Adaptor), data and resource management (Resource Catalog, Data Collector and Actuator Controller), resource discovery through context data (Resource Discovery), and graphical interface for visualization (Resource Viewer).

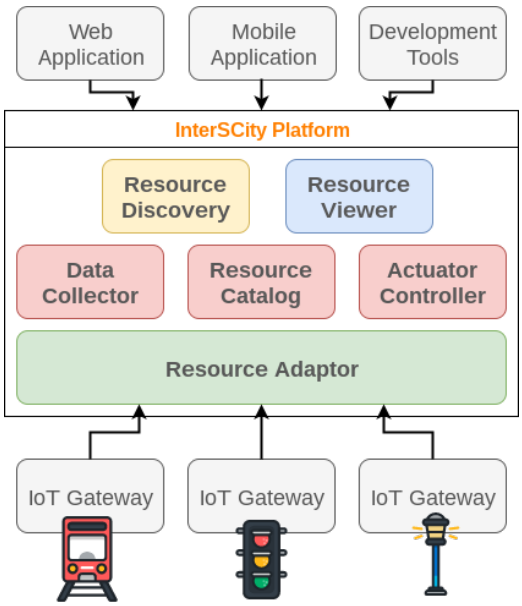


Figure 1. The InterSCity Platform Architecture (Del Esposte et al., 2017).

All microservices are implemented with REST APIs for synchronous messaging over HTTP and RabbitMQ of the Advanced Message Queuing Protocol (AMQP) for assynchronous calls. In addition, another important aspect of the platform is the mapping of each physical entity that makes up the city (cars, buses, lampposts, traffic lights, etc.) to a logical resource. These resources comprise attributes (e.g., location and description) and functional capabilities to provide data and receive commands.

Some design principles were considered when building the InterSCity platform, with emphasis on scalability (in terms of the number of devices, users and components, and volume of city-related data) and evolvability (very dynamic urban environments tend to change constantly in terms of organization, regulations, problems, opportunities and challenges). However, regarding the support for MCS applications (queries) and resources, the InterSCity architecture has two limitations: (1) lack of runtime adaptation of query processing; and (2) lack of transparent support for composite resources (crowd/group of sensors). This paper addresses these limitations with an extension of the

InterSCity platform to support the processing and management of MCS applications through a model-driven approach.

4. MCS Architecture

In general, the architecture of smart city platforms must include components to support the construction of applications, manage and communicate with city network nodes, integrate with existing social networks, store and manage the collected data, and capture context variations and adapt to it (Santana et al., 2017). In line with this generic approach, we propose an architectural extension of InterSCity to support construction and processing of MCS queries according to the model-driven approach used in CSML.

The proposed architecture, shown on the left part of Figure 2, comprises all components already implemented in InterSCity, augmented with a new microservice called CrowdSensing Engine, responsible for processing MCS queries and described next.

4.1. CrowdSensing Engine

The CrowdSensing Engine is a microservice responsible for processing MCS queries. For the construction of this microservice the InterSCity design principles were maintained so that the extension does not compromise the original structure. As such, this component was developed according to the evolution requirements of the platform, maintaining the characteristics of microservices in a way that is weakly coupled, scalable and has well-defined interfaces for external communication. The CrowdSensing Engine has a five internal components, as shown in Figure 2 and described next.

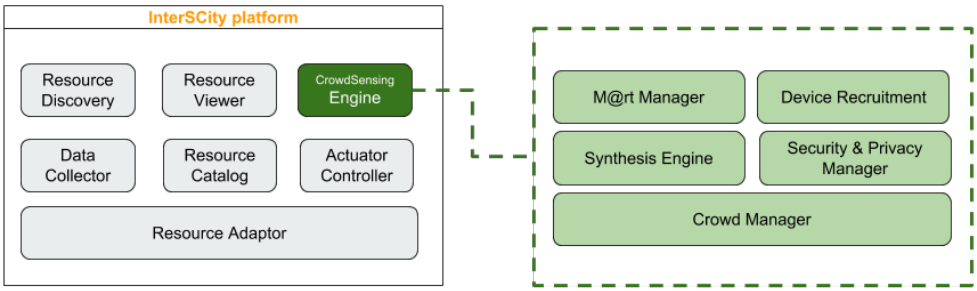


Figure 2. The InterSCity CrowdSensing Engine (left) and its internal components (right).

Crowd Manager. This component is responsible for keeping the crowd always up to date. A crowd represents the set of devices recruited to provide data for a query. Crowd Manager manages the status of such devices to monitor for failures and availability, in which case it interacts with the Device Recruitment component to select another device. In addition, it is also responsible for registering the crowd in the Resource Catalog as a logical resource with a specific capability (e.g., temperature, humidity etc.) making its data available to other applications (not necessarily crowdsensing ones). Note that each query can generate a different crowd, all of which are managed by this component.

Each crowd has a CSML model that represents it and is maintained at runtime, so that changes identified by Crowd Manager trigger commands for the M@rt Manager to update the model. This model is based on the query description in CSML (more specifically a user-generated QCS) and composed of the recruited devices.

M@rt Manager. This component keeps the runtime model up to date. This includes all aspects of the MCS environment, notably query models and crowd models. More specifically, it manages adaptation rules, so that when a rule is triggered, it

communicates the event to the Synthesis Engine and Device Recruitment components for appropriate handling. An adaption rule can be triggered when a device is unavailable, in which case the M@rt Manager must change the model by removing the device and inserting a newly recruited device.

Synthesis Engine. This is the central component of the microservice and all queries must pass through it. It is responsible for interpreting all the models described in CSML and received by the microservice. It has an interface for communication with the other components inside the microservice, and provides a REST API for communication with MCS applications. Therefore, all internal communication with this component is performed through method calls and external calls are carried out via HTTP REST protocol. This is the only component that interacts directly with MCS applications.

As part of model interpretation, it parses a JSON-based input model and converts it into an internal model described in CSML; then it transforms the elements of the CSML model into HTTP commands that carry out the intent expressed in the model.

Device Recruitment. This component manages internal and specific recruitment policies to access CrowdSensing resources. It communicates with Resource Discovery to select resources according of a specific query and to construct a QCS instance. It has a direct communication interface with Resource Discovery. It functions as a broker between the CrowdSensing Engine and the set of cataloged (registered) resources.

Security and Privacy Manager. It is responsible for applying pre-defined privacy and security policies. With regard to security aspects, this component should guarantee confidentiality, availability, integrity, authenticity, non-repudiation, and auditing. To do this, it implements communication protocols that employ encryption and access control through an authentication system and access control lists (ACLs). Privacy is managed based on a set of user-defined rules (usually restrictive) associated with each user, informing access restrictions to specific sensors or by certain types of application.

The remainder of this section describes the interaction protocols that these components follow in order to process MCS queries.

4.2 MCS Query Processing

To perform CrowdSensing for an application, a platform must allow the registration of devices and the subscription (or sending) of CSML queries. **Device Registration** is performed by the generic components of InterSCity as shown Figure 3. In this process, (1) the device sends an HTTP POST command with its capabilities (sensors it wants to share), (2) Resource Adaptor sends the resource meta-data to Resource Catalog, (3) Resource Catalog publishes an event to the RabbitMQ message bus, which may notify (4) the Data Collector and Actuator Controller to inform that the resource has the specified sensor and actuation capabilities.

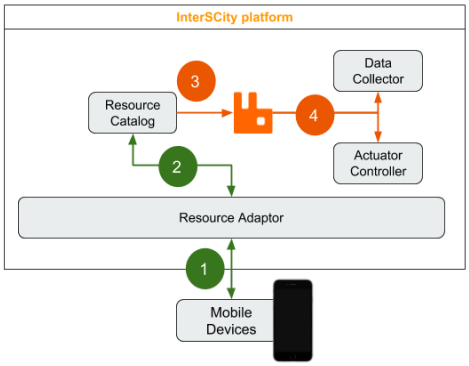


Figure 3. MCS Device Registration in InterSCity platform

Query Subscription (query processing), in turn, directly involves the CrowdSensing Engine microservices as shown Figure 4(a) and 4(b). The following steps describe how components interact during query processing. First, the application describes the query model in CSML and (1) sends the model to the platform through the REST API provided by CrowdSensing Engine (more specifically, via the Synthesis Engine component interface). The Synthesis Engine component performs the parsing, (2) sends the model to the M@rt Manager for storage, and converts the query described in CSML into commands to recruit the devices, which are then send (3) to Device Recruitment.

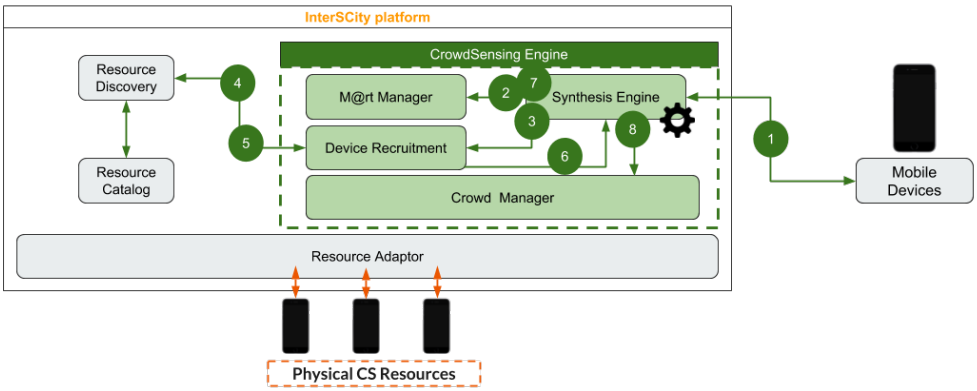


Figure 4(a). MCS Query Processing part 1

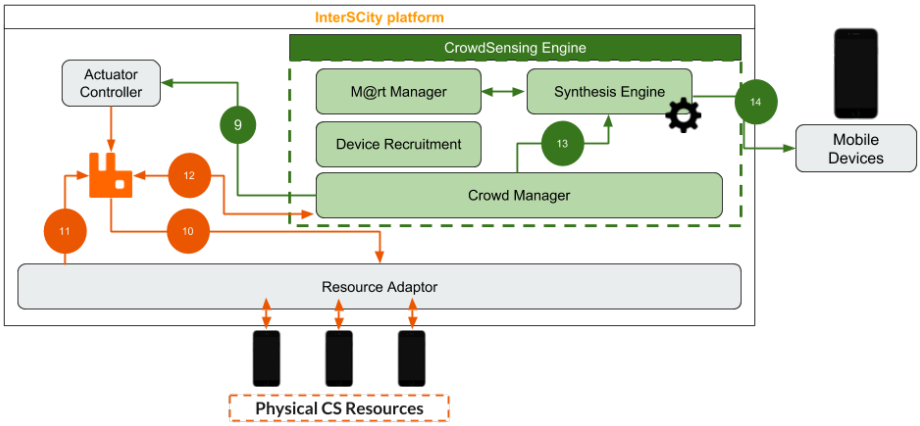


Figure 4(b). MCS Query Processing part 2

Device Recruitment applies policies to access the MCS resources and (4) sends recruitment requests to get data about the recruited resources. If devices are available in accordance with the query, Resource Discovery (5) returns a list of the devices that were recruited, identified by uuid (notation used in InterSCity to identify each cataloged resource). Synthesis Engine (6) receives the list sent by Device Recruitment in JSON, performs parsing, converts the list into a QCS instance (in CSML), (7) sends the up to date the model to M@rt Manager, and (8) sends to Crowd Manager the group of recruited devices (at this point, it creates the group/crowd and an id for it).

Crowd Manager generates and sends commands (9) to Actuator Controller to obtain current data from recruited devices and to generate asynchronous commands to get the status of devices. Resource Adaptor is notified by RabbitMQ and forwards the notification to the devices (10). After data capture, Resource Adapter (11) publishes the data obtained. Crowd Manager (12) receives the notification, consumes the message and (13) sends it to Synthesis Engine, which applies the appropriate business rules and

performs data merging (referring to M@rt Manager). Finally, (14) the response is sent to the requestor.

5. Implementation

In this section, we briefly describe some aspects of a proof-of-concept prototype of the CrowdSensing Engine microservice. The prototype comprises an implementation of the five internal components described in Section 4, which are responsible for processing CSML MCS queries. The interaction between these components are performed through remote calls to methods, as well as through trigger events.

CrowdSensing Engine was implemented in Ruby. Its implementation is encapsulated as a microservice that provides a RESTful interface to communicate with other platform microservices and external applications. For asynchronous calls, we used the RabbitMQ implementation AMQP. More specifically, for the processing of MCS queries, an `MCS' topic was created.

Like the other microservices of the platform, CrowdSensing Engine is encapsulated in its own Docker lightweight container which can be deployed and maintained independently.

6. Example Scenario

In this section we present a scenario of applicability of the proposed approach in order to demonstrate the feasibility of the solution and its ability to meet the major specific requirements for MCS in the smart cities domain. In order to demonstrate the scope of the proposal, this scenario comprises the complete cycle of processing MCS queries applied in the domain of smart cities.

The example application in this scenario corresponds to the monitoring of noise level in an urban space. One of the possible ways for handling this scenario is to offer both citizens and city managers new ways for managing environmental monitoring. Such management can be done by a smart city platform with MCS components that exploit the capabilities of the microphones embedded in citizen's smartphone's as sound sensing devices in order to create large-scale noise maps and suggest city managers suitable noise reduction interventions.

The first step is to specify CSML queries (using an application designed for this purpose) related to the problem domain, informing the sensors (type and quantity) that are required for the monitoring, as well as the location to be monitored (e.g., "*100 audio sensors located in the X neighborhood during 10 hours*"). It is also possible to specify an operation to be performed over the collected data (e.g., average). Once the query is created, it is sent to the InterSCity platform.

The second step occurs within the platform, where the CrowdSensing Engine microservice receives the request and initiates its processing by following the steps described in Section 4.2. After interpreting the query, the CrowdSensing Engine microservice sends commands to discover resources (devices) that meet the requirements specified in the query. At this point, the resources are searched in a catalog of resources maintained by the platform.

A list of 100 devices must be returned to the CrowdSensing Engine to update the query model, which will include the sensor type, location, query duration, and devices that have been recruited. If any of the devices fail or if the user changes the query parameters during its execution, the CrowdSensing Engine identifies the fact and initiates the adaptation process. This process involves updating the model by removing the failed device and/or applying the new requirements stipulated by the user as changes in the model kept at runtime.

Finally, the data obtained are processed and the results are returned to the user, informing in this scenario the noise level of that specified region.

7. Conclusion and Future Work

The development and deployment of smart city platforms faces a number of challenges such as privacy, data management, heterogeneity, communication, scalability, city models and dynamism. Through these challenges there is an increasing need for smart city solutions that take advantage of the latent potential of existing open source platforms.

In this paper, we presented an architecture for processing MCS queries in the smart cities domain. More specifically, we propose an extension of the InterSCity platform to support CrowdSensing applications through the development of a microservice based on a models@runtime approach for MCS. The approach uses the CSML modeling language to model crowdsensing queries, thus supporting the development of applications in this domain.

The main contribution this work is the demonstration of feasibility of integrating MCS in the domain of smart cities using a models@runtime approach. In this way, the approach proposed in this article makes it possible to opportunistically (or in a participative way) use the latent potential of sensors embedded in smartphones through the implementation of the Mobile CrowdSensing paradigm as part of a platform for smart cities. This paradigm also allows to combine different resources in order to generate more precise and useful information for the applications. In addition, this work also includes the use of models@runtime to allow real-time adaptation and improve the modeling of city aspects (through MCS query modeling in CSML) from which it differs from the other platforms mentioned in Section 3.

Our ongoing work includes the performance of experiments to evaluate the scalability and performance of the proposed microservice. For this evaluation, we will consider the time spent by each internal component of the microservice, in order to identify potential bottlenecks and propose improvements in its implementation.

Acknowledgement

This research is part of the INCT of the Future Internet for Smart Cities funded by CNPq, proc. 465446/2014-0, CAPES proc. 88887.136422/2017-00, and FAPESP, proc. 2014/50937-1 and proc. 2015/24485-9.

References

Alvear, O., et al. "Crowdsensing in Smart Cities: Overview, Platforms, and Environment Sensing Issues." Sensors 18.2 (2018): 460.

Anastasi, Giuseppe, et al. "Urban and social sensing for sustainable mobility in smart cities." Sustainable Internet and ICT for Sustainability (SustainIT), 2013. IEEE, 2013.

Aubry, E., Silverston, T., Lahmadi, A., & Festor, O. (2014, March). “CrowdOut: a mobile crowdsourcing service for road safety in digital cities.” In Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on (pp. 86-91). IEEE.

Batista, D. M., Goldman, A., Hirata, R., Kon, F., Costa, F. M., & Endler, M. "Interscity: Addressing future internet research challenges for smart cities." Network of the Future (NOF), 2016 7th International Conference on the. IEEE, 2016.

Blair, G.; Bencomo, N.; France, R. B. “Models@ run.time.” Computer, v. 42, n. 10, 2009.

Borgia, E. “The Internet of Things vision: Key features, applications and open issues.” Computer Communications, v. 54, p. 1-31, 2014.

Borja, R., & Gama, K. (2014). “Middleware para cidades inteligentes baseado em um barramento de serviços. ” X Simpósio Brasileiro de Sistemas de Informação (SBSI), 1, 584-590.

Celino, I., Kotoulas, S. “Smart Cities [Guest editors' introduction].” IEEE Internet Computing, v. 17, n. 6, p. 8-11, 2013.

Del Esposte, A. D. M., Kon, F., Costa, F. M., & Lago, N. “InterSCity: A Scalable Microservice-based Open Source Platform for Smart Cities”, 6th International Conference on Smart Cities and Green ICT Systems (SMARTGREENS), Porto, Portugal, 2017

Diniz, H. B., Silva, E. C. G. F., and Gama, K. "Uma Arquitetura de Referência para Plataforma de Crowdsensing em Smart Cities." XI Brazilian Symposium on Information System. 2015.

Filipponi, L., Vitaletti, A., Landi, G., Memeo, V., Laura, G., & Pucci, P. (2010, July). “Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors.” In Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on (pp. 281-286). IEEE.

Ganti, R. K.; YE, F.; LEI, H. “Mobile crowdsensing: Current state and future challenges.” Communications Magazine, IEEE, 49(11):32–39, 2011

Melo, P. C. F. "CSVM: Uma plataforma para crowdsensing móvel dirigida por modelos em tempo de execução." (2014).

OMG, Q. V. T. “Meta object facility (mof) 2.0 query/view/transformation specification.” Final Adopted Specification (November 2005), 2008.

Petkovics, A., et al. "Crowdsensing solutions in smart cities: Introducing a horizontal architecture." Proceedings of the 13th International Conference on Advances in Mobile Computing and Multimedia. ACM, 2015.

Piro, G., et al. "Information centric services in smart cities." Journal of Systems and Software 88 (2014): 169-188.

Santana, E. F. Z., Chaves, A. P., Gerosa, M. A., Kon, F., & Milojicic, D. S. (2017). Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture. ACM Computing Surveys (CSUR), 50(6), 78.

Soldatos, J., Kefalakis, N., Hauswirth, M., Serrano, M., Calbimonte, J. P., Riahi, M. & Skorin-Kapov, L. (2015). Openiot: Open source internet-of-things in the cloud. In Interoperability and open-source solutions for the internet of things (pp. 13-25). Springer, Cham.

Stojanovic, D., Predic, B. and Stojanovic, N. "Mobile crowd sensing for smart urban mobility." European Handbook of Crowdsourced Geographic Information (2016): 371.

Zappatore, M., Longo, A., & Bochicchio, M. A. Using mobile crowd sensing for noise monitoring in smart cities. In Computer and Energy Science (SpliTech), International Multidisciplinary Conference on (pp. 1-6). IEEE, 2016.

Análise do Impacto de Chuvas na Velocidade Média do Transporte Público Coletivo de Ônibus em Recife

Alexandre S. G. Vianna, Michael O. Cruz, Luciano Barbosa, Kiev Gama

[1]Centro de Informática – Universidade Federal de Pernambuco (UFPE) – Recife - PE
Av. Jornalista Aníbal Fernandes S/N – 50740-560 – Recife – PE – Brasil

{asgv,moc,luciano,kiev}@cin.ufpe.br

Abstract. *The adverse weather conditions represent an emblematic element that adversely affects the quality of public transport, particularly in tropical climate regions the rains constitute the main climatic event of this type. This paper explores the relationship between the rainfall episodes and the changes in the behavior of the average speed of public transport buses in the city of Recife. The work encompasses the use of descriptive statistics techniques to evaluate the itinerary of buses, positioning and speed data in contrast with the rainfall data recorded in the rain gauges scattered throughout the city of Recife. It is detailed in this study the analysis of regions known for traffic problems on rainy day. The results are presented and discussed in the paper.*

Resumo. *As condições climáticas adversas representam um fator emblemático que afeta negativamente a qualidade do transporte público, especialmente em regiões de clima tropical as chuvas são o principal evento climático deste tipo. Este artigo explora as relações entre eventos de chuva e o comportamento da velocidade média dos ônibus de transporte público na cidade do Recife. O trabalho envolve o uso de técnicas de estatística descritiva para analisar dados do itinerário, posicionamento e velocidade dos ônibus em contraste com os dados de precipitação em estações pluviométricas espalhadas pela cidade do Recife. Foram detalhadas as análises de locais conhecidos por problemas de trânsito em dias de chuva, os resultados são apresentados e discutidos no artigo.*

1. Introdução

A influência do clima na dinâmica das redes viárias pode ter diferentes tipos de impacto nas cidades. Alguns estudos mostram correlações entre mudanças climáticas e o número de acidentes [Keay and Simmonds 2006], [Songchitruksa and Balke 2006], [Koetse and Rietveld 2009], enquanto outros mostram que o nível de congestionamentos pode ser agravado [Hofmann and O'Mahony 2005], [Smith et al. 2004]. Os impactos do clima afetam fatores econômicos, comportamento do tráfego, mudanças de itinerário, rotas, entre outros.

Avaliar esses impactos e obter informações que deem suporte para melhorar as tomadas de decisões na gestão do trânsito é um fator importante. Por exemplo, o governo britânico, por meio do Departamento de Transporte[1], tem analisado vários dados de trânsito, desde 1950, como acidentes, congestionamentos, nível de poluição de forma a melhorar tomadas de decisões relacionadas ao tráfego.

Nos últimos anos, dados de GPS *(Global Position System)* têm sido coletados com maior facilidade graças à popularização de aparelhos celulares e dispositivos de IoTs

[1]https://www.gov.uk/government/uploads/system/uploads/attachment_data/file/661933/tsgb-2017-report-summaries.pdf

(Internet of Things). Muitas iniciativas existem para tornarem alguns desses dados abertos como Bikely[2], Microsoft Geolife[3] e Facebook[4].

A massiva quantidade de dados gerados por dispositivos com GPS dá a oportunidade de analisar e entender a dinâmica do tráfego viário, além de favorecer o desenvolvimento de vários tipos de aplicações na área de ITS *(Intelligent Transport System)* como mineração de rotas [Savage et al. 2010], descobertas de sub-trajetórias [Pelekis et al. 2007], detecção de anomalias [Pan et al. 2013] etc. Além dos dados de trajetórias, outros dados de diferentes naturezas podem ser analisados conjuntamente como dados de clima, índices de poluentes, acidentes, ou seja, o volume de dados e a sua pluralidade caracteriza o que se chama de *Big Data* e, diante disso, dá a oportunidade de análise e criação de aplicações sobre dados que outrora não seria possível.

Neste trabalho, pretende-se fazer uma análise inicial do impacto das chuvas no serviço de transporte público da cidade do Recife, Brasil. A análise é feita por meio de técnicas de estatística descritiva a fim de conhecer o comportamento dos ônibus e qual o impacto as chuvas podem causar na dinâmica das velocidades.

A cidade do Recife apresenta um perfil interessante para este estudo pois é um local propício a ocorrência de alagamentos por diversas causas: chuvas torrenciais, inundações fluviais, planície de baixa altitude e marés altas de maior amplitude [Cabral and Alencar 2005]. Além disso, a estrutura viária apresenta problemas conhecidos em pontos de alagamento.

A principal contribuição deste trabalho é, portanto, responder as seguintes perguntas no contexto da cidade do Recife: Como os níveis de chuva afetam a dinâmica do serviço de ônibus? Se afetam, como medir o impacto quantitativamente principalmente com relação à velocidade?

Este trabalho está organizado da seguinte forma: (i) a seção 2 apresenta uma revisão da literatura de trabalhos que trazem abordagens de análise de tráfego com mudanças climáticas; (ii) seção 3 descreve qual o conjunto de dados utilizado; (iii) a seção 4 descreve a metodologia utilizada para analisar os dados; (iv) na seção 5 os resultados da análise são discutidos e (iv) na seção 6 conclusões e trabalhos futuros.

2. Trabalhos relacionados

Nesta seção são apresentados alguns trabalhos que analisam o impacto de mudanças climáticas no trânsito. Os impactos podem causar variações no fluxo de veículos, velocidade, número de acidentes, densidade (número de veículos em um determinado comprimento da rodovia), tempo médio de viagem, etc.

[Hofmann and O'Mahony 2005] realiza um trabalho com objetivo de investigar o impacto das condições climáticas no transporte urbano de forma que se possa medir o comportamento dos passageiros que utilizam o serviço. Para alcançar tal objetivo, o trabalho propõe utilizar dados de meteorologia e de ônibus de forma a traçar correlações de forma que se possa medir alguns fatores: quantidade de pessoas que utilizam o serviço público *(ridership)*, frequência do serviço (disponibilidade do serviço), a distância entre os ônibus em uma rota *(headways)*, agrupamento de ônibus *(bus bunching)*, tempo de viagem e a variabilidade do tempo de viagem. Todos os fatores são analisado levando em consideração dois cenários: dias com chuva e sem chuva. De acordo com o autor, a chuva

[2]http://www.bikely.com/

[3]http://research.microsoft.com/en-us/projects/geolife/

[4]www.facebook.com

influencia negativamente as medidas *ridership* e *travel time*, entretanto nas medidas de *bus bunching* e *headway*, a chuva contribui positivamente, ou seja, menos agrupamento e melhora na frequência dos ônibus.

[Cools et al. 2010] objetiva examinar se as condições climáticas alteram uniformemente a intensidade do tráfego na Bélgica. De acordo com os autores, alguns fatores do tempo como chuva, neve, neblina e outros, influenciam vários aspectos do tráfego, por exemplo, intensidade do tráfego, velocidade etc. Para avaliar os efeitos do clima, os autores utilizaram dados reais providos por *inductive loop detector* que coletou dados durante o período de 1 ano (2003-2004). Para avaliar o impacto que as variações climáticas causam no trânsito, o trabalho utilizou a técnica de correlação não paramétrica de *Spearman* e a técnica de Regressão Linear. Os resultados mostraram que chuvas, nebulosidade e ventania são negativamente correlacionadas com a intensidade do tráfego, enquanto boas condições climáticas (dias ensolarados) são proporcionais a intensidade do tráfego. O trabalho apresentou os seguintes resultados: a intensidade do tráfego é maior quando ocorre máximas temperaturas; o impacto das condições climáticas são claramente mais homogêneas em locais próximos de onde as informações foram coletadas do que em diferentes locais. Também foi notado que rodovias que são mais usadas para lazer são mais suscetíveis às mudanças de tráfego devido às variações climáticas do que àquelas avenidas que são utilizadas para acesso *casa -> trabalho, trabalho -> casa*.

[Chung et al. 2005] apresentam uma análise do impacto das chuvas na cidade de Tóquio (Japão). Os autores utilizam dados reais tanto de meteorologia quanto de tráfego. Os dados foram coletados durante 6 anos e a análise foi feita apenas considerando a via arterial (Tokyo Metropolitan Expressway). Para realizar a análise, o trabalho configurou o experimento da seguinte forma: dias da semana (segunda até sábado) e feriados (domingos e feriados de fato). Os autores realizaram a análise levando em consideração dias com chuva e sem chuva. Um dia só foi considerado chuvoso se a precipitação fosse maior ou igual a 13 mm, assim, 216 dias foram considerados chuvoso num espaço de 6 anos. Os resultados mostraram que os dias de sábado e domingo são mais sensíveis no sentido de haver um decrescimento na quantidade de viagem quando ocorre chuva. O trabalho também encontrou uma correlação positiva do número de acidentes nos finais de semana e que a frequência de acidentes em dias chuvosos é maior do que nos outros dias.

[Andersson and Chapman 2011] propõem analisar acidentes de tráfego no município *West Midlands*, Reino Unido, durante o inverno com o objetivo de aplicar cenários de mudança climática da UKCIP *(UK Climate Impact Program)* para estimar mudanças quanto ao número de acidentes no tráfego, juntamente com dados de clima gerados artificialmente com base nos cenários. Os dados foram gerados levando em consideração dois modelos estocásticos usados pelo *EARWIG (Environment Agency Rainfall and Weather Impacts Generator)*. Os resultados dos cenários (2020, 2050 e 2080 anos) mostram que a quantidade de dias com temperaturas abaixo de 0°e 5°tendem a diminuir, proporcionando dias mais quentes. Foi encontrada uma correlação entre a redução de dias com temperatura abaixo de 5°e a redução do número de acidentes. Mesmo notando tal correlação, outras variáveis que não são levadas em consideração podem afetar os resultados. O trabalho conclui que o número de dias com temperatura abaixo de 0°diminuirão no futuro e isto reduzirá de 43% no número de acidentes em *West Midlands*.

Embora os trabalhos citados apresentem análise sobre como as condições climáticas podem influenciar o comportamento dos usuários de transportes urbanos [Hofmann and O'Mahony 2005], a intensidade do tráfego [Cools et al. 2010], a quantidade de acidentes [Chung et al. 2005], [Andersson and Chapman 2011], nenhum deles,

até o nosso atual conhecimento, propõem-se a verificar pontualmente o quanto a velocidade média dos transportes públicos (ônibus) aumentam ou diminuem de acordo com os níveis de precipitação pluviométrica. Informações pontuais, como as produzidas neste trabalho podem ser utilizadas de forma precisa pelos gestores de tráfego viário para realizar tomadas de decisão.

3. Conjunto de Dados

Os dados do sistema de ônibus foram fornecidos pelo Grande Recife Consórcio de Transporte Metropolitano [5] que gerencia o transporte de ônibus na região metropolitana do Recife, e incluem coordenadas das rotas, registros de GPS dos ônibus, os quais denominamos de *ping*. Cada *ping* contém os identificadores da linha e rota, localização e velocidade dos ônibus. O conjunto de dados disponibilizado abrange apenas o período entre 17/10/2017 e 24/01/2018, possui 129 linhas e 891 ônibus. Os ônibus emitem um *ping* a uma frequência de 30 segundos, e a base disponibilizada para este trabalho tem 73 milhões de *pings*. Estes *pings* estão visualmente representados em vermelho no mapa da Figura 1, o qual apresenta em destaque o perímetro do município do Recife.

Os dados pluviométricos foram obtidos no site do CEMADEN (Centro Nacional de Monitoramento e Alertas de Desastres Naturais[6]), o qual disponibiliza o volume de chuvas acumuladas a cada hora em cada uma das 18 estações pluviométricas que estão geograficamente distribuídas pela cidade. As localizações das estações pluviométricas também são fornecidas e estão apresentadas com a cor alaranjada no mapa da Figura 1.

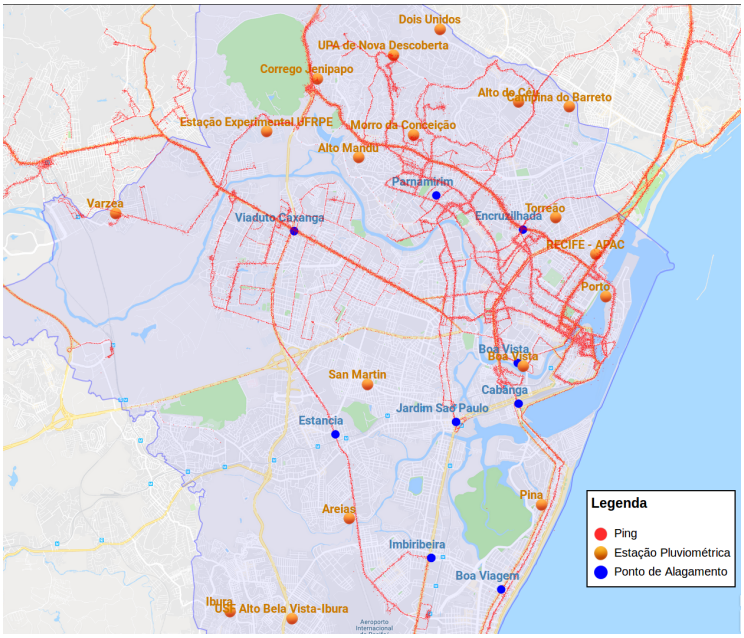


Figura 1. Localização do Conjunto de Dados Georreferenciados

Este trabalho envolve a análise do impacto das chuvas no transporte público em regiões críticas de alagamento do Recife, as regiões críticas de alagamento foram obtidas em um estudo da Emlurb (Empresa de Manutenção e Limpeza Urbana) [EMLURB 2013] que catalogou 159 pontos de alagamentos. Todos os dados sobre ônibus, pluviômetros e

[5]http://www.granderecife.pe.gov.br/web/grande-recife
[6]http://www.cemaden.gov.br/pluviometrosautomaticos/

locais críticos de alagamento foram inseridos em um banco de dados relacional fazendo uso dos recursos geográficos e índices espaciais, sob o qual as análises foram realizadas.

4. Método

Nesta seção apresentamos a abordagem utilizada para avaliar o impacto das chuvas na velocidade média dos ônibus. Para isso, foram realizadas duas tarefas: limpeza e seleção dos dados, e a análise em três etapas.

4.1. Limpeza e Seleção dos Dados

A limpeza dos dados é importante para removermos valores com problemas, tais como dados nulos e valores atípicos *(outliers)* que podem afetar a análise. Para esta tarefa realizamos um processo semi-automatizado em que removemos os registros com problema. Os dados dos ônibus foram disponibilizados em formato CSV *(Comma Separated Values)*, e do total de 73 milhões de *pings*, somente 26 milhões puderam ser aproveitados para os objetivos deste trabalho, pois os demais apresentavam dados de velocidade, localização ou linha, nulos. Os dados dos pluviômetros da CEMADEN não apresentaram problemas relevantes e puderam ser importados automaticamente de um arquivo CSV para a base relacional. Os dados dos pontos de alagamento foram disponibilizados em formato de relatório pela Emlurb, e foi necessário um processo manual de importação.

Após a limpeza, foi realizada a seleção dos dados com propósito de delimitar a análise em dias úteis, e assim os feriados e fins de semana foram removidos. Pois, em principio, estes apresentam comportamento atípico, em que o trânsito é menos intenso e a velocidade média dos ônibus é bastante superior em comparação aos dias úteis. Dentro desse subconjunto foram selecionados os dados de pluviômetros e velocidades de ônibus apenas na faixa de horário entre 6h e 20h, pois entre 20h e 6h a quantidade de ônibus circulando é menor.

Por fim, dos 159 locais de alagamento apresentados pela Emlurb foram selecionados os 8 locais considerados mais críticos. Os critérios para seleção são os locais em que circulam mais ônibus e que estão próximos a vias arteriais principais e secundárias, e cruzamentos de avenidas, sendo representados com a cor azul no mapa da Figura 1.

4.2. Processo de Análise

A ideia de análise proposta é verificar o impacto do volume de chuvas na velocidade média dos ônibus perto dos locais de alagamento, esta região é delimitada por uma circunferência de raio R. Dentro da região comparamos a velocidade média registrada para cada linha de ônibus em horários de chuva versus a média dos dados históricos.

Na primeira etapa da análise, para cada local de alagamento foi calculada a velocidade média histórica e o desvio padrão do histórico dos *pings* enviados dentro da região nos dias em que não ocorreu chuva, este cálculo discrimina dia da semana, faixa de hora e linha de ônibus. Na segunda etapa, uma data alvo e um local alvo são escolhidos para análise, neste trabalho as datas alvos são dias de ocorrência de chuvas e os locais são regiões críticas de alagamento. Mais especificamente, é calculada a velocidade média registrada pelos ônibus na região do local alvo na data alvo. Este cálculo é realizado a cada janela de uma hora e discrimina os valores para cada linha de ônibus. Na terceira etapa, selecionamos estações pluviométricas próximas ao local alvo de análise e realizamos as médias dos acumulados de chuvas agrupando os dados em faixas de uma hora.

A Tabela 1 apresenta um exemplo dos dados resultantes do processo de análise da Linha 644 - Largo Maracanã perto do local Encruzilhada no dia 19/01/2018. Nota-se que

	HORA														
Variáveis	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Max (Km/h)	19	17	14	13	15	15	15	16	17	15	18	17	17	20	21
Min (Km/h)	14	8	7	8	10	10	9	9	9	11	8	7	9	13	16
Mediana	15	9	10	11	12,5	11,5	10,5	12,5	13,5	12,5	13	12	12,5	16	18,5
Média (Km/h)	15,7	11,4	10,1	10,9	12,3	12	11,4	12,5	13,1	12,8	13,3	11,9	13,3	16,5	18,3
Desvio Padrão	1,7	3,3	2,1	1,4	1,9	1,9	2,1	2,5	2,8	1,2	2,8	2,8	3	1,9	1,8
Vel. Registrada 19/01/18	15	15	5	6	9	14	16	16	16	17	14	16	15	20	19
Acumulado Chuvas (mm)	4	5,2	10,5	12,4	8,2	1,5	0	0	0	0	0	0	0,2	0	0

Tabela 1: 19/1/2018 - Encruzilhada - Linha 644

nesta data e local a linha 644 registrou uma velocidade de 5Km/h às 8h enquanto que a média histórica das sextas-feiras nesse mesmo horário é 10.1Km/h.

Para a realização efetiva das análises foi desenvolvido um sistema em linguagem Python [7] fazendo uso da biblioteca Pandas [8] que disponibiliza um conjunto de estruturas de dados e funções estatísticas apropriados para *softwares* do contexto de ciência de dados. Por fim, a Figura 2 sintetiza a metodologia descrita nesta seção.

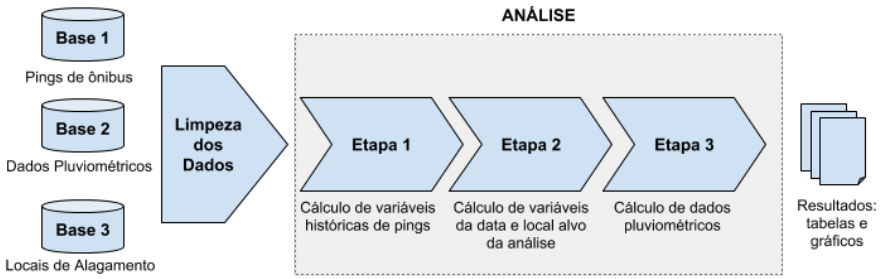


Figura 2. Abordagem utilizada para seleção e análise dos dados.

5. Análises

A abordagem de análise foi aplicada e para tal foi selecionada a data 19/01/2018, pois nesta data ocorreram chuvas com volumes significativos, com acumulados maiores que 10 milímetros por hora em diversas estações pluviométricas. Dentro do período de abrangência dos dados disponibilizados não há outras ocorrências de volume de chuvas de maior intensidade. Foi selecionado o local Viaduto da Caxangá para detalhar a análise, pois é um local próximo às estações pluviométricas que registraram maior volume de chuvas e também é um ponto conhecido por problemas de trânsito em dias de chuva.

5.1. Configurações da Análise

O processo e análise possui algumas variáveis que podem ser configuradas, tais como, faixa de horário diária, remoção de fins de semana, remoção de feriados e o raio da região de análise.

As análises são realizadas nas regiões dos locais típicos de alagamento, e estas regiões são delimitadas por uma circunferência de raio R. Foram realizados alguns testes empíricos com diferentes valores de raio R com intervalos de 100 metros. Os testes demonstraram que $R < 400$ não incluem dados suficientes de *pings* de ônibus que circulam na região, deixando a análise com muitos dados faltantes em algumas faixas de horário. Para $R \geq 1000$ metros são incluídos muitos dados de ônibus que não são afetados pelo

[7] https://www.python.org/
[8] https://pandas.pydata.org/

alagamento, assim os desvios de velocidade média são suavizados por *pings* que não sofrem impacto por estarem muito longe do local alvo. Foi adotado um raio de 700 metros, pois apresenta um equilíbrio entre os parâmetros com uma quantidade *pings* satisfatória.

Por fim, para cada local de alagamento foram selecionadas estações pluviométricas próximas. A disposição geográfica das estações na cidade é bastante irregular, como ilustrado na Figura 1, havendo concentração de estações em algumas regiões e poucas estações em outras. Por esta razão foi adotada uma distância bastante abrangente, assim as estações pluviométricas que se localizam a um $R < 4500$ metros do local de alagamento são inclusas na análise.

5.2. Análise do Dia 19/01/2018

O volume de chuvas da sexta-feira dia 19/01/2018 pode ser observado na Tabela 2. Todas as estações pluviométricas registraram chuvas maiores que 10mm, sendo que metade das estações chegou a 20mm, e em específico a estação Várzea atingiu 36mm. As chuvas se concentraram na faixa de horário de 6h às 11h.

	HORA															
Estação	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**Acumulado**
Várzea	6	1	16	36	12	2	0	0	0	0	0	0	0	0	0	73
Córrego Jenipapo	6	2	13	16	17	2	0	0	0	0	0	0	3	1	0	60
Alto Mandu	5	4	18	19	10	2	0	0	0	0	0	0	1	0	0	59
UPA de Nova Descoberta	7	3	10	21	15	2	0	0	0	0	0	0	1	0	0	59
Areias	2	0	21	27	3	3	0	0	0	0	0	0	0	0	0	56
Estação Experimental UFRPE	4	2	17	15	13	2	0	0	0	0	0	0	0	0	0	53
Alto do Céu	5	8	6	13	12	1	0	0	0	0	0	0	0	0	0	45
San Martin	2	2	15	24	4	3	0	0	0	0	0	0	0	0	0	50
Morro da Conceição	5	4	13	14	12	1	0	0	0	0	0	0	1	0	0	50
USF Alto Bela Vista-Ibura	0	0	13	24	2	3	0	0	0	0	0	0	0	0	0	42
Campina do Barreto	6	11	4	11	9	1	0	0	0	0	0	0	0	0	0	42
Porto	5	8	7	11	8	1	0	0	0	0	0	0	0	0	0	40
Torreão	5	6	8	10	10	1	0	0	0	0	0	0	0	0	0	40
Pina	0	4	19	11	2	2	0	0	0	0	0	0	0	0	0	38
Dois Unidos	5	5	10	10	7	0	0	0	0	0	0	0	0	0	0	37
Ibura	0	0	13	20	1	3	0	0	0	0	0	0	0	0	0	37
Boa Vista	3	2	11	8	4	2	0	0	0	0	0	0	0	0	0	30
Recife - APAC	1	5	8	2	9	1	0	0	0	0	0	0	0	0	0	26

Tabela 2: Chuvas em milímetros - 19/1/2018 - Acumulado em frequência 1 hora

A Tabela 3 apresenta os dados do dia 19/01/2018 com os locais de alagamento que demonstram diferenças mais expressivas na velocidade média, tais como, Viaduto Caxangá, Encruzilhada, Parnamirim e Boa Vista, enquanto os demais locais apresentaram impactos menos contundentes e não foram inclusos na tabela.

		HORA														
Local	**Variáveis**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Viaduto Caxangá	Max	20,7	19,7	19,3	20,7	20	20,3	20	22	21	19,7	24,7	19	18	21,3	22,3
	Min	17,7	16,3	18,3	18,3	17,3	17,7	15,5	16	15,7	15,7	16	14	11,7	15	13,7
	Mediana	19,5	18,7	18,8	19,2	18,8	19	18	18,3	18,2	17,5	16,7	16,5	15,2	17,7	20,7
	Média	19,3	18,3	18,8	19,3	18,7	19,1	17,9	18,6	18,4	17,4	17,9	16,2	15,3	18,1	19,8
	Desvio Padrão	1,1	1,3	0,3	0,9	0,9	1,3	1,7	2,6	2,1	1,6	3,4	1,8	2,2	2,5	3,2
	V. R. 19/1/2018	18	19	16,7	16	9,3	7,7	16,7	19	17,3	15,3	16	17	17,7	19,7	22
Boa Vista	Max	26	24	26	23	26	25	27	28	23	22	23	20	22	24	26
	Min	22	15	17	18	16	19	18	19	14	11	11	10	8	13	18
	Mediana	24	20	22	21	22,5	20	23,5	24	16,5	16,5	13,5	11,5	12	18	23
	Média	23,7	20,1	21,1	20,7	21,8	21,4	22,9	23,6	18,2	16,7	15,4	13,2	14	17,9	23
	Desvio Padrão	1,3	2,8	2,7	1,8	3	2,2	2,7	2,8	3,2	3,2	3,9	3,5	4,9	3,1	2,4
	V. R. 19/1/2018	20	21	8	16	22	22	21	24	21	21	21	19	19	21	28
Encruzilhada	Max	19,7	17,3	15,7	14,7	16	17,7	17,7	18,7	17,7	17,7	17,3	16,3	18,3	21	23
	Min	15	9	8,3	10,3	7,7	11,7	11,7	10,3	9,7	12	10,3	10,3	12	13,3	18,3
	Mediana	16,3	10	10	12,5	14	14,7	12,8	15	14,2	15,8	14,7	12,8	13,2	17,3	21,2
	Média	16,7	12,2	11,3	12,6	13,4	14,6	13,9	14,7	14,1	15,4	14,5	13,2	14,5	17,5	20,9
	Desvio Padrão	1,5	3,4	2,5	1,5	2,6	2	2,3	2,8	2,8	1,8	2,2	2,1	2,4	2,2	1,4
	V. R. 19/1/2018	17	14	7,3	7,7	10,7	16	17,7	16,7	19,3	17	15,7	16,7	15,3	20,3	21,3

Tabela 3: Velocidade Média Registrada nos Locais de Alagamento no dia 19/1/2018

Um caso interessante é o local Viaduto da Caxangá às 11h, quando a velocidade registrada no dia 19/01/2018 foi 7,7Km/h bastante inferior aos 17,7Km/h que é a mínima

histórica dos dias que não choveu. O gráfico da Figura 3 apresenta a média em milímetros da intensidade de chuva das estações pluviométricas impactantes, e também o comportamento histórico da velocidade média, desvio padrão em dias que não ocorreu chuva e a velocidade registrada no dia 19/01/2018. No gráfico nota-se que, na faixa de horário de 8h as 11h, o comportamento da velocidade média no dia 19/01/2018 se manteve inferior à média histórica: 34% menor que a média histórica para esta faixa de horário.

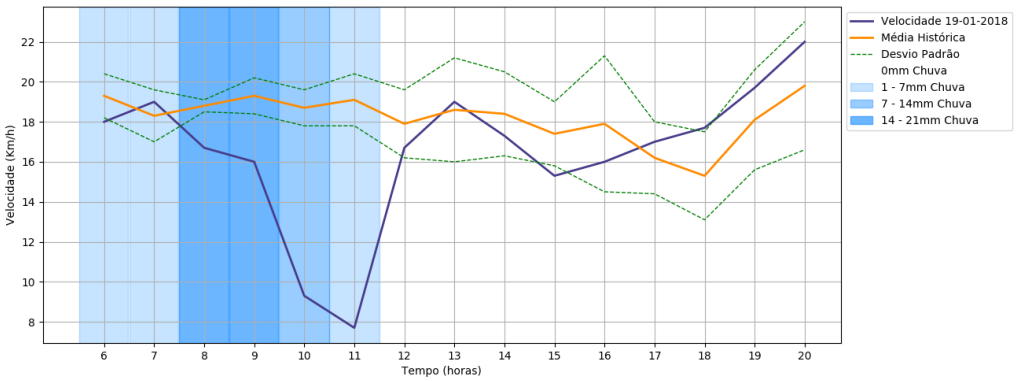


Figura 3. Velocidade Média no Viaduto da Caxangá no dia 19/1/2018

A linha 2480 TI Camaragipe/Derby BRT foi selecionada para uma análise isolada. A Figura 4 exibe em amarelo a circunferência com raio de 700 metros do ponto de alagamento e em verde o percurso de linha 2480 na região do local Viaduto Caxangá. Observa-se que o trajeto da linha passa pela avenida Caxangá e cruzando a BR101 por baixo do viaduto nas rotas de ida e volta. A distância percorrida é de 1400 metros na ida e 1400 metros na volta. A tabela 4 apresenta os dados históricos de máximo, mínimo, média, mediana, desvio padrão, e os dados de acumulados de chuvas e velocidade registrada no dia 19/01/2018. A Figura 5 apresenta o gráfico com os dados de chuvas, velocidade média registrada no dia 19/01/2018 e velocidades médias e desvios padrões históricos para o local Viaduto Caxangá.

Figura 4. Trajeto Linha 2880 - Viaduto Caxangá

	HORA														
Variáveis	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
Max (Km/h)	23	22	22	23	22	25	22	32	23	22	21	22	22	21	22
Min (Km/h)	18	18	20	20	18	18	17	12	16	18	17	17	13	18	16
Mediana	21,5	19,5	21	21,5	21	21	20	22	20	20,5	19,5	18,5	19,5	20	21,5
Média (Km/h)	20,8	19,8	21	21,5	20,5	21,2	19,8	21,6	19,6	20,2	19,3	19	18,2	19,7	20,5
Desvio Padrão	1,9	1,5	0,9	1	1,4	2,6	1,9	7,4	2,5	1,5	1,4	2,1	3,8	1,4	2,3
Vel. Registrada 19/01/18	20	20	21	19	13	5	22	22	24	20	20	22	22	23	24
Acumulado Chuvas (mm)	4,7	2,5	15,3	20,7	11,3	2	0	0	0	0	0	0	0,8	0,2	0

Tabela 4: Dados Linha 2480 Viaduto Caxangá dia 19/1/2018

Ao analisar a linha 2480 em isolado, observa-se que, durante o período de 9h às 11h, a velocidade média diminuiu além do desvio padrão. Ao comparar a velocidade média registrada em 19/01/2018 de 14,5Km/h há uma diminuição de 31% em relação à média histórica que é 21,05Km/h, isso reflete no tempo médio para realizar o percurso

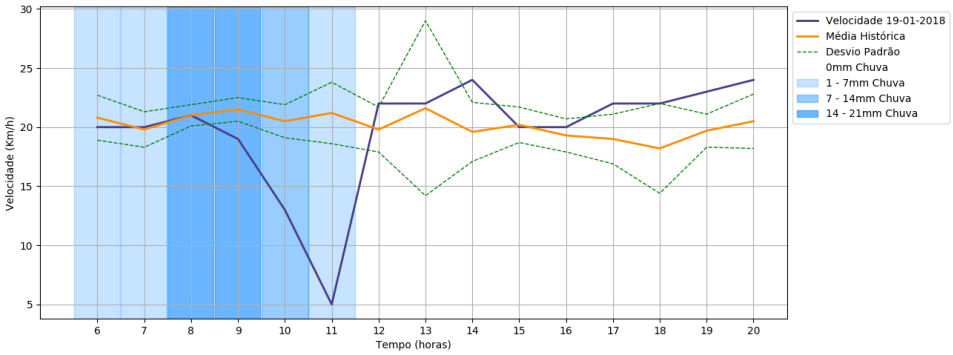


Figura 5. Velocidade Média Linha 2480 no Viaduto da Caxangá em 19/1/2018

de 1400 metros que tipicamente em uma sexta-feira sem chuva é de 4 minutos e em 19/01/2018 foi de 5 minutos 42 segundos.

Uma ressalva é que os resultados foram obtidos com base em períodos que não refletem o cotidiano da cidade, os meses de Dezembro e Janeiro são bastante atípicos e há diversos fatores que influenciam na variança dos dados, tais como, férias escolares, feriados etc. Outro ponto importante é o regime de chuvas do verão, há poucos eventos de chuva relevantes para o trabalho e assim não foi possível realizar análises mais variadas e avançadas. Mesmo com estes pontos a serem considerados, as análises realizadas trouxeram resultados que estão alinhados com a ideia de que chuvas provocam problemas de trânsito nos locais estudados. No caso específico do dia 19/01/2018, há notícias que evidenciam a ocorrência de problemas de tráfego que prejudicam o transporte público. [G1 2018]

6. Conclusão e Trabalhos Futuros

Este trabalho apresentou um processo de análise de dados do sistema de ônibus baseado em estatística descritiva, e focou a identificação de impactos que as chuvas podem causar na velocidade média dos ônibus. Como contribuição principal a metodologia utilizada se mostrou efetiva em responder as questões inciais sobre como os níveis de chuva afetam a dinâmica do serviço de ônibus e como medir o impacto quantitativamente com relação à velocidade. Foi possível identificar a variação da velocidade média dos ônibus em um dia de chuva. A análise da linha 2480 na região Viaduto da Caxangá no dia 19/01/2018 apresentou uma redução de 31% da velocidade média em relação aos dados históricos de dias sem chuva.

Como trabalhos futuros o processo de análise desenvolvido pode ser replicado em outras cidades com objetivo de comparação. A análise também pode ser expandida para verificar os impacto de outras variáveis, como acidentes de trânsito, fluxo de veículos, horários de pico, grande eventos, obras públicas, entre outros. Atualmente está em andamento a disponibilização do sistema de software em interface Web que possa ser utilizado por gestores para análise e suporte à tomada de decisão no contexto de transporte público. Por fim, os trabalhos futuros devem envolver a análise de mais variáveis e o uso de outras técnicas estatísticas, tais como a realização de testes de hipóteses e correlações entre locais de redução de velocidade média e o volume de chuvas em estações pluviométricas.

7. Agradecimentos

Esta pesquisa foi parcialmente financiada pelo INES 2.0, FACEPE PRONEX APQ 0388-1.03/14 e CNPq 465614/2014-0. Os autores agradecem a Fernando Guedes e Alexandre

Severo, do Grande Recife Consórcio de Transporte Metropolitano, pelo provimento dos dados utilizados nesta análise.

Referências

Andersson, A. K. and Chapman, L. (2011). The impact of climate change on winter road maintenance and traffic accidents in west midlands, uk. *Accident Analysis & Prevention*, 43(1):284–289.

Cabral, J. J. S. P. and Alencar, A. V. (2005). Recife e a convivência com as águas. In *Gestão do Território e Manejo Integrado das águas urbanas*, pages 109–117.

Chung, E., Ohtani, O., Warita, H., Kuwahara, M., and Morita, H. (2005). Effect of rain on travel demand and traffic accidents. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 1080–1083. IEEE.

Cools, M., Moons, E., and Wets, G. (2010). Assessing the impact of weather on traffic intensity. *Weather, Climate, and Society*, 2(1):60–68.

EMLURB (2013). Elaboração dos estudos de concepção para gestão e manejo de águas pluviais e drenagem urbana do recife (versão concedida em visita técnica). *Empresa de Manutenção e Limpeza Urbana*.

G1 (2018). Chuva alaga ruas e avenidas do grande recife. disponível em: <https://g1.globo.com/pe/pernambuco/noticia/chuvas-causam-alagamentos-e-transtornos-no-transito-no-grande-recife.ghtml> acesso 2 de abril de 2018.

Hofmann, M. and O'Mahony, M. (2005). The impact of adverse weather conditions on urban bus performance measures. In *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, pages 84–89. IEEE.

Keay, K. and Simmonds, I. (2006). Road accidents and rainfall in a large australian city. *Accident Analysis & Prevention*, 38(3):445–454.

Koetse, M. J. and Rietveld, P. (2009). The impact of climate change and weather on transport: An overview of empirical findings. *Transportation Research Part D: Transport and Environment*, 14(3):205–221.

Pan, B., Zheng, Y., Wilkie, D., and Shahabi, C. (2013). Crowd sensing of traffic anomalies based on human mobility and social media. *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - SIGSPATIAL'13*, pages 334–343.

Pelekis, N., Kopanakis, I., Marketos, G., Ntoutsi, I., Andrienko, G., and Theodoridis, Y. (2007). Similarity Search in Trajectory Databases. *14th International Symposium on Temporal Representation and Reasoning (TIME'07)*, pages 129–140.

Savage, N. S., Nishimura, S., Chavez, N. E., and Yan, X. (2010). Frequent trajectory mining on GPS data. *Proceedings of the 3rd International Workshop on Location and the Web - LocWeb '10*, pages 1–4.

Smith, B. L., Byrne, K. G., Copperman, R. B., Hennessy, S. M., and Goodall, N. J. (2004). An investigation into the impact of rainfall on freeway traffic flow. In *83rd annual meeting of the Transportation Research Board, Washington DC*. Citeseer.

Songchitruksa, P. and Balke, K. (2006). Assessing weather, environment, and loop data for real-time freeway incident prediction. *Transportation Research Record: Journal of the Transportation Research Board*, (1959):105–113.

Patrocinador Diamante

GOVERNO
DO RIO GRANDE DO NORTE

Patrocinadores Bronze

ESiG
Software & consultoria

dynavideo

CABOTELECOM

Sistema FIERN

PREFEITURA DO NATAL
A NOSSA CIDADE

PARNAMIRIM
Cuidando de você.

Apoio Financeiro

CAPES

MINISTÉRIO DA EDUCAÇÃO

BRASIL
GOVERNO FEDERAL

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

nic.br
Núcleo de Informação e Coordenação do Ponto BR

cgi.br
Comitê Gestor da Internet no Brasil